Instituto Tecnológico Superior de Misantla

Generación de protocolos de investigación implementando procesamiento de lenguaje natural

TESIS

QUE PARA OBTENER EL GRADO DE:

Maestro en Sistemas Computacionales

Presenta:

Ing. Sergio Faustino Vázquez Reyes

Director:
Dr. Eddy Sánchez de la Cruz

Co-Director:
Mtro. César Primero Huerta

Misantla, Veracruz, Julio de 2023

INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA

AUTORIZACIÓN DE IMPRESIÓN DE TRABAJO DE TITULACIÓN MAESTRÍA

FECHA: 29 DE AGOSTO 2023

ASUNTO: **AUTORIZACIÓN DE IMPRESIÓN: TESIS.**

A QUIEN CORRESPONDA:

Por medio de la presente se hace constar que el (la) C:

SERGIO FAUSTINO VÁZQUEZ REYES

estudiante de la maestría en Sistemas Computacionales con No. de Control 212T0440 ha cumplido satisfactoriamente con lo estipulado por el **Lineamiento de Posgrado para la obtención del grado de Maestría** mediante **Tesis.**

Por tal motivo se **Autoriza** la impresión del **Tema** titulado:

"GENERACIÓN DE PROTOCOLOS DE INVESTIGACIÓN IMPLEMENTANDO PROCESAMIENTO DE LENGUAJE NATURAL"

Dándose un plazo no mayor de un mes de la expedición de la presente a la solicitud del examen para la obtención del grado de maestría.

SEP SEV INSTITUTO TECNOLOGICO SUPERIOR DE MISANTLA

ATENTAMENTE

DR. EDDY SÁNCHEZ DE LA CRUZ
Presidente

COORDINACIÓN DE TITULACIÓN

DR. DAVID LARA ALABAZARES
Secretario

DR. IRAHAN OTONIEL JOSÉ GUZMÁN
Vocal

Archivo.

Dedicatoria

Este trabajo de tesis esta dedicado a mi amada esposa Adriana Cabrera Cruz quien me apoyó durante toda la maestría y aguantó la distancia que nos separaba mientras yo estudiaba.

También está dedicado a aquél ser que está formándose en el vientre de mi esposa, quien espero con tantas ganas de conocer y ser un ejemplo en su vida.

Agradecimientos

Agradezco de corazón a todas aquellas personas que me brindaron su apoyo incondicional a lo largo de esta etapa académica tan importante en mi vida. Sus palabras de aliento fueron fundamentales para alcanzar este logro.

A mi amada esposa, mi compañera de vida y mi mayor fuente de inspiración, te agradezco profundamente por tu amor incondicional, paciencia y comprensión durante este arduo proceso. Tu apoyo inquebrantable, tus palabras de aliento y tu presencia constante fueron el motor que me impulsó a seguir adelante en los momentos más desafiantes. Tu sacrificio y comprensión de mis largas horas de estudio no tienen precio, y no puedo expresar con palabras cuánto valoro tu apoyo.

A mis respetados profesores, agradezco por su dedicación y compromiso en transmitirme sus conocimientos. Sus enseñanzas no solo me permitieron crecer académicamente, sino que también me motivaron a seguir superándome constantemente. Sus comentarios constructivos y su orientación han sido invaluables para el desarrollo de este trabajo.

A mi familia, agradezco por su amor y apoyo constante a lo largo de mi vida. Gracias por creer en mí, por motivarme y por brindarme un ambiente propicio para mi desarrollo académico. Su aliento y confianza en mis capacidades me han dado la fuerza necesaria para enfrentar cualquier obstáculo.

A mis queridos amigos, quienes han estado a mi lado en cada paso del camino, les agradezco por su amistad sincera y por su apoyo. Sus palabras de aliento, momentos de distracción y ánimo fueron cruciales para mantenerme enfocado y motivado. Gracias por comprender mis ausencias y por celebrar mis éxitos como si fueran propios.

Finalmente, quiero expresar mi gratitud a todas las personas que, de alguna manera u otra, han contribuido a este logro. Cada una de sus palabras de aliento, consejos y buenos deseos ha dejado una huella profunda en mi corazón.

Resumen

El protocolo de un proyecto de investigación es una de las partes más importantes de una investigación, en ella se delimita el proyecto, se establece una hipótesis y se plantean objetivos. Por esto mismo, el desarrollo de un buen protocolo puede consumir más tiempo del deseado durante el proyecto y es necesario encontrar maneras para realizarlo de forma más eficiente y eficaz.

En esta tesis se presenta la implementación del algoritmo de procesamiento de lenguaje natural GPT2-xl en una API (Application progamming interface) de python con comunicación con una aplicación web para la generación de protocolos de proyectos de investigación con texto ingresado por el usuario y una plantilla de contexto.

Se presenta la arquitectura del proyecto, su funcionamiento en 2 servidores y las tecnologías implementadas en cada uno para su correcto funcionamiento.

Igualmente, se realizaron pruebas con usuarios para verificar la calidad del texto generado por el modelo de lenguaje con la plantilla de contexto creada. La población que realizó las pruebas consta de hombres y mujeres de entre 25-30 años, con educación superior o maestría, laborando en el ámbito de desarrollo de software.

Abstract

The protocol of a research project is one of the most important parts of an investigation. It defines the project scope, establishes a hypothesis, and sets objectives. Therefore, developing a good protocol can consume more time than desired during the project, and it is necessary to find ways to make it more efficient and effective.

This thesis presents the implementation of the GPT2-xl language model in a Python API (Application progamming interface) with communication to a web application for generating research project protocols using user-entered text and a contextual template. The project's architecture is presented, along with its functioning on two servers and the technologies implemented in each for proper operation.

Additionally, tests were conducted with users to verify the quality of the text generated by the language model using the created contextual template. The test population consists of men and women aged 25-30, with higher education or a master's degree, working in the software development field.

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

Capítulo 1

Generalidades

1.1. Introducción

En los albores del siglo XXI la investigación reviste especial importancia pues la creatividad y la generación de conocimiento se han convertido en factores claves de éxito y progreso [1].

Realizar trabajos de investigación es la forma en la que aumenta el conocimiento del universo [2], se crea nueva tecnología [3], se solucionan problemas [4], y se mejora la calidad de vida de las personas [5]. Por esto, es esencial poder realizarlos de manera eficiente y eficaz.

La implementación de tecnologías basadas en inteligencia artificial es un hecho que se ha posicionado en nuestro día a día de forma que cada vez es más común leer acerca de sus nuevos usos en el desarrollo de trabajos anteriormente solo desarrollados por humanos [6].

Actualmente, se han implementado modelos lenguaje para realizar tareas tales como la creación de arte [7], escritura de artículos periodísticos [8] y conversación fluida con chatbots [9], se ha demostrado que la generación de contenido basado en texto es posible, fácil y cómoda de utilizar para los usuarios.

De esta forma, es lógico pensar que una aplicación para el desarrollo de proyectos de investigación implementando modelos de lenguaje con base de texto es el siguiente paso.

1.2. Planteamiento del problema

Una investigación científica conlleva un proceso que debe cumplirse previamente a la experimentación y obtención de resultados, consistente en la selección del tema, reconocimiento de la problemática, planteamiento de objetivos, justificación y delimitación de la investigación, marco de referencia, hipótesis de la investigación, diseño experimental, obtención y procesamiento de información [10].

Desde estudiantes universitarios hasta investigadores consolidados, todos inician un proyecto desde la definición del tema de estudio y deben realizar un protocolo de investigación para dicho proyecto. Ya que es una parte esencial en cada proyecto, encontrar la forma

de ayudar a los investigadores a realizar esta tarea de forma eficiente y eficaz es de suma importancia.

“El aspecto del planteamiento... es la etapa, siempre difícil, de seleccionar un problema de investigación interesante, novedoso, importante, verificable y bien delimitado"[11]

El protocolo debe considerar el planteamiento del problema, la justificación, la metodología que se empleará, los recursos con los que se cuenta, así como proponer una hipótesis y reconocer alcances y limitaciones del proyecto, todo esto para la realización de este.

La realización del protocolo es una parte sumamente importante y la cual conlleva un considerable tiempo dedicado el cual podría mermar los alcances que el proyecto tenia considerado en primera instancia.

1.2.1. Justificación

El uso de herramientas para la realización de proyectos de investigación con técnicas de aprendizaje automático no es ajeno, desde el uso de software para recabar datos hasta la utilización de modelos pre-entrenados. Por ese motivo, se desarrollará una aplicación web para el apoyo en la realización de estos.

En un proyecto de investigación, el protocolo es la sección en la cual se define este mismo; planteando el problema que se busca resolver, definiendo los objetivos a cumplir, generando una hipótesis, y reconociendo los alcances y limitaciones del proyecto, por esto, la correcta realización del protocolo puede demorar más de lo previsto.

Methodology for Experiments with Machine Learning Algorithms (MEMLA) es una metodología para la experimentación con técnicas de aprendizaje automático la cual tiene como objetivo el proveer una guía que haga más eficiente el proceso de investigación por medio de 5 fases para cada parte del proceso.

Con la ayuda de la aplicación web de MEMLA, la realización del protocolo para los proyectos de investigación se llevará a cabo de manera más eficaz. Empleando un modelo de inteligencia artificial de procesamiento de lenguaje natural la generación de la hipótesis, los alcances y limitaciones del proyecto se generarán de manera automática en un tiempo inferior al que un humano podría realizarlo.

La aplicación pretende generar la hipótesis, los alcances y las limitaciones de un proyecto de investigación ya que estas son las secciones de este en las que identificaremos la escala y viabilidad del mismo. La hipótesis define el objetivo último de la investigación, se busca la confirmación o negación de esta con evidencia que se obtendrá durante la investigación. Los alcances y las limitaciones suelen ser confundidos, pues ambos establecen la envergadura del proyecto, los alcances determinan los factores a los que se pretende afectar con la investigación, por otra parte, las limitaciones son todas aquellas interferencias que podría sufrir la investigación.

Puede hacerse uso de la aplicación las veces que se desee para generar un protocolo acorde al proyecto deseado o puede usarse como base para construir un protocolo y, una vez que se obtenga el resultado deseado este es entregado en formato pdf.

1.2.1.1. Impacto social

Se busca que la aplicación tenga un fuerte impacto entre los investigadores que trabajan con técnicas de aprendizaje automático por medio de la facilitación de su trabajo.

Se espera recibir a investigadores de otras áreas afines en la aplicación, aún si no trabajan con técnicas de aprendizaje automático, al permitirles generar un protocolo de investigación con enfoque las ciencias.

Finalmente, los estudiantes de licenciatura y posgrados son parte del público esperado. El tiempo con el que cuentan los estudiantes al momento de desarrollar un proyecto de investigación es muy limitado, el uso de la aplicación les permitirá realizar su protocolo de investigación de forma más eficaz.

1.2.1.2. Impacto tecnológico

El desarrollo de una aplicación web capaz de generar un protocolo de proyecto de investigación con base en texto ingresado por el usuario es una implementación de un modelo de lenguaje que no se ha realizado y ofrecería un apoyo para estudiantes o investigadores en su proceso de trabajo.

1.2.1.3. Impacto económico

La investigación y desarrollo siempre conlleva un avance del que puede obtenerse una ganancia económica, con el ahorro en tiempo que la aplicación proporciona se espera un retorno de inversión para los investigadores y compañías beneficiarias.

Igualmente, la inversión en investigación y desarrollo tiene el potencial de impulsar el crecimiento económico, la creación de empleo, la mejora de la productividad y la competitividad internacional de una economía.

1.3. Objetivos

1.3.1. Objetivo general

Implementar un algoritmo de procesamiento de lenguaje natural en una aplicación web (API[1]) para generar protocolos de proyectos de investigación.

1.3.2. Objetivos específicos

- Analizar el estado-del-arte de implementación de algoritmos de procesamiento de lenguaje natural.
- Implementar un algoritmo de procesamiento de lenguaje natural en la aplicación web MEMLA.

[1] Application programming interface

- Generar protocolos de proyectos con el algoritmo de procesamiento de lenguaje natural y una plantilla de contexto.
- Validar los resultados utilizando métricas de aprendizaje automático.

1.4. Hipótesis

Es posible, implementando un algoritmo de procesamiento de lenguaje natural, desarrollar una API que genere de forma semi-automática, protocolos de proyectos de investigación, útiles para investigadores y estudiantes.

1.5. Alcances y limitaciones

1.5.1. Alcances

El proyecto busca que cualquier investigador, desarrollador o estudiante pueda usar la aplicación para generar el protocolo de su proyecto.

1. Generar protocolos de proyecto de investigación con base en texto de entrada y una plantilla de contexto.
2. Generar protocolos de proyectos de investigación con texto sin faltas de ortografía.
3. Generar protocolos de proyectos de investigación con texto congruente a la información ingresada por el usuario.

1.5.2. Limitaciones

1. El hardware podría no ser el correcto para el desarrollo del proyecto.
2. El algoritmo de procesamiento de lenguaje natural podría no ser el correcto para el correcto desempeño del proyecto.
3. El algoritmo de procesamiento de lenguaje natural correcto podría requerir de hardware especializado, como una o varias tarjetas de vídeo con gran memoria RAM.
4. El tiempo de procesamiento del algoritmo podría ser demasiado largo y no ser conveniente para su uso en producción.
5. La propia naturaleza de los algoritmos de procesamiento de lenguaje natural podría no dar un resultado satisfactorio para ciertas situaciones.
6. El algoritmo fue entrenado con texto en inglés y eso podría afectar los resultados en otros idiomas.

Capítulo 2

Marco teórico

2.1. Aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial que tiene como objetivo utilizar algoritmos matemáticos para generar modelos analíticos los cuales son capaces de identificar patrones y tomar decisiones estadísticas.

El surgimiento del aprendizaje automático se debe a la necesidad de hacer que las computadoras resuelvan problemas difíciles de programar con el uso de algoritmos y métodos para descubrir patrones en los datos. Debido a la idea de aprender de los datos, se han reunido técnicas de los campos de la informática, estadística, matemáticas y ciencias. Arthur Samuel describió en 1956 el aprendizaje automático como "la capacidad de las computadoras para aprender sin ser programado explícitamente" y Tom Mitchell en 1997 como "el proceso de enseñar a una computadora a realizar una tarea particular al mejorar su medida de rendimiento con experiencia".

Gracias a los avances en tecnología computacional, el aprendizaje automático ha experimentado cambios significativos en su historia, se originó a partir del reconocimiento de patrones y de la idea de que las computadoras pueden aprender sin requerir una programación explícita para tareas específicas.

La característica iterativa del aprendizaje automático es crucial, ya que los modelos entrenados tienen la capacidad de adaptarse al ser expuestos a nuevos datos. Aprenden de cálculos previos para generar resultados y decisiones confiables y consistentes.

2.2. Aprendizaje automático supervisado

Se requiere de un conjunto de características para predecir el resultado de una variable objetivo mediante la construcción de un modelo de aprendizaje. Las columnas del conjunto de datos se denominan características, variables o atributos del conjunto de datos y las filas se denominan muestras, puntos de datos u observaciones. El objetivo es enseñar a la computadora a aprender los patrones de características del conjunto de datos.

2.2.1. Clasificación

Con frecuencia, los desafíos del aprendizaje automático se asocian con el problema de clasificación, el cual implica asignar una clase a un objeto de la base de datos en función de sus características o atributos.

Para lograr esto, el clasificador requiere definir una función discreta para formar una relación entre los espacios de las clases y los atributos. Dicha función puede definida a partir de los datos de entrenamiento, que son conjuntos de datos previos. Existen varios tipos de clasificadores que representan estas funciones de diferentes maneras, siendo los más comunes los clasificadores bayesianos, los árboles de decisión, las funciones de discriminación y las redes neuronales.

La clasificación se emplea comúnmente para el diagnóstico médico, describiendo las características propias del paciente como su sexo, ubicación del dolor, edad, peso, altura, entre otros. Y el objetivo del clasificador es generar un diagnóstico posible basado en las clases o categorías conocidas, como saludable, gripe o influenza.

2.2.2. Regresión

Los problemas de regresión se definen como aquellos en los que se trabaja con un conjunto de objetos, que son los ejemplos de aprendizaje, y se describen mediante un conjunto de atributos o características que determinan las variables independientes. La variable dependiente es el objetivo de la regresión y consiste en un valor continuo que está determinado por una función de las variables independientes. El objetivo de la predicción por regresión es determinar el valor de la variable continua no observable para una instancia específica que se desea analizar.

En la regresión lineal, por lo general, se utiliza a los atributos como valores continuos y se normalizan adecuadamente. Se asume la existencia de una relación lineal[1] entre las variables dependientes y las independientes. La regresión se encarga de calcular los coeficientes de la función lineal de manera que se minimice el error cuadrático medio.

2.3. Aprendizaje por refuerzo

El aprendizaje por refuerzo implica descubrir cómo asignar acciones a situaciones específicas con el objetivo de maximizar una señal de recompensa numérica. A diferencia de otras formas de aprendizaje automático, no se le indica al alumno qué acciones tomar, sino que debe experimentar y probar diferentes acciones para determinar cuáles generan la mayor recompensa. Este enfoque difiere del aprendizaje supervisado, donde se aprende a partir de ejemplos proporcionados por un supervisor externo experto.

[1]Se refiere a una relación matemática o estadística en la cual los valores de una variable dependiente están directamente relacionados con los valores de una variable independiente a través de una línea recta en un gráfico.

En situaciones interactivas, puede resultar difícil obtener ejemplos precisos y representativos del comportamiento deseado en todas las situaciones en las que el agente debe actuar. Es especialmente beneficioso en territorios desconocidos, donde se espera que el aprendizaje propio sea más efectivo. El aprendizaje por refuerzo ha ganado mucha atención debido a su éxito en la resolución de diversos juegos, a veces superando el rendimiento humano. Un ejemplo notable es "AlphaGo" y "AlphaGo Zero". "AlphaGo" fue entrenado utilizando una gran cantidad de juegos humanos y logró un rendimiento sobrehumano mediante el uso de una red de valores y la búsqueda de árboles de Monte Carlo en su red de políticas.

2.3.1. Clasificador bayesiano ingenuo

La función principal de un clasificador bayesiano consiste en calcular las probabilidades condicionales para todas las posibles clases, dadas las características de una instancia. Este clasificador busca minimizar el error esperado, asumiendo que las características de la clase son independientes. Además, suele trabajarse únicamente con valores discretos, lo que implica la necesidad de discretizar los valores continuos.

Estos clasificadores son modelos utilizados para representar la relación entre variables. Una red bayesiana se compone de nodos y arcos que los conectan. Cada nodo representa una variable o atributo X y tiene asociada una probabilidad P(X). La estructura de la red está definida por los nodos y los arcos entre ellos, mientras que los parámetros de probabilidad son asignados a esta estructura. A partir del teorema de Bayes, es posible calcular la probabilidad de los nodos subsiguientes en la red.

Las redes bayesianas, como método de clasificación, actúan como discriminadores entre las clases al dividir el espacio de características en K regiones de decisión. Estas regiones están asociadas a las clases y se utilizan para determinar a qué clase pertenecen las instancias de entrada. Los ejemplos que se ubican dentro de las regiones correspondientes a una clase son aceptados como pertenecientes a esa clase, mientras que aquellos que no lo hacen son rechazados. De esta manera, las redes bayesianas utilizan la información de las características para realizar la clasificación de manera precisa.

2.3.2. Árbol de decisión

Los árboles de decisión son herramientas de aprendizaje automático que permiten generar reglas lógicas a partir de un extenso conjunto de datos. Estas reglas se organizan en forma de árbol, lo que facilita su comprensión y extracción. Un árbol de decisión consta de múltiples nodos que implementan una función específica con valores de salida discretos. Al ingresar un dato de entrada, se sigue un proceso recursivo a través del árbol, evaluando las ramas y aplicando las reglas definidas por cada nodo. Este proceso comienza en el nodo raíz y se repite hasta llegar a los nodos hoja, donde se determina el valor de salida.

Cada nodo en el árbol establece un criterio de discriminación para los valores de entrada, dividiendo así el espacio de características en regiones más pequeñas que también pueden ser divididas. Existen diversos tipos de árboles de decisión, en los cuales los nodos y las funciones

que se aplican en ellos son calculados de manera diferente. Algunos de los algoritmos más comunes son ID3, C4 y C4.5, desarrollados por John Ross Quinlan [12], junto con otras variaciones propuestas por otros investigadores. Los árboles de decisión utilizados para la clasificación, también conocidos como árboles de clasificación, determinan la pertenencia a una clase específica en sus nodos hoja.

2.3.3. Bosques Aleatorios

Los Bosques Aleatorios son un conjunto de árboles de decisión entrenados con subconjuntos del total de datos de entrenamiento para un mismo problema. De forma que, al combinar los resultados, los errores de unos árboles compensan con los errores de otros se obtiene una predicción más generalizable.

En problemas de clasificación, se emplea comúnmente la técnica de "voto suave" para combinar los resultados de los árboles de decisión. En este enfoque, se otorga mayor importancia a los resultados en los que los árboles estén más seguros de su clasificación.

Por otro lado, en problemas de regresión, se suele utilizar la media aritmética para combinar los resultados de los árboles de decisión. No obstante, también es posible utilizar un enfoque personalizado para combinar las predicciones de un bosque aleatorio. Esto se debe a que se puede acceder a las predicciones individuales de cada árbol, lo que nos brinda la posibilidad de utilizar aquellas predicciones que sean más relevantes o adecuadas para nuestro caso particular.

2.3.4. Máquinas de soporte vectorial

Las máquinas de soporte vectorial (SVM) son algoritmos de aprendizaje automático que se utiliza tanto para la regresión como para la clasificación. Su objetivo principal es encontrar la mejor separación posible entre las clases en un conjunto de datos, incluso en espacios de alta dimensionalidad.

Las SVM buscan un hiperplano que maximice el margen de separación entre las clases. Este hiperplano actúa como una frontera que divide los datos en diferentes categorías.

Los puntos que definen el margen máximo de separación se llaman vectores de soporte. Estos puntos, también conocidos como vectores, representan las instancias de datos en un espacio de n dimensiones, donde n es la cantidad de características o dimensiones presentes en el conjunto de datos. Los vectores de soporte son esenciales para determinar la ubicación y orientación óptima del hiperplano de separación [13].

2.3.5. Vecinos más cercanos

El algoritmo K-NN (vecinos más cercanos) clasifica nuevas instancias asignándolas al grupo que mejor se ajuste en función de la cercanía a sus vecinos. Calcula la distancia entre la nueva instancia y las existentes, ordenando las distancias de menor a mayor para asignar la instancia al grupo más frecuente y con distancias menores.

K-NN es un algoritmo de aprendizaje supervisado que busca clasificar correctamente las nuevas instancias basándose en un conjunto de datos inicial. Este conjunto de datos contiene atributos descriptivos y un atributo objetivo o clase.

A diferencia de otros algoritmos supervisados, K-NN no genera un modelo de aprendizaje durante el entrenamiento, sino que aprende y clasifica en tiempo real al probar los datos de prueba. Es utilizado en minería de datos y tiene aplicaciones clave en el aprendizaje automático, como la creación de sistemas de recomendación.

K-NN es un algoritmo simple que busca clasificar un punto de interés basándose en la mayoría de los datos cercanos a él. Se basa en la observación de vecindarios para determinar la clasificación de una instancia [14].

2.4. Aprendizaje automático no supervisado

El aprendizaje no supervisado intenta determinar lo desconocido, ya que el conjunto de datos no tiene objetivos específicos como en el supervisado. Por lo tanto, el objetivo es construir un modelo que captura la distribución subyacente de los datos. Un algoritmo útil para este proceso es el análisis de componentes principales. El cual comprime una gran cantidad de características para una fácil visualización, mediante la búsqueda se subgrupos de conjunto de datos. Esto se le conoce como agrupamiento [15].

2.5. Aprendizaje profundo

"Los modelos de aprendizaje profundo han mejorado dramáticamente, los cuales se utilizan para el reconocimiento de voz, objetos, detección de objetos entre otros" [16]. Las redes neuronales son algoritmos inspirados por la estructura del cerebro humano. Debido a esto el algoritmo de aprendizaje profundo funciona muy bien en modelos de predicción, enfocados a problemas complejos como la visión por computadora o modelado del lenguaje.

Una subrama del aprendizaje automático es el aprendizaje profundo, el propósito de este campo de estudio es que los modelos generados sean capaces de resolver problemas de la misma forma en la que un ser humano podría resolverlos.

2.6. Redes neuronales

Según Saez [17], el cerebro humano fue la inspiración para el desarrollo de las redes neuronales, se compone de más de cien millones de redes neuronales, las cuales se encargan del procesamiento de la información para que las personas puedan aprender y recordar. Las redes neuronales son sistemas que pueden aprender imitando el funcionamiento del cerebro humano. Una red neuronal posee la siguiente estructura [15]:

- Capa de entrada: Recibe la información de las características del conjunto de datos y la pasa a la siguiente capa oculta.

- Capa oculta: Recibe la información de la capa anterior y realiza cálculos. Puede consistir de múltiples módulos.
- Capa de salida: Recibe la información de la capa anterior y clasificará el resultado. Esta capa contendrá tantas neuronas como clasificaciones haya.

2.6.1. Redes neuronales Recurrentes

Las redes neuronales recurrentes (RNN) son utilizadas en el procesamiento y obtención de información de datos secuenciales. Se utilizan ampliamente en tareas como el modelado del lenguaje, traducción automática, el procesamiento del lenguaje natural (PLN) y el reconocimiento de voz. A diferencia de las redes neuronales artificiales convencionales, que asumen la independencia entre los datos de entrada, las RNN capturan activamente las dependencias secuenciales y temporales.

Una característica distintiva de las RNN es el hecho de compartir parámetros. Al compartir parámetros, el modelo asigna los mismos parámetros para representar cada dato en una secuencia, lo que le permite realizar inferencias en secuencias de longitud variable. Esto es especialmente importante en el procesamiento del lenguaje natural. Una red multicapa tradicional fallaría al interpretar el lenguaje, ya que asignaría parámetros únicos para cada posición (palabra) en una frase. En cambio, las RNN son más adecuadas para esta tarea, ya que comparten pesos entre los datos secuenciales, como las palabras en una frase.

Las RNN se caracterizan por tener ciclos que conectan nodos adyacentes o pasos de tiempo, lo que les permite tener memoria interna para evaluar las propiedades del dato actual en relación con los datos del pasado inmediato. Además, a diferencia de las redes neuronales convencionales de tipo perceptrón que se limitan a un mapeo uno a uno entre la entrada y la salida, las RNN pueden realizar mapeos de uno a muchos, de muchos a muchos (por ejemplo, traducción) y de muchos a uno (por ejemplo, reconocimiento de voz). Se utiliza un grafo computacional para representar los mapeos entre las entradas y las salidas, así como para calcular la pérdida del modelo.

2.6.2. Procesamiento de lenguaje natural

El procesamiento del lenguaje natural (PLN) es un campo del aprendizaje automático centrado en darle la capacidad a una computadora de entender el lenguaje humano escrito o hablado igual que a una persona. En este ámbito también se encuentra la generación de lenguaje que sea lo más fiel al lenguaje humano, tanto formal como informal.

Este campo tiene como principal problemática el hecho de que el lenguaje es inherentemente ambiguo en distintos niveles, léxico, referencia, estructural y pragmático, y cada uno de estos niveles debe ser procesado correctamente para que el resultado final tenga sentido para un humano.

En 1950 Alan Turing publicó su artículo *Computing machinery and intelligence* [18] en el cual proponía una prueba de inteligencia para las computadoras con la cual se evalúa si

dicha maquina es capaz de imitar la inteligencia humana, hoy día todavía es utilizado para evaluar algoritmos de PLN.

El PLN hace uso de RNN para procesar corpus de texto asignándole un valor, llamado peso, a cada palabra con relación a la palabra anterior de forma que se generen relaciones entre palabras que permiten predecir la palabra siguiente más probable a una sucesión de palabras.

2.6.3. Generación de texto

Se trata de una tarea realizada por un modelo de PLN el cual, dada una entrada inicial de texto, es capaz de generar secuencias de texto. El resultado se puede ver influenciado tanto por el entrenamiento del propio modelo como por el input inicial.

El objetivo de esta tarea es la de generar textos indistinguibles a un texto escrito por un ser humano, tanto en un nivel léxico, gramático, sintáctico y de contenido.

2.6.4. Modelos pre-entrenados

El resultado de usar una técnica de aprendizaje automático con un conjunto de datos es un modelo entrenado, dicho modelo es capaz de implementarse en una aplicación con la finalidad de resolver una tarea en concreto.

Por el internet existen páginas web en las cuales se recaban cientos o miles de modelos entrenados, gracias a dichas paginas es posible obtener modelos sin el consumo de tiempo que significa entrenarlos y emplearlos en nuestros proyectos, Sin embargo, utilizar modelos que no conocemos del todo no es el mejor curso de acción para el desarrollo de un proyecto de investigación, este tipo de modelos son útiles por otros motivos como puntos de comparación o refinarlos.

2.6.4.1. Bidirectional Encoder Representations from Transformers

BERT es un modelo de lenguaje basado en redes neuronales pre-entrenadas que utiliza la arquitectura de transformers. Desarrollado por Google, su nombre significa representaciones encoder bidireccionales de transformadores. BERT se entrena utilizando una gran cantidad de datos de texto y se puede adaptar para diversas tareas de procesamiento de lenguaje natural, como el reconocimiento de entidades nombradas, la respuesta a preguntas y la clasificación de texto.

Aunque existen modelos más actuales que han superado a BERT en rendimiento o calidad de resultados generales, BERT sigue siendo el más utilizado en la industria por sus excelentes resultados para tareas específicas.

2.6.5. Grandes modelos de lenguaje

Es un término recientemente acuñado para los modelos de lenguaje pre-entrenados que constan de cientos de miles de millones de parámetros, ya que al escalar a tan altas magnitu-

des los modelos empiezan a emerger habilidades no previstas para otros modelos de lenguaje, como la capacidad de traducción, comprensión del contexto y seguimiento de instrucciones [19].

2.6.5.1. Generative pre-Trained transformer

GPT, se trata de un gran modelo de lenguaje cuyo propósito es resolver problemas de lenguaje natural, como comprensión de texto, traducción, escritura automática, entendimiento de lenguaje y procesamiento de lenguaje natural. Está diseñado para ser entrenado en una gran cantidad de datos y es capaz de generar contenido de calidad. Existiendo múltiples versiones, la versión 3 cuenta con 175 mil millones de parámetros.

2.6.5.2. Megatron-Turing

Megatron-Turing es un modelo de lenguaje en inglés sumamente poderoso con 530 mil millones de parámetros. Este modelo de 105 capas mejora los modelos previos del estado-del-arte en configuraciones de cero, uno y pocas pruebas. Demuestra una precisión sin igual en una amplia gama de tareas de lenguaje natural, como predicción de texto, comprensión de lectura, razonamiento de sentido común, inferencias de lenguaje natural, certeza de significado de palabras, etc.

2.6.6. Transferencia de aprendizaje

Una vez que se tiene un modelo entrenado es posible hacer un refinamiento a dicho modelo o la creación de uno completamente nuevo transfiriendo el aprendizaje previo. El refinamiento se realiza entrenando nuevamente al modelo con información más específica a la tarea que se desea solucionar. Por ejemplo, el modelo GPT-2 es capaz de realizar diferentes tareas de procesamiento del lenguaje natural como traducción de texto, generación de texto, generación de código, chatbot, entre otras, con un refinamiento es posible convertir el modelo en uno que solo realice una de estas tareas, pero de forma mucho más competente.

Un problema existente al realizar este proceso es que el modelo original se haya entrenado con una cantidad de información mucho mayor a la que se pretende usar para el refinamiento, por lo que es posible que el resultado sea un modelo que realice la tarea de forma menos competente.

2.6.7. Ingeniería de peticiones

Una de las nuevas habilidades desarrolladas por los ingenieros con la popularización de los modelos de lenguaje y los generadores de arte con entradas de texto es la ingeniería de peticiones, mayormente conocida en inglés como *Prompt enginnering*, en la cual el ingeniero debe crear un contexto suficientemente conciso para que el modelo genere el resultado deseado ajustando únicamente el texto de entrada para el modelo [20].

Como observaron Dai *et al.*, [21] en muchas ocasiones el desarrollo de una buena petición con el contexto correcto puede dar resultados superiores al ajuste fino del propio modelo de lenguaje.

2.6.8. Meta-aprendizaje

Uno de los temas más interesantes de los grandes modelos de lenguaje es la capacidad de aprender de contenido generado por estos mismos. Con base en un prompt inicial básico es posible generar ejemplos de tareas con las cuales puede entrenarse una versión ajustada del modelo para esa tarea. Esto resulta especialmente útil cuando no se cuenta con un dataset amplio con el cual ajustar al modelo [22].

2.6.9. Alucinaciones

La característica más controversial de los modelos de lenguaje es su capacidad de generar información no verídica, a esto se le conoce como alucinación. Esto es debido a la propia naturaleza de estos modelos de generar una respuesta con base a probabilidades y no ha contenido citado, a su vez, es posible que el usuario guíe intencionalmente al modelo a generar una respuesta [23].

Debido a esta particularidad puede llegar a ocurrir un efecto de bola de nieve cuando el modelo ha generado una alucinación dentro de una respuesta y posteriormente usa esta misma respuesta como base para una siguiente respuesta, tomando como verídico la alucinación inicial y generando aún más texto no verídico [24].

Sin embargo, esta cualidad no debe considerarse perjudicial en todos los escenarios, ya que hay ocasiones en las que el usuario desea obtener una respuesta con base a un concepto previamente desconocido, como podría ser el caso de la escritura de una historia o la generación de conocimiento nuevo.

2.7. Algoritmo de procesamiento de lenguaje natural

El algoritmo de generación de texto empleado en este proyecto es la GPT2-xl, creado y liberado en noviembre del 2019 por la compañía OpenIA, este es un masivo modelo del lenguaje pre-entrenado utilizando textos mayormente en inglés que cuenta con 1.7 mil millones de parámetros [25].

La arquitectura que compone el modelo es llamada *Transformer*, la Figura 2.1 muestra el diagrama de la arquitectura, esta es una arquitectura altamente paralelizable basada en mecanismos de atención, por lo cual es sumamente eficiente al codificar secuencia de palabras por medio de ecuaciones senoidales en lugar de utilizar vectores one-hot como otras arquitecturas de redes neuronales enfocadas en el procesamiento del lenguaje [26].

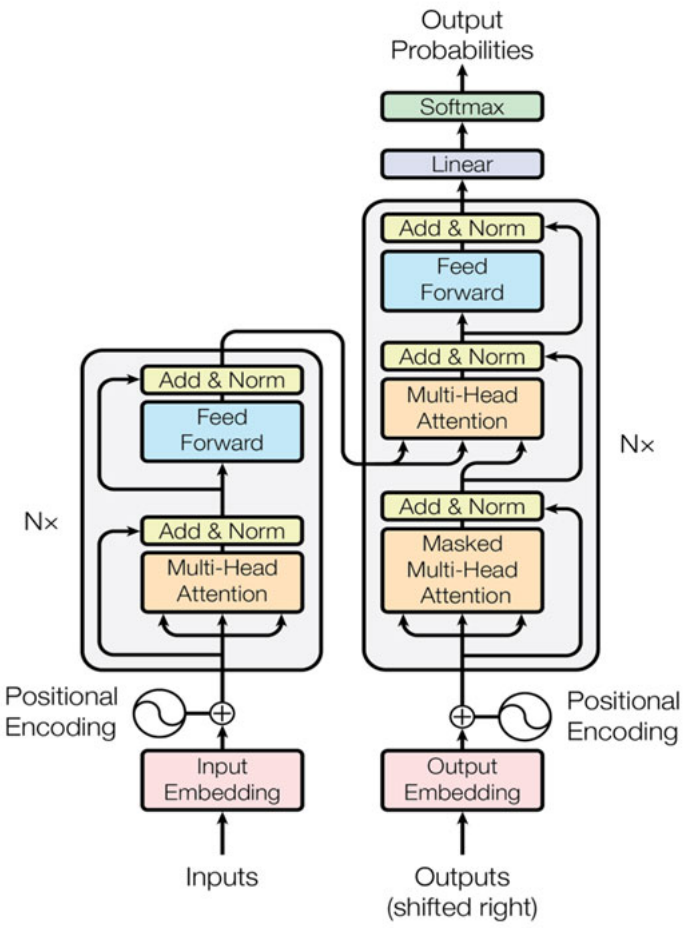


Figura 2.1: Arquitectura transformer [27]

2.7.0.1. Dataset

El modelo GPT-2 fue entrenado con un conjunto de datos que consta de texto extraído de aproximadamente 8 millones de páginas web distintas. Esto fue posible gracias al uso de técnicas de web scraping sobre enlaces recopilados de la red social Reddit, dichos enlaces fueron seleccionados si cumplían con el requisito de poseer 3 puntos de "karma"(puntaje usado en la red social) con el fin de identificar aquellos enlaces que los usuarios encontraron útiles.

2.7.0.2. Configuración implementada

Para el proyecto se utilizó la configuración por defecto del algoritmo GPT2-xl, ya que este cuenta con las características necesarias para generar texto congruente de acuerdo a las instrucciones ingresadas.

El modelo tiene un total de 12 capas en su arquitectura de transformador. Cada capa

consta de mecanismos de auto-atención de múltiples cabezas y redes neuronales de avance.

La "ventana de contexto" en GPT-2 se refiere al número de tokens anteriores que el modelo tiene en cuenta para generar el siguiente token en el texto. El tamaño predeterminado de la ventana de contexto es de 1024 tokens, aproximadamente 650 palabras.

GPT-2 puede ser afinado en tareas específicas con conjuntos de datos más pequeños después de ser pre-entrenado en un corpus grande. Este enfoque de transferencia de aprendizaje le permite adaptarse a varias aplicaciones, como traducción de idiomas, resumen de texto, preguntas y respuestas, etc.

Si bien el afinamiento del modelo es poderoso, no siempre es posible o práctico realizar el afinamiento para cada tarea nueva o consulta específica. En lugar de eso, se puede mejorar la calidad de las respuestas al redactar una petición específica que sea más clara y precisa. La redacción de la solicitud puede incluir instrucciones más detalladas, restricciones específicas o ejemplos que guíen al modelo hacia la respuesta deseada. Con una solicitud bien formulada, incluso un modelo de lenguaje general sin afinar puede generar respuestas más adecuadas y relevantes.

2.7.0.3. Algoritmo GPT-2

El desarrollo del modelo GPT-2 se realizó siguiendo el algoritmo descrito a continuación1. Se utiliza un *encoder* para convertir el texto de entrada en números, así como un *decoder* para realizar el proceso inverso.

Algoritmo 1 Algoritmo de procesamiento de lenguaje natural

```
Inicio;
Iniciar parámetros del modelo
iniciar secuencia de entrada
iniciar secuencia de salida
#Encoder
para Capa en capas_encoder hacer
    secuencia_entrada = capas(secuencia_entrada)
fin para
#Decoder
para capa en capas_decoder hacer
    secuencia_salida = capa(secuencia_salida)
fin para
#Ciclo de entrenamiento
para epoca en rango(numero_epocas) hacer
    #pase hacia delante
    entrada_decode = encoder(secuencia_entrada)
    salida_decode = decoder(secuencia_salida)
    #Cálculo de pérdida
    perdida = calculo_perdida(salida_decode, salida_objetivo)
    #Propagación hacia atrás
    backward(perdida)
    #Actualizar parámetros del modelo
    actualizar_parametros()
fin para
para i en rango(num_token_generado): hacer
    #Pase hacia delante
    entrada_encoder = encoder(secuencia_entrada)
    token_generado = decoder.generar_token(entrada_encoder)
    #Concatenar token generado a la secuencia de salida
    secuencia_salida.concatenar(token_generado)
fin para
#Salida final
retornar secuencia_salida
Fin;
```

Capítulo 3

Estado-del-arte

En esta sección se presentan múltiples trabajos relacionados al tema de este documento, desde el uso de metodologías para el desarrollo de proyectos hasta otras plataformas que desempeñan funciones similares a las propuestas en nuestro trabajo.

3.1. Estrategias de desarrollo de software para aprendizaje automático

Gupta *et al.*, proponen el modelo I-MIN para el descubrimiento y manejo de conocimiento en bases de datos centrado en el usuario, el cual emula la organización de un sistema gestor de bases de datos para los administradores y usuarios finales. Este modelo fue implementado en [28] y fue probado experimentalmente con el uso del dataset Iris. El sistema I-MIN refuerza un acercamiento iterativo e interactivo para los usuarios, lo que facilita su implementación en proyectos dinámicos enfocados en la exploración, la arquitectura permite el manejo, conservación, renovación y compartición del conocimiento para su correcta administración [29].

Sharma & Osei-Bryson, identificaron que los modelos de procesamiento actuales para el descubrimiento de conocimiento y la minería de datos constan de un diseño fragmentado, una lista de tareas separadas que carecen de apoyo unas con las otras. *The Integrated Knowledge Discovery and Data Mining process model* identifica los problemas antes mencionados y provee soporte de ejecución entre las tareas propuestas por el modelo predecesor. Evaluaron su efectividad y eficacia contra el modelo *Cross Industry Standard Process for Data Mining* [30].

Coleman *et al.*, señalaron en su artículo que el mundo del big data no necesariamente es exclusivo para las compañías más grandes, sin embargo, las limitaciones y dificultades que se deben enfrentar son numerosas. Las pequeñas y medianas empresas son una parte vital para la economía de todos los países y estas mismas también deberían tener a su alcance el acceso a técnicas de trabajo con grandes volúmenes de datos ya que existe hardware y software a precios asequibles. Como bien se puntúa en el artículo, las pequeñas y medianas

empresas deben lidiar con problemas como lo son la falta de experiencia con técnicas y herramientas, seguridad del sistema, el desarrollo de habilidades propias, disponibilidad de la información y costos de mantenimiento. Los autores proponen la maduración de la empresa por medio de mejorar puntos clave como la estrategia de negocios, el manejo de datos, existencia de personal especializado, la infraestructura tecnológica, entre otros. Tras realizar dichas mejoras es posible implementar una metodología de trabajo con big data enfocada a sus necesidades [31].

3.1.1. Mecanismos de atención

En 2014, Dzmitry Bahdanau *et al.*, presentaron los mecanismos de atención como una extensión para mejorar el rendimiento del modelo Encoder-Decoder en la tarea de traducción automática [32]. En los modelos Encoder-Decoder convencionales, el encoder captura toda la información de la frase de entrada en un único vector de tamaño fijo llamado estado oculto final. Sin embargo, el rendimiento de estos modelos se deteriora rápidamente a medida que aumenta la longitud de la cadena de entrada, según mencionan los propios autores.

Los mecanismos de atención se introdujeron para abordar precisamente este problema. Esta técnica permite generar un conjunto de vectores en lugar de un solo vector de dimensión fija. En cada paso de tiempo, se selecciona un subconjunto de estos vectores que están cerca del término que se desea traducir. De esta manera, los mecanismos de atención mejoran la capacidad del modelo para capturar información relevante de toda la secuencia de entrada, incluso en casos de frases más largas.

Existen varias propuestas basadas en este mecanismo:

3.1.1.1. Atención visual

Fue introducida en 2015 por Xuet *et al.*, [33] con el objetivo de desarrollar una aplicación capaz de describir una imagen de entrada generando una palabra correspondiente como salida. La arquitectura del sistema consistía en una red convolucional para extraer características de la imagen, seguida de una red recurrente con mecanismos de atención que generaba la palabra de salida. El mecanismo de atención permitía enfocarse en elementos específicos de la imagen que la definían, de manera que la frase generada se correspondiera con dichos elementos.

3.1.1.2. Atención jerárquica

En 2016, se presentó la atención jerárquica por Yang *et al.*, [34] esta técnica se aplicó en un sistema de clasificación de documentos y utilizaba mecanismos de atención en dos niveles. La arquitectura incluía dos módulos de encoder-decoder + atención, uno a nivel de frase y otro a nivel de palabra.

3.1.1.3. Transformer

Uno de los trabajos más interesantes resultando de los mecanismos de atención fue creación de los modelos transformer, propuestos por Vaswani *et al.*, en 2017 [27]. Esta arquitectura supuso un nuevo nivel en el campo del Procesamiento del Lenguaje Natural ya que su alta paralelización lo hace sumamente eficiente al codificar secuencia de palabras por medio de ecuaciones senoidales, en lugar de utilizar vectores one-hot como otras arquitecturas de redes neuronales enfocadas en el procesamiento de datos secuenciales.

3.2. Implementación de procesamiento de lenguaje natural

Tal como mencionan Kais *et al.*, existen diferentes propuestas para realizar de forma semiautomatizada la identificación y análisis de datos. El propósito de productos o prototipos tales como *The Automatic Statistician, Data Science Machine, Autodiscovery (Butler Scientifics), or Wordsmith (Automated Insights)* es recibir datos crudos de entrada y ordenarla para fines de análisis estadístico. Parte de dicha semi-automatización es la generación automática de reportes de resultados en formatos fáciles de leer por medio de NLP [35].

Nakano *et al.*, crearon un ambiente web para interactuar con su modelo pre-entrenado de NLP WebGPT, en el cual es posible realizarle preguntas *long-form question answering* de las que se espera que el modelo responda realizando búsquedas a través de la web y recuperando información relevante. Se espera que este tipo de sistemas sea la forma en que en un futuro podamos aprender de cualquier tema de forma más eficiente. Este modelo fue entrenado utilizando el conjunto de datos ELI5 [36] que consta de preguntas open-ended recopiladas del subreddit "Explain Like I'm Five". El modelo fue probado utilizando TruthfulQA [37], un punto de referencia para verificar la capacidad de una respuesta simular la expresión humana, dichas respuestas fueron calificadas por su veracidad y su informatividad, rondando entre el 60 % y 80 % de veracidad, pero solo acercándose al 60 % de informatividad [38].

Zhang *et al.*, publicaron recientemente un modelo pre-entrenado de NLP, OPT-175B, el cual cuenta con 175 mil millones de parámetros entrenados. La publicación no solo es del modelo sino también del código necesario para entrenar nuevos modelos con corpus de texto diferentes. Si bien este modelo cuenta con la misma cantidad de parámetros que el modelo GPT-3, el publicado por META utiliza la arquitectura Transformer lo cual lo hace mucho más rápido de entrenar en comparación con una red neuronal recurrente [39].

Touvron *et al.*, publicaron modelos de lenguaje con rangos de parámetros entre los 7 y 65 mil millones llamados LLaMA, este modelo se desarrolló con la intención de demostrar que se puede alcanzar un nivel de estado-del-arte utilizando únicamente datos públicos y con un número menor de parámetros que otros modelos actuales (GPT-3 175 mil millones, Chinchilla 70 mil millones y PaLM 540 mil millones). Los modelos LLaMA fueron liberados como código abierto a principios de 2023 [40].

Ali *et al.* proponen el uso de ChatGPT para la generación de cartas clínicas con el fin de realizar el registro y comunicación de información clínica importante de forma eficiente. Esta propuesta es reciente y tras intentar implementar otros medios como el reconocimiento

de voz o el uso de plantillas para las cartas, demostrando la implementación de modelos de lenguaje apunta a ser la solución a problemas de semi-automatización de procesos [41].

SmartPaper.ia es una plataforma web para la redacción de proyectos de investigación científica y académica con ayuda de inteligencia artificial. Implementando modelos comerciales como ChatGPT y Bingchat, SmartPaper genera secciones del protocolo de proyecto como la hipótesis o los objetivos. Sin embargo, esta plataforma es de pago y utiliza modelos de terceros de código cerrado [42].

3.3. Programación con peticiones

White *et al.*, definen a las peticiones, o prompts, como la forma en la podemos programar a un gran modelo de lenguaje para devolver los resultados que deseamos, como bien demuestran en su artículo programando a ChatGPT para preguntar información a cerca de desplegar una aplicación python con servicios AWS y finalmente generar un script. Todo esto con tan solo unas lineas de texto, sin modificar parametros ni reentranando al modelo [43].

La ingeniería de peticiones ha demostrado ser óptima para obtener todo tipo de resultados, Wang *et al.* recopilaron ejemplos de como la petición correcta puede resolver problemas de clasificación, detección, generación, inferencia y resolución de dudas, todo en el ámbito médico [44].

3.4. Plataformas de proyectos

En 2015, Metascape team publicó al público una herramienta la cual facilita la anotación y análisis de genes con la finalidad de ayudar a los biólogos a construir listas ordenadas de genes para su estudio, dicha herramienta conocida como Metascape. En tan solo un año la web ya había procesado 70,000 listas de genes diferentes, apoyando el trabajo de investigadores que realizaron 241 publicaciones. Este sistema es un ejemplo de como es posible mejorar la calidad de vida para los investigadores de cualquier campo de estudio con el desarrollo de herramientas adecuadas [45].

Shuster *et al.*, publicaron en Agosto de 2022 Blenderbot 3 un modelo de dialogo de 175B de parámetros. Las características novedosas de este chatbot es que cuenta con acceso a internet, su ajuste basado en dialogo y la publicación de sus pesos, código y dataset conversacional. Es posible interactuar con el modelo a través de la página web de Meta, únicamente en los Estados Unidos. Sin embargo, el enfoque conversacional del modelo lo hace subóptimo para otras tareas [46].

En febrero de 2023, Microsoft implemento en su buscador Bing a ChatGPT nombrando a su nueva forma de trabajo como *modelo Prometeus*, esto, con el objetivo de mejorar sus búsquedas y experiencias al usuario, Microsoft expone la evolución de Bing como la culminación de cuatro avances técnicos; un modelo de OpenAI de ultima generación, modelo Prometeo de Microsoft, aplicación de IA en algoritmos de búsqueda y la mejora de la experiencia al

usuario, por el momento se requiere acceder a una lista de espera para poder realizar uso del chat de Bing [47].

ChatGPT es considerada una inteligencia artificial revolucionaria, capaz de trabajar con lenguaje natural generando respuestas a una petición o entrada determinada mediante texto. Su libre acceso ha permitido que usuarios de todo tipo le den uso para resolver una infinidad de tareas distintas; conversación, traducción, creación de contenido, resumir texto, escribir musica o poesía, entre muchas otras. Específicamente en el ámbito escolar, ChatGPT es empleado por estudiantes que necesitan un texto simplificado de un tema para comprenderlo mejor y poder reiterar sobre sus respuestas para expandir su conocimiento. Igualmente, ha apoyado a investigadores a procesar grandes volúmenes de datos por su capacidad de filtrar temas y crear subconjuntos de datos [48]. Sin embargo, su uso, funcionamiento y disponibilidad en su versión gratuita depende totalmente de la carga de trabajo de sus servidores.

Sin embargo, estas tecnologías no satisfacen las necesidades que se pretende cubrir con esta investigación, si bien el trabajo que se presenta a continuación esta basado en el modelo GPT-2, el cual se integra a la aplicación mediante una API y utilizando una *plantilla de contexto*, se busca la generación de la hipótesis, justificación y alcances de un proyecto de investigación con el uso de NLP.

3.5. Acercamientos mixtos

Moyle & Alípio, propusieron RAMSYS una metodología colaborativa para el trabajo remoto en proyectos de minería de datos, dicha metodología tiene como propósito optimizar el desarrollo de proyectos con múltiples equipos de trabajo. A su vez, se presenta un sistema web el cual implementa la filosofía RAMSYS, dicho sistema soluciona posibles dificultades que se presentan al emplear únicamente la metodología propuesta [49].

Crowston *et al.*, reconocieron una necesidad en el campo de la ciencia de datos de una guía de trabajo conjunto. Si bien el desarrollo de algoritmos de machine learning y el estudio de los datos recopilados es parte fundamental de un proyecto de ciencia de datos, también lo son aquellas personas que realizan ese proyecto. Por lo cual, plantearon un marco de trabajo como soporte a la coordinación por un medio físico (estigmérgica) y la implementación de una herramienta web (MIDST) para el desarrollo de proyectos de ciencia de datos. En 2018 se realizó una prueba de la metodología y la herramienta MIDST con 72 estudiantes separados en 10 grupos de trabajo, tras una encuesta se encontró que el sistema MIDST permite una mejor modularidad del código y favoreciendo la productividad con grupos de trabajo que se encuentran separados físicamente, lo cual es necesario para contar con un ambiente de trabajo coordinado estigmérgicamente [50].

3.6. Análisis y nicho de oportunidad

En la Tabla 3.1 se muestran las principales contribuciones de los trabajos relacionados y la comparación con el presente estudio. Las metodologías de trabajo han sido un factor importante en el desarrollo de proyectos de investigación cada vez mejores y más ordenados, por lo cual es importante proveer una forma de mantener ese orden de trabajo. A pesar del progreso que ha existido en las herramientas, aún no existe una herramienta que permita construir protocolos de proyectos, por lo que, nuestra propuesta implementa NLP para semiautomatizar la generación del protocolo de investigación a partir de información como el título, la problemática, justificación, los recursos disponibles y los objetivos, estos datos son recolectados por el framework web y analizados por la arquitectura GPT-2. A su vez, existen herramientas tales como SCIgen a las cuales se les ha dado un mal uso generando tesis o artículos en su totalidad sin la correcta verificación o corrección. Nosotros no pretendemos automatizar en su totalidad el proceso de investigación, así como, todavía es necesario que el usuario haya definido previamente varios puntos de su proyecto, como lo son la problemática, justificación y objetivos.

Tabla 3.1: Comparación con trabajos relacionados.

Tipo de estudio	**Año**	**Contribuciones sobresalientes**	**Referencia**
Estrategias de desarrollo de software para aprendizaje automático	2005	Propone el modelo I-MIN con un acercamiento iterativo e interactivo para usuarios que facilite la implementación de proyectos dinámicos.	[29]
	2008	Propone IKDDP, el cual provee apoyo para unir tareas del proyecto.	[30]
	2016	Sugiere la maduración de la compañía mejorando puntos clave como la estrategia de negocios, manejo de datos, disponibilidad de personal especializado, infraestructura tecnológica, entre otros.	[31]
Mecanismos de atención	2014	Presentan los mecanismos de atención como extensión para mejorar el modelo Encoder-Decoder para la tarea de traducción.	[32]
	2015	Se emplean mecanismos de atención en combinación con redes neuronales recurrentes para describir el contenido de una imagen con texto.	[33]

Continua en la siguiente página

Tabla 3.1 – continuación de la página anterior

Tipo de estudio	Año	Contribuciones sobresalientes	Referencia
	2016	Se aplicaron mecanismos de atención en 2 niveles para la clasificación de documentos de acuerdo a su contenido.	[34]
	2017	Se presenta la nueva arquitectura transformer para el procesamiento de lenguaje natural, basada únicamente en mecanismos de atención.	[27]
Implementación de procesamiento de lenguaje natural	2017	Analizar propuestas de informes automáticos de resultados en formatos fáciles de leer usando NLP.	[35]
	2021	Crearon un ambiente web interactivo con el modelo pre-entrenado WebGPT.	[38]
	2022	Presentaron un modelo pre-entrenado de NLP llamado OPT-175B.	[39]
	2023	Publicaron un nuevo modelo pre-entrenado únicamente con datos públicos con resultados mejores al estado-del-arte.	[40]
	2023	Proponen el uso de ChatGPT para la generación de cartas clínicas.	[41]
	2023	Implementan algoritmos de lenguaje natural comerciales para generar proyectos de investigación.	[42]
Programación con peticiones	2023	Definieron la programación de modelos de lenguaje por medio de texto como peticiones o *prompts*.	[43]
	2023	Demostraron que por medio de las correcta ingeniería de peticiones un mismo modelo de lenguaje puede resolver una gran cantidad de tareas diferentes.	[44]
Plataformas de proyectos	2019	Presenta la herramienta Metascape que facilita la notación y análisis de genes.	[45]
	2022	Publicacion de un modelo de dialogo de 175mil millones de parámetros, con acceso a internet y de codigo abierto.	[46]

Continua en la siguiente página

Tabla 3.1 – continuación de la página anterior

Tipo de estudio	Año	Contribuciones sobresalientes	Referencia
	2023	Microsoft implemento ChatGPT en su navegador Bing para mejorar sus predicciones de busquedas y ayudar a los usuarios a filtrar información.	[47]
	2023	Recopilaron información a cerca del uso de ChatGPT por los usuarios, demostrando su gran capacidad para resolver tareas en muchos campos de estudio.	[48]
Acercamientos mixtos	2001	Propone la metodología RAMSYS y desarrollo un sistema web para trabajo remoto.	[49]
	2019	Propone un framework para apoyar la coordinación e implementación de la herramienta web MIDST para el desarrollo de proyectos de ciencia de datos.	[50]
	2023	Se propone una aplicación web con el objetivo de semi-automatizar el desarrollo de protocolo de proyectos de investigación con un algoritmo de procesamiento de lenguaje natural.	Nuestro

Capítulo 4

Experimentos y análisis de resultados

4.1. Metodología

En esta sección se describen aquellos materiales, herramientas, métodos y procesos empleados en la realización de la aplicación, los cuales constan de 4 fases: Selección del modelo de lenguaje 4.1.1,implementación del modelo de lenguaje 4.1.2, comunicación con la aplicación web 4.1.2.1 y pruebas y validación. La Figura 4.1 muestra el proceso de desarrollo.

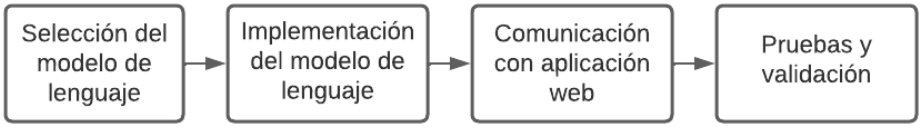


Figura 4.1: Fases de desarrollo del módulo

4.1.1. Fase 1. Selección del algoritmo de procesamiento de lenguaje natural

El primer paso fue la selección de un algoritmo de procesamiento de lenguaje natural para la generación de texto, existen múltiples modelos pre-entrenados disponibles para su uso libre en páginas webs como Hugging Face [51], el modelo seleccionado fue GPT2 en su versión xl [25] ya que cuenta con el mayor numero de parámetros, entre sus versiones, y un amplia ventana de contexto retornando mejores resultados.

4.1.1.1. Modelo GPT2

El algoritmo de generación de texto empleado en este proyecto es la GPT2-xl, creado y liberado en noviembre del 2019 por la compañía OpenIA, este es un masivo modelo del lenguaje pre-entrenado utilizando textos mayormente en inglés. Cuenta con 1.7 mil millones de parámetros [25].

En comparación con otros algoritmo de procesamiento de lenguaje natural, GPT2 supera a BERT en su numero de parámetros (340 millones) y en su capacidad para generalizar información. RoBERTa [10] es un modelo refinado de BERT especializado en la clasificación de textos, una habilidad no requerida para este proyecto.

4.1.2. Fase 2. Implementación de algoritmo de procesamiento de lenguaje natural en API

Una *application programming interface* (API) es un conjunto de subrutinas, funciones y/o procedimientos que ofrece una biblioteca para ser utilizada por otro software [52], para nuestro caso, la API se compone por métodos que reciben los datos ingresados por el usuario y se procesan para devolver un resultado generado por el modelo de inteligencia artificial.

La aplicación fue desarrollada con la finalidad de semi-automatizar el proceso de creación del protocolo para un proyecto de investigación. Como solución se plantea una aplicación web donde el usuario capture datos que sirvan como base para la generación del protocolo, estos datos son enviados a un servidor en el cual se integran los datos capturados a una plantilla de contexto[1] y enviar esta plantilla como petición al algoritmo GPT-2 en su versión *"xl"* 2, una vez que se obtiene un resultado este es devuelto la aplicación web, la arquitectura puede observarse en la Figura 4.2.

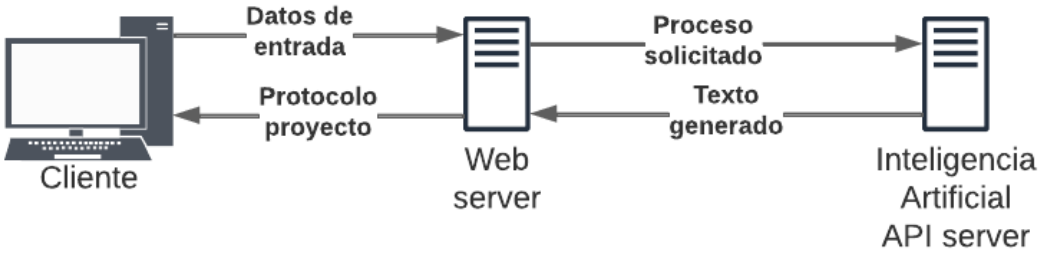


Figura 4.2: Arquitectura de la API

4.1.2.1. Desarrollo del módulo de procesamiento de lenguaje natural

El módulo opera siguiendo la siguiente estructura 2; Una vez que el usuario ingresa los datos relacionados a su proyecto, título de proyecto, problemática, justificación y objetivo general, iniciara el proceso.

Se concatenará el título de proyecto a la plantilla de contexto, así como la problemática, justificación y el objetivo general; posteriormente la plantilla será ingresada al modelo GPT2 como una petición para que genere texto; se generará la hipótesis, se concatena la hipótesis

[1]La plantilla de contexto puede encontrarse en el apéndice B.

a la plantilla de contexto; se ingresa la nueva plantilla al modelo; se generan los alcances del proyecto y se concatenan a la plantilla de contexto; nuevamente, se ingresa la plantilla de contexto al modelo; se generan las limitaciones del proyecto; finalmente, se retorna el resultado final.

Algoritmo 2 Operación del módulo

Entrada: Título del proyecto, Problemática, Justificación, Objetivo general
Salida: Hipótesis, Alcances, Limitaciones

```
Inicio;
Plantilla de contexto ← Título del proyecto;
Plantilla de contexto ← Problemática;
Plantilla de contexto ← Justificación;
Plantilla de contexto ← Objetivo general;
GPT2 ← Plantilla de contexto;
    Se genera hipótesis
Plantilla de contexto ← Hipótesis;
GPT2 ← Plantilla de contexto;
    Se generan alcances
Plantilla de contexto ← Alcances;
GPT2 ← Plantilla de contexto;
    Se generan limitaciones
Mostrar resultados en aplicación web
Fin;
```

4.1.3. Fase 3. Integración de la API y modelo NLP

Una vez que el módulo de procesamiento de lenguaje es operativo se dispone a recibir los datos provenientes de la aplicación web. Dicha aplicación consta de las siguientes secciones.

En la pantalla de selección del tema de estudio (ver Figura 4.3), el primer dato solicitado al usuario es el título del proyecto que esté realizando. El título debe ser lo más explicativo posible. Subsecuentemente, se debe ingresar tanto la problemática como la justificación del proyecto. Estos datos describen casi completamente al proyecto que se pretende elaborar y son utilizados para la generación del apartado *Project demarcation.*

A continuación, se le realizará un pequeño cuestionario al usuario (ver Figura 4.4), el cual tiene como objetivo identificar el tipo de proyecto que se está realizando, con esto se define el algoritmo de aprendizaje automático que se pretende utilizar.

Posteriormente se deben definir los recursos con los que cuenta el proyecto, como se ve en la Figura 4.5, el tiempo que se tiene disponible para la totalidad del proyecto y el número de integrantes del equipo de trabajo, esto con la finalidad de generar un cronograma de actividades lo más realista posible acorde a las especificaciones dadas.

Una de las partes más importantes del protocolo es la definición de objetivos, en la Figura 4.6 se muestra la pantalla correspondiente, se debe escribir el objetivo principal del proyecto

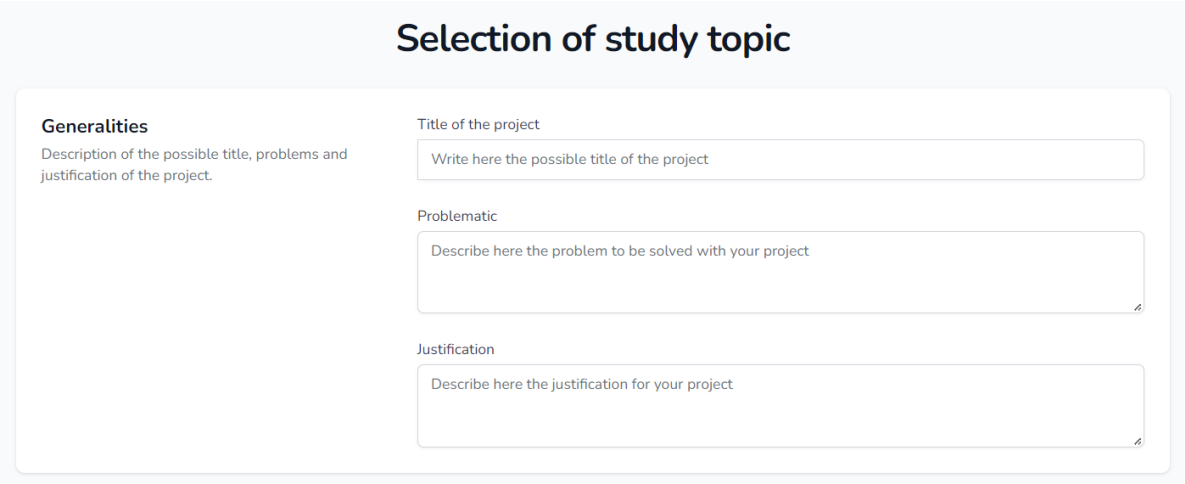


Figura 4.3: Sección de título, problemática y justificación

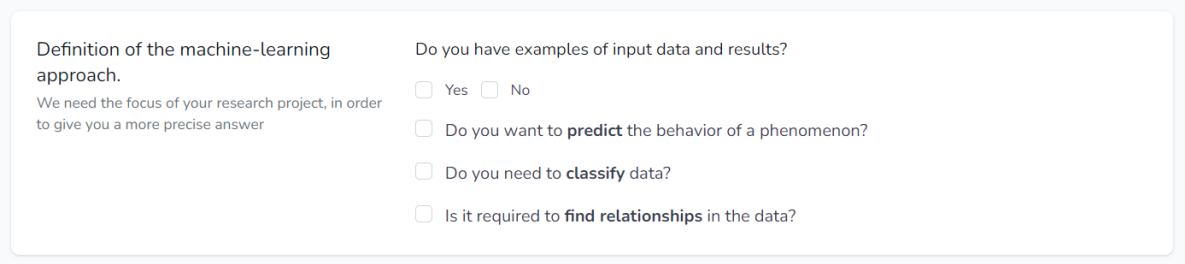


Figura 4.4: Definición del algoritmo

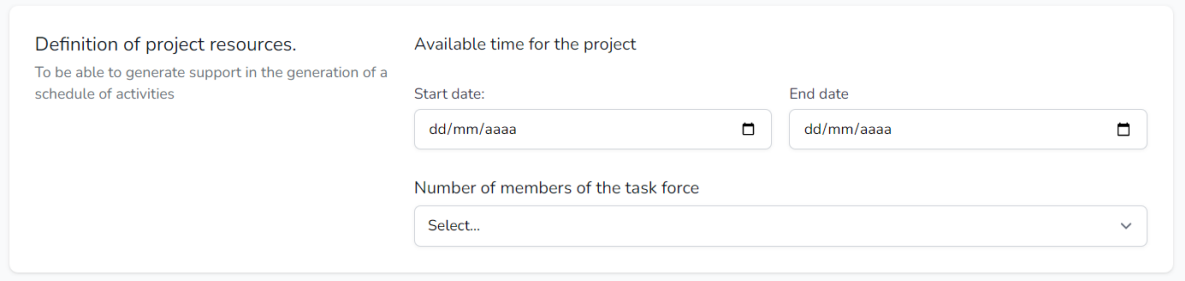


Figura 4.5: Definición de recursos

seguido de los objetivos secundarios, estos últimos ya cuentan con 5 objetivos sugeridos, los cuales se consideran primordiales para todo proyecto de investigación con machine learning. Sin embargo, todos los objetivos pueden ser modificados si se considera necesario.

Los objetivos específicos corresponden a tareas que se deben realizar durante el desarrollo del proyecto y serán desplegados en el cronograma generado.

La definición de hipótesis, alcance y limitaciones del proyecto se realizará de manera semiautomática al presionar el botón "Generate", el cual se aprecia en la Figura 4.7, con

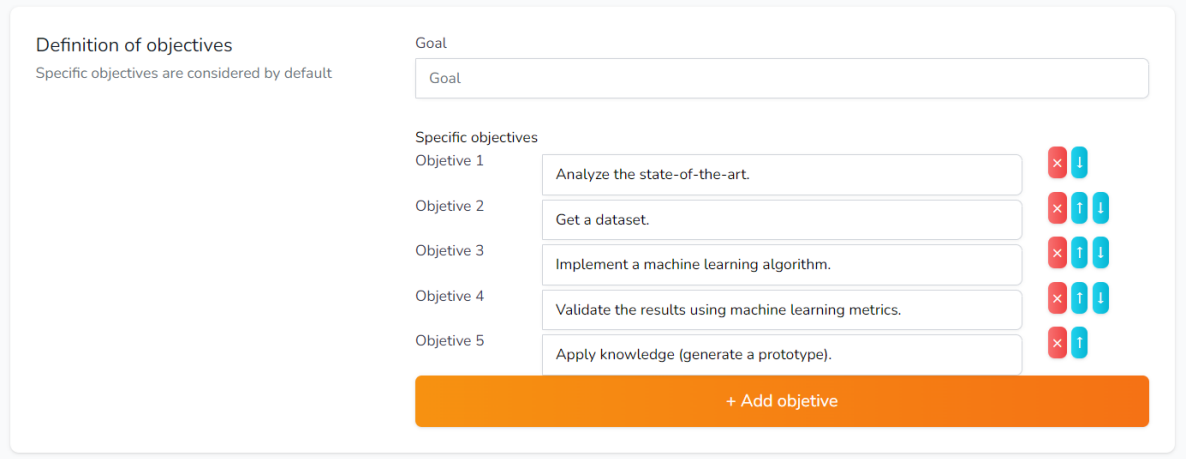


Figura 4.6: Definición de objetivos

la implementación del modelo de lenguaje GPT2-xl, el cual definirá dichas secciones de acuerdo a la entrada dada durante las fases anteriores. En caso de que el usuario considere que la propuesta del modelo no es correcta, este puede modificar estas secciones a su gusto y necesidades.

Las especificaciones y capacidades del modelo se explican en la sección selección del modelo de lenguaje 4.1.1.

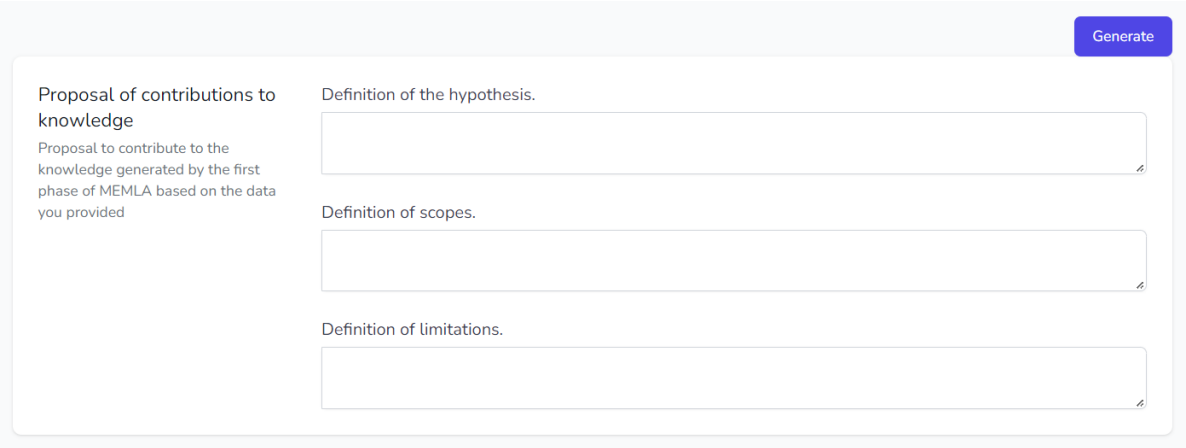


Figura 4.7: Definición de hipótesis, alcances y limitaciones

4.1.4. Fase 4. Pruebas y validación

Durante la estancia de investigación que se realizó en la Unidad de investigación y desarrollo tecnológico (UNINDETEC) de la Marina Armada de México, se realizaron pruebas de usuarios para validar la funcionalidad y utilidad del proyecto.

La población con la que se realizaron las pruebas consta de 24 hombres y mujeres de entre 18 a 35 años, con educación superior o maestría, laborando en el ámbito del desarrollo de proyectos de investigación y software.

A los usuarios se les otorgaron los datos de una tesis de investigación realizada en la unidad y se les asignó una hora para realizar las pruebas con la aplicación que consideraran necesarias. Una vez terminadas las pruebas los usuarios contestaron una encuesta de satisfacción, cuyos resultados pueden observarse en la tabla 4.1, o de forma individual en el apéndice A.

Las pruebas realizadas fueron de funcionamiento para cada una de las secciones de la aplicación y del modelo. A su vez, las pruebas del modelo se desempeñaron en distintos equipos con la finalidad de evaluar las limitaciones que el hardware puede imponer, estas pruebas se describen con detalle en la sección resultados 4.2.

4.2. Resultados

Para los experimentos se utilizó una plantilla para ser procesada por el agoritmo GPT2-xl, la cual se compone por: información ingresada por el usuario (título, problemática, justificación y objetivo principal), hipótesis generada por el algoritmo, alcances generador por el algoritmo y limitaciones generadas por el algoritmo. Las secciones generadas por el algoritmo se ingresan secuencialmente con la finalidad de conservar coherencia entre cada sección, es decir, la hipótesis generada alimenta a la siguiente generación de los alcances, y los alcances alimentan a las limitaciones.

Una vez capturados los datos en la aplicación web, estos son utilizados por el modelo GPT2-xl, el cual retorna resultados como se ven en la Figura 4.8. En promedio, el modelo retorna resultados en un tiempo aproximado de 10 minutos.

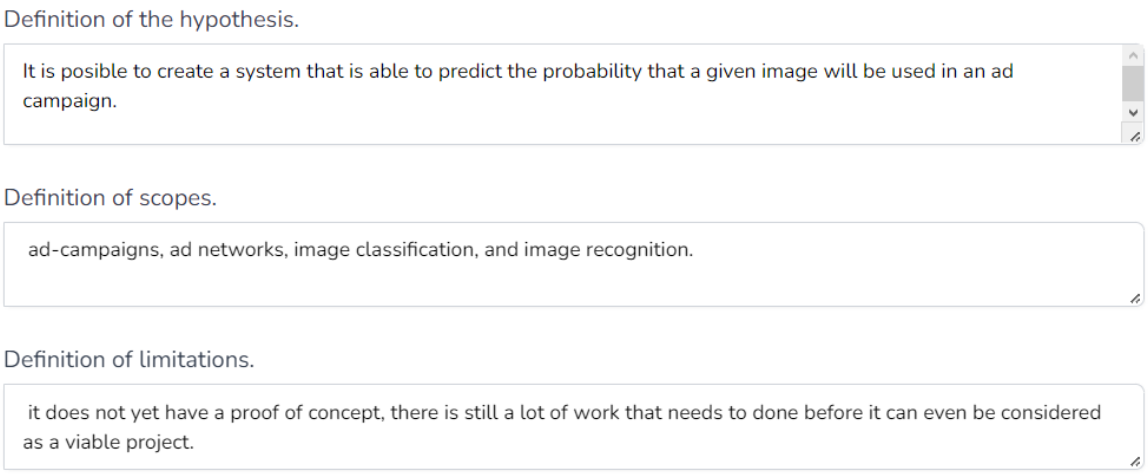


Figura 4.8: Resultados generados con el modelo de NLP

Al ser estos resultados propuestas para el usuario no es fácil calificar la calidad del texto generado por el modelo, ya que una hipótesis puede ser acertada o no de acuerdo al criterio de cada usuario.

En la tabla 4.1, se encuentran las preguntas de la encuesta de satisfacción efectuada, las posibles respuestas para cada pregunta y junto está la cantidad de respuestas obtenidas.

Tabla 4.1: Preguntas de encuesta de satisfacción.

Preguntas	**Posibles respuestas**	**Respuestas obtenidas**
¿Has utilizando algún software generador de texto anteriormente?	Si	16
	No	8
¿Cómo calificarías el tiempo de ejecución de memla.mx?	Rápido	3
	Bueno	15
	Normal	4
	Regular	1
	Lento	1
¿Qué tan congruente es el resultado generado por memla.mx con los datos ingresados?	Fuertemente relacionado	6
	Relacionado	17
	Neutral	-
	No relacionado	1
	Fuertemente no relacionado	-
¿Encontraste algún error ortográfico o gramatical en el resultado generado por memla.mx?	Si, de todo tipo	2
	Si, errores gramaticales	1
	Si, errores ortográficos	9
	No, ningún error	-
¿Usarías los resultados otorgados por memla.mx en tu proyecto?	Si, absolutamente	1
	Si, con cambios	19
	No, pero es de ayuda	1
	No	-
Califica el diseño de la página de memla.mx	Malo	-
	Mejorable	-
	Regular	4
	Bueno	14
	Excelente	6

Continua en la siguiente página

Tabla 4.1 – Continuación de la página previa

Preguntas	Posibles respuestas	Respuestas obtenidas
¿Cómo calificarías la facilidad para usar la página de memla.mx?	Excelente Bueno Regular Mejorable Malo	3 21 - - -
¿Crees que memla.mx puede ayudarte a desarrollar tu proyecto más eficientemente?	Fuertemente de acuerdo De acuerdo Neutral En desacuerdo Fuertemente en desacuerdo	6 16 2 - -
¿Crees que es necesario mejorar alguna sección de memla.mx?	Generalidades Definición de acercamiento de aprendizaje automático Definición de recursos del proyecto Definición de objetivos Definición del proyecto	6 4 6 2 2

Gracias a las encuestas de satisfacción realizadas por los usuarios tras la prueba del sistema se pudo extraer la siguiente información:

- La mayoría de los usuarios han utilizados herramientas para generar texto.
- Se considera que el tiempo de ejecución de la aplicación es adecuado.
- El texto generado es congruente con el texto ingresado por el usuario.
- Es posible que el algoritmo genere algunos errores ortográficos en su resultado.
- Los usuarios harían uso de los resultados generados por la aplicación en sus proyectos personales.
- El diseño de la aplicación web es agradable para los usuarios.
- La aplicación web es sumamente amigable al usuario.
- Los usuarios consideran que la aplicación es útil en el desarrollo de sus proyectos.
- El proyecto cuenta con margen de mejora, pero no se detectaron puntos especialmente débiles.

4.2.0.1. Experimentos de tiempo de ejecución

Al realizar experimentos en diferentes equipos de computo se obtuvieron diferentes tiempos de ejecución, en la Tabla 4.2 se puede ver el promedio de tiempo de ejecución que demoró el algoritmo en generar cada sección del protocolo.

Los experimentos procesados con equipos de computo que no cuentan con unidades gráficas de procesamiento (GPU), procesan el algoritmo en un tiempo promedio de 30 minutos, mientras que las maquinas con GPU realizaron los ejercicios en menos del 50 % del tiempo. Para esto el algoritmo se proceso dentro de los núcleos de Arquitectura Unificada de Dispositivos de Cómputo (CUDA) de las GPU.

Tabla 4.2: Comparación de tiempo al ejecutar NLP

Descripción del equipo	Tiempo de generación de hipótesis	Tiempo de generación de alcances	Tiempo de generación de limitaciones
Procesador i5-6300HQ 2.3GHz y 8Gb RAM	15min	13min	12min
Procesador i5-6300HQ 2.3GHz GPU Nvidia GTX 960m y 8Gb RAM	6.31min	5.51min	5.19min
Procesador i5-6300HQ 2.3GHz, GPU Nvidia GTX 960m y 16Gb RAM	2.22min	1.24min	1.52min
Procesador i5-9400HQ 2.9GHz (6), GPU Nvidia RTX 2060 (2), Nvidia RTX 3060, y 16Gb RAM	18s	12s	12s

4.3. Configuración experimental

Este experimento se realizó empleando 2 computadoras, una como servidor de la API y otra como servidor para la aplicación web. El servidor API corre sobre un sistema operativo windows 10 de 64 bits, con un procesador Intel (R) core i5-6300HQ CPU @ 2.30GHz, 16Gb de memoria RAM y una tarjeta gráfica Nvidia 960M con 4Gb de memoria. La versión de Python es 3.10.7. La aplicación de python utilizado es Flask 2.2[2].

[2]`https://flask.palletsprojects.com/en/2.2.x/installation/`

Para el desarrollo de la aplicación presentado en este trabajo se construyó con la herramienta de desarrollo Laravel versión 8[3], utilizando PHP version 7.4, un servidor de python 3.10.7, bajo un sistema operativo OSX versión 10.13m con 8 GB de memoria RAM y un procesador Intel Core i5 a 2.4 GHz, el sistema fue puesto en linea en un servidor virtual privado, con un sistema operativo Ubuntu 22.04 (LTS) de 64 bits, 24 GB de almacenamiento, 1GB de memoria RAM, 1 procesador virtual de Intel.

[3] `https://laravel.com/docs/9.x/upgrade`

Capítulo 5

Conclusiones y trabajos a futuro

Con el desarrollo de esta investigación se demostró que si es posible implementar un algoritmo de procesamiento de lenguaje natural en una API web en comunicación con una aplicación web para crear protocolos de proyectos de investigación con base en texto ingresado por el usuario.

Gracias al código abierto y las nuevas tecnologías, la implementación de un algoritmo de procesamiento de lenguaje natural en una API para consumo desde una aplicación web es completamente viable y sin complicaciones, sin embargo, debe tomarse a consideración las capacidades de hardware con las que se cuentan, pues dichos grandes modelos de lenguaje son cada vez más poderosos y requieren de mayor poder computacional para ser operados a nivel de producción.

La aplicación demora un lapso de entre 10 y 15 minutos para generar un protocolo de proyecto, un tiempo bastante bueno considerando que el desarrollo del mismo protocolo puede tardar horas o días a un investigador poco experimentado.

Los modelos de lenguaje actuales son capaces de generar dialogo indistinguible al de un humano, por lo cual usarlos para generar texto en un formato científico es sencillo con la petición correcta. Con ayuda de la plantilla de contexto se diseño una petición que genera la hipótesis, los alcances y las limitaciones de un protocolo de tesis.

Durante las pruebas con los usuarios se descubrió que los resultados generados por el modelo fueron, en su mayoría, de buena calidad, sin errores ortográficos ni gramaticales y relacionados con el tema de la investigación, sin embargo, si existieron casos negativos. Gracias a la disponibilidad de la aplicación pueden realizarse pruebas adicionales con la finalidad de descartar o solucionar dichos casos negativos.

La mayoría de los usuarios consideraron que el uso de la aplicación web MEMLA les permitiría desarrollar sus proyectos de investigación de forma más rápida y eficiente utilizando los resultados generados, ya sea de forma parcial o como base para desarrollar sus propios conceptos.

El proyecto cumple con su propósito de ayudar a estudiantes o investigadores a desarrollar sus protocolos de proyecto de forma eficaz y eficiente, y se espera poder continuar el desarrollo del mismo hasta su perfección en los años venideros.

Para la continuación del proyecto se proponen la siguiente lista de oportunidades de mejora:

- Mejorar el hardware del servidor de inteligencia artificial.
- Implementar un algoritmo de procesamiento de lenguaje natural más avanzado.
- Desarrollar un algoritmo de procesamiento de lenguaje natural propio.
- Modificar la aplicación web para solicitar más detalles del proyecto que se realiza.
- Modificar la plantilla de contexto para considerar más detalles del proyecto.
- Mejorar el archivo resultante.
- Crear una guía de uso para los clientes.

Referencias

[1] M. C. Sánchez, *Guıa para la formulación de proyectos de investigación*. Coop. Editorial Magisterio, 2004.

[2] B. Cheng, E. Asphaug, R.-L. Ballouz, Y. Yu y H. Baoyin, "Numerical Simulations of Drainage Grooves in Response to Extensional Fracturing: Testing the Phobos Groove Formation Model," *The Planetary Science Journal*, vol. 3, n.º 11, pág. 249, 2022.

[3] J. Miller, A. Emadi, A. Rajarathnam y M. Ehsani, "Current status and future trends in more electric car power systems," en *1999 IEEE 49th Vehicular Technology Conference (Cat. No. 99CH36363)*, IEEE, vol. 2, 1999, págs. 1380-1384.

[4] J. Sun, X. Dai, Q. Wang, M. C. van Loosdrecht y B.-J. Ni, "Microplastics in wastewater treatment plants: Detection, occurrence and removal," *Water research*, vol. 152, págs. 21-37, 2019.

[5] J. S. Stevenson, L. Clifton, E. Kuźma y T. J. Littlejohns, "Speech-in-noise hearing impairment is associated with an increased risk of incident dementia in 82,039 UK Biobank participants," *Alzheimer's & Dementia*, vol. 18, n.º 3, págs. 445-456, 2022.

[6] X.-Y. Zhou, Y. Guo, M. Shen y G.-Z. Yang, "Application of artificial intelligence in surgery," *Frontiers of medicine*, vol. 14, págs. 417-430, 2020.

[7] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu y M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[8] M. Broussard, N. Diakopoulos, A. L. Guzman, R. Abebe, M. Dupagne y C.-H. Chuan, "Artificial intelligence and journalism," *Journalism & mass communication quarterly*, vol. 96, n.º 3, págs. 673-695, 2019.

[9] E. Adamopoulou y L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, pág. 100 006, 2020.

[10] C. A. Bernal Torres y col., "Metodologıa de la Investigación para Administración y Economıa," 2000.

[11] V. Morles, "Guıa para la elaboración y evaluación de proyectos de investigación," *Revista de pedagogıa*, vol. 32, n.º 91, págs. 131-146, 2011.

[12] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, págs. 81-106, 1986.

[13] T. Evgeniou y M. Pontil, "Support vector machines: Theory and applications," en *Advanced course on artificial intelligence*, Springer, 1999, págs. 249-257.

[14] B. W. Silverman y M. C. Jones, "E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)," *International Statistical Review / Revue Internationale de Statistique*, vol. 57, n.º 3, págs. 233-238, 1989, ISSN: 03067734, 17515823. dirección: `http://www.jstor.org/stable/1403796` (visitado 29-06-2023).

[15] E. Bisong y col., *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019.

[16] Y. LeCun, Y. Bengio y G. Hinton, "Deep learning," *nature*, vol. 521, n.º 7553, págs. 436-444, 2015.

[17] A. Saez De La Pascua, "Deep learning para el reconocimiento facial de emociones básicas," B.S. thesis, Universitat Politècnica de Catalunya, 2019.

[18] A. M. Turing, "Computing machinery and intelligence," en *Parsing the turing test*, Springer, 2009, págs. 23-65.

[19] W. X. Zhao, K. Zhou, J. Li y col., "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[20] T. Gao, "Prompting: Better Ways of Using Language Models for NLP Tasks," *The Gradient*, 2021.

[21] D. Dai, Y. Sun, L. Dong, Y. Hao, Z. Sui y F. Wei, "Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta Optimizers," *arXiv preprint arXiv:2212.10559*, 2022.

[22] Z. Hou, J. Salazar y G. Polovets, "Meta-learning the difference: Preparing large language models for efficient adaptation," *Transactions of the Association for Computational Linguistics*, vol. 10, págs. 1249-1265, 2022.

[23] R. Azamfirei, S. R. Kudchadkar y J. Fackler, "Large language models and the perils of their hallucinations," *Critical Care*, vol. 27, n.º 1, págs. 1-2, 2023.

[24] M. Zhang, O. Press, W. Merrill, A. Liu y N. A. Smith, "How language model hallucinations can snowball," *arXiv preprint arXiv:2305.13534*, 2023.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever y col., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, n.º 8, pág. 9, 2019.

[26] I. Solaiman, M. Brundage, J. Clark y col., "Release strategies and the social impacts of language models," *arXiv preprint arXiv:1908.09203*, 2019.

[27] A. Vaswani, N. Shazeer, N. Parmar y col., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[28] S. Maheshwari, "Incremental clustering in intension mining framework," Tesis doct., Indian Institute of Technology, 2000.

[29] S. Gupta, V. Bhatnagar y S. K. Wasan, "Architecture for knowledge discovery and knowledge management," *Knowledge and Information Systems*, vol. 7, n.º 3, págs. 310-336, 2005.

[30] S. Sharma y K. Osei-Bryson, "An integrated knowledge discovery and data mining process model," Tesis doct., Virginia Commonwealth University, 2008.

[31] S. Coleman, R. Göb, G. Manco, A. Pievatolo, X. Tort-Martorell y M. S. Reis, "How can SMEs benefit from big data? Challenges and a path forward," *Quality and Reliability Engineering International*, vol. 32, n.º 6, págs. 2151-2164, 2016.

[32] D. Bahdanau, K. Cho e Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[33] K. Xu, J. Ba, R. Kiros y col., "Show, attend and tell: Neural image caption generation with visual attention," en *International conference on machine learning*, PMLR, 2015, págs. 2048-2057.

[34] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola y E. Hovy, "Hierarchical attention networks for document classification," en *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, págs. 1480-1489.

[35] M. Kais y col., "Bootstrapping the CRISP-DM Process," Tesis de mtría., 2017.

[36] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston y M. Auli, "ELI5: Long form question answering," *arXiv preprint arXiv:1907.09190*, 2019.

[37] S. Lin, J. Hilton y O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," *arXiv preprint arXiv:2109.07958*, 2021.

[38] R. Nakano, J. Hilton, S. Balaji y col., "WebGPT: Browser-assisted question-answering with human feedback," *arXiv preprint arXiv:2112.09332*, 2021.

[39] S. Zhang, S. Roller, N. Goyal y col., "OPT: Open Pre-trained Transformer Language Models," *arXiv preprint arXiv:2205.01068*, 2022.

[40] H. Touvron, T. Lavril, G. Izacard y col., "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[41] S. R. Ali, T. D. Dobbs, H. A. Hutchings e I. S. Whitaker, "Using ChatGPT to write patient clinic letters," *The Lancet Digital Health*, vol. 5, n.º 4, e179-e181, 2023.

[42] *SmartPaper AI*, `https://smartpaper.ai`, Accessed: 2023-07-17.

[43] J. White, Q. Fu, S. Hays y col., "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

[44] J. Wang, E. Shi, S. Yu y col., "Prompt engineering for healthcare: Methodologies and applications," *arXiv preprint arXiv:2304.14670*, 2023.

[45] Y. Zhou, B. Zhou, L. Pache y col., "Metascape provides a biologist-oriented resource for the analysis of systems-level datasets," *Nature communications*, vol. 10, n.º 1, págs. 1-10, 2019.

[46] K. Shuster, J. Xu, M. Komeili y col., "Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage," *arXiv preprint arXiv:2208.03188*, 2022.

[47] *Reinventing search with a new AI-powered Microsoft Bing and EDGE, your copilot for the web*, `https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/`, Accessed: 2023-03-15, feb. de 2023.

[48] D. Kalla y N. Smith, "Study and Analysis of Chat GPT and its Impact on Different Fields of Study," *International Journal of Innovative Science and Research Technology*, vol. 8, n.º 3, 2023.

[49] S. Moyle y A. Jorge, "Ramsys-a methodology for supporting rapid remote collaborative data mining projects," en *ECML/PKDD01 Workshop: Integrating Aspects of Data Mining, Decision Support and Meta-learning (IDDM-2001)*, vol. 64, 2001.

[50] K. Crowston, J. S. Saltz, A. Rezgui, Y. Hegde y S. You, "Socio-technical affordances for stigmergic coordination implemented in MIDST, a tool for data-science teams," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, n.º CSCW, págs. 1-25, 2019.

[51] *Hugging Face*, `https://huggingface.co`, Accessed: 2022-12-01.

[52] K. Manikas, "Revisiting software ecosystems research: A longitudinal literature study," *Journal of Systems and Software*, vol. 117, págs. 84-103, 2016.

Apéndice A

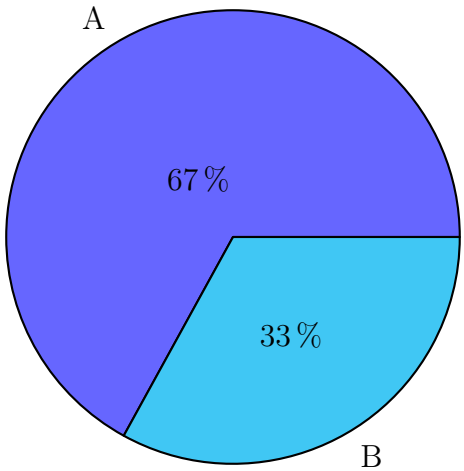
Resultados encuestas de satisfacción

A continuación se presentan los resultados obtenidos de las encuestas de satisfacción aplicadas a los usuarios tras la prueba de la aplicación. Los resultados se desglosan por pregunta, una lista de las posibles respuestas y la gráfica correspondiente.

"1) ¿Has utilizando algún software generador de texto anteriormente?"

A. Si

B. No



"2) ¿Cómo calificarías el tiempo de ejecución de memla.mx?"

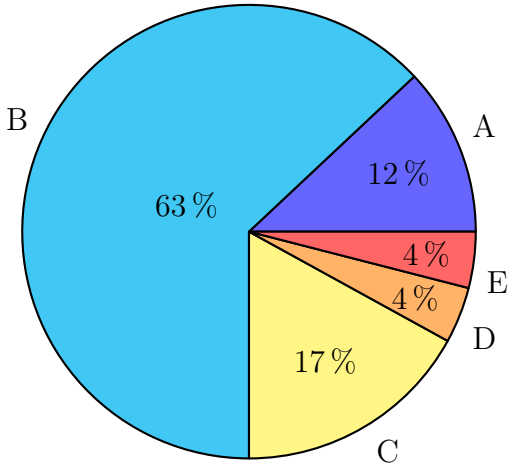
A. Rápido

B. Bueno

C. Normal

D. Regular

E. Lento



"¿Qué tan congruente es el resultado generado por memla.mx con los datos ingresados?"

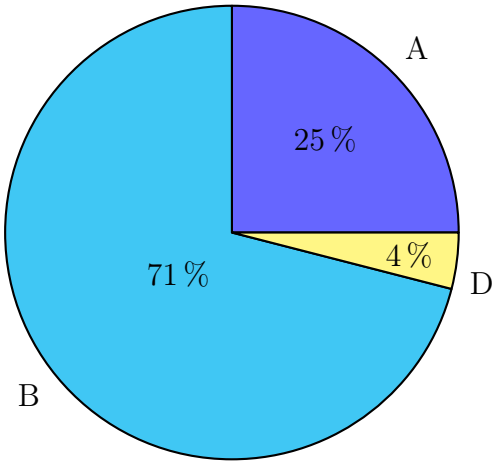
A. Fuertemente relacionado

B. Relacionado

C. Neutral

D. No relacionado

E. Fuertemente no relacionado



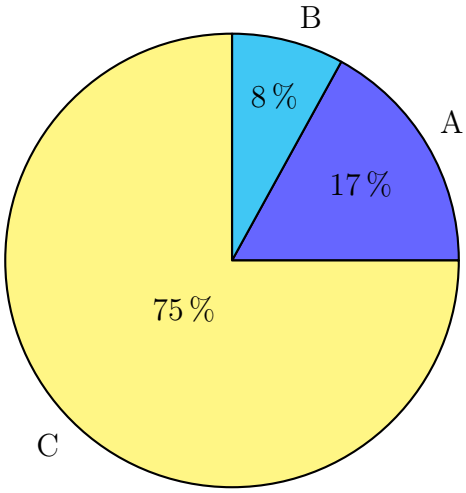
"¿Encontraste algún error ortográfico o gramatical en el resultado generado por memla.mx?"

A. Si, de todo tipo

B. Si, errores gramaticales

C. Si, errores ortográficos

D. No, ningún error



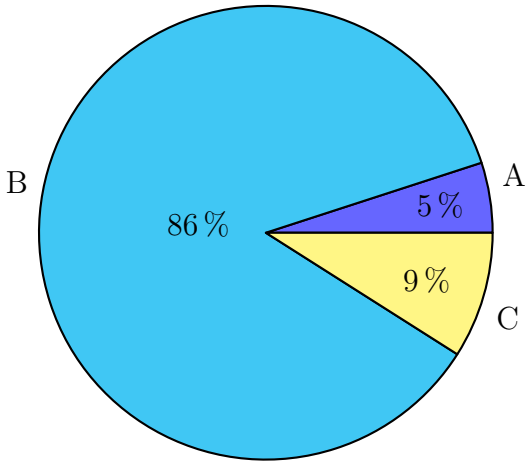
"¿Usarías los resultados otorgados por memla.mx en tu proyecto?"

A. Si, absolutamente

B. Si, con cambios

C. No, pero es de ayuda

D. No



Çalifica el diseño de la página de memla.mx"

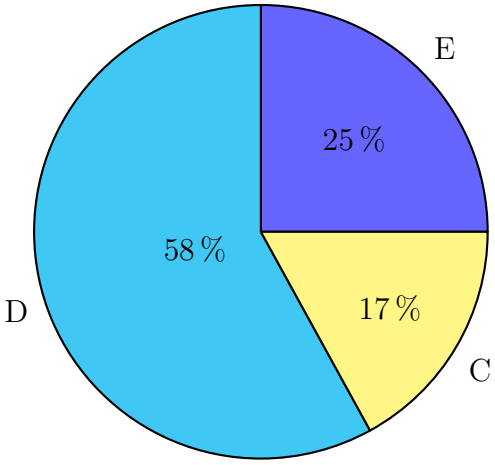
A. Malo

B. Mejorable

C. Regular

D. Bueno

E. Excelente



"¿Cómo calificarías la facilidad para usar la página de memla.mx?"

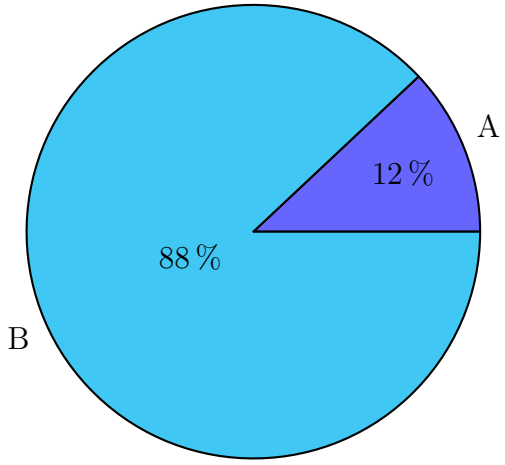
A. Excelente

B. Bueno

C. Regular

D. Mejorable

E. Malo



"¿Crees que memla.mx puede ayudarte a desarrollar tu proyecto más eficientemente?"

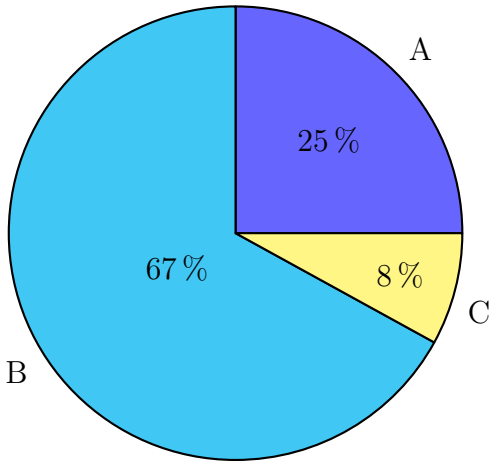
A. Fuertemente de acuerdo

B. De acuerdo

C. Neutral

D. En desacuerdo

E. Fuertemente en desacuerdo



"¿Crees que es necesario mejorar alguna sección de memla.mx?"

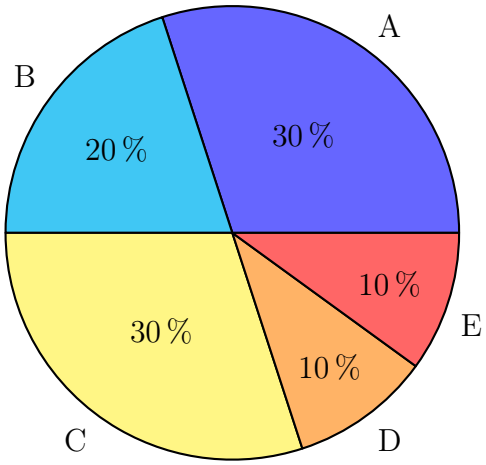
A. Generalidades

B. Definición de acercamiento de aprendizaje automático

C. Definición de recursos del proyecto

D. Definición de objetivos

E. Definición del proyecto



Apéndice B

Plantilla de contexto

Para realizar una petición al modelo GPT2-xl es necesario indicarle por texto el resultados que se espera, utilizando la siguiente plantilla es posible dar contexto de los datos de entrada y guiar al modelo para devolver un resultado correcto.

"The following is the tittle of an investigation project that uses different technologies to achieve its objectives: " *ingresar el título del proyecto*

"The problem that the project seeks to solve is: " *ingresar la problemática del proyecto*

"The justification for the project is: " *ingresar la justificación del proyecto*

"The main goal the proyect seek to achieve is: " *ingresar el objetivo general del proyecto*

"A hypothesis of just one paragraph for said project could be: It is posible to " *hipótesis generada*

"A list of maximum 5 scopes for the project could be: 1)" *alcances generados*

"A list of maximum 5 limitations for the project could be: 1)" *limitaciones generadas*