SEP **SES** **TNM**

TECNOLÓGICO NACIONAL DE MÉXICO CAMPUS CHIHUAHUA II

“OBTENCIÓN RÁPIDA DE PREGUNTAS DE OPCIÓN MÚLTIPLE A PARTIR DE UNA LISTA DE ORACIONES TEXTUALES PARA LA GENERACIÓN DE EXÁMENES”

TESIS

PARA OBTENER EL GRADO

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA

IRVING FERNANDO HERNÁNDEZ CARRASCO

CODIRECTOR DE TESIS

DR. RICARDO BLANCO VEGA

DIRECTOR DE TESIS

DRA. MARISELA IVETTE CALDERA FRANCO

Dictamen

Chihuahua, Chihuahua, 05 de junio del 2020

M.C. MARÍA ELENA MARTÍNEZ CASTELLANOS
COORDINADORA DE POSGRADO
Presente. -

Por medio de este conducto el comite tutorial revisor de la tesis para obtención de grado de Maestro en Sistemas Computacionales, que lleva por nombre "**OBTENCIÓN RÁPIDA DE PREGUNTAS DE OPCIÓN MÚLTIPLE A PARTIR DE UNA LISTA DE ORACIONES TEXTUALES PARA LA GENERACIÓN DE EXÁMEN**", que presenta el (la) **C. IRVING FERNANDO HERNÁNDEZ CARRASCO**, hace de su conocimiento que despues de ser revisado ha dictaminado la **APROBACIÓN** del mismo.

Sin otro particular de momento, queda de Usted

Atentamente

La Comisión de Revisión de Tesis.

DRA. MARISELA IVETTE CALDERA FRANCO
Director de Tesis

MTRO. RICARDO BLANCO VEGA
Co-Director

Leonardo Nevárez Chávez
M.C. LEONARDO NEVÁREZ CHÁVEZ
Revisor

A. Alvarado G.
M.I.S.C. JESÚS ARTURO ALVARADO GRANADINO
Revisor

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I. INTRODUCCIÓN

En este primer capítulo se pretende brindar un panorama general del trabajo presente, y dar a conocer la temática que se desarrolla en cada uno de los capítulos que conforma la presente tesis titulada “Obtención rápida de preguntas de opción múltiple a partir de una lista de oraciones textuales para la generación de exámenes”.

1.1 Introducción

Este es un proyecto que se realiza en el Tecnológico Nacional de México Campus Chihuahua II con el propósito de obtener el grado de Maestro en Sistemas Computacionales, proponiendo una aportación al ámbito educativo a través del Procesamiento de Lenguaje Natural (PLN). El proyecto de esta tesis se incorpora a la creación de recursos interactivos de la empresa Paco el Chato (PEC), lo cual sirve para la generación de exámenes que son de utilidad para dicha plataforma.

PEC es una empresa de Coppel S.A de C.V, la cual está enfocada al desarrollo e implementación de las nuevas tecnologías a favor de la educación en México, dicha empresa cuenta con una plataforma de contenidos de fácil acceso y basada en la nube.

Actualmente los exámenes de PEC se generan con preguntas de forma semi-manual. El material con el que se cuenta para dicha generación de reactivos está basado en los libros de texto (Secretaría de Educación Pública, 2020) alineados a la Secretaría de Educación Pública de México (SEP). Por lo que esta tesis se enfoca en crear cuestionarios de una manera rápida, para poder formar exámenes, que tal como menciona Butler (Butler & Roediger, 2008) ayudan a los alumnos en su aprendizaje debido a la retroalimentación. Los cuestionarios ayudan a las personas a fomentar la reflexión y razonamiento referente a un tema, la capacidad de verificar lo que se ha aprendido y estimular la autoevaluación.

Mediante la implementación de cuestionarios como una herramienta que emplean los docentes, se da lugar a la evaluación de conocimientos significativos en el alumno y toda persona en general.

Se parte de un conjunto de oraciones y se genera un cuestionario con preguntas de opción múltiple con selección simple. El producto final de la tesis es una librería que contienen los paquetes de clases desarrollados en el lenguaje de programación Java. Las pruebas se realizan con temas de los libros de primaria de la SEP.

El cometido de hacer uso de esta herramienta educativa (los exámenes) son: reflejar el nivel de procesamiento de la información que tiene el alumno, evaluar el conocimiento adquirido y estimular el pensamiento. Según Mugsnoticias (Rivas, 2016), para que los niños iniciaran con el hábito de la lectura se imprimió el libro Español: Lecturas, para primer grado de primaria, en 1997, con un total de 39 cuentos cortos, de los cuales el primero fue “Paco el chato”, el cual resalta la importancia de que los niños conozcan su nombre.

Se utiliza el modelo de desarrollo de software incremental el cual permite obtener los requerimientos a través de un análisis, diseñar, codificar, probar y su debido mantenimiento son etapas que son fundamentales en el ciclo de vida del software propuesto para la tesis, de esta forma se pueden realizar iteraciones hasta encontrar una versión adecuada.

1.2 Planteamiento del problema

La generación de cuestionarios para PEC de inició es de forma manual, lo cual es tardado, por lo que es necesario obtener dichos cuestionarios de una forma más rápida.

Los exámenes se generan con una herramienta de edición. En dos alternativas, una manera es la creación manual desde cero y la otra alternativa es de forma semiautomática utilizando los exámenes de ENLACE (Evaluación Nacional de Logro Académico en Centros Escolares), la herramienta de edición realiza la búsqueda automática de las preguntas albergadas en un archivo de formato PDF y luego el usuario seleccionaba cual era la respuesta correcta, y finalmente se logra una colección de los cuadernillos de ENLACE de la SEP que tienen las respuestas y podemos obtenerla automáticamente.

Actualmente se cuenta con un CMS (Content Management System) o Sistema de gestión de contenidos, que permite crear manualmente los exámenes. Los exámenes están publicados en pacoelchato.org y en pacoelchato.com.

La forma manual de la generación de preguntas en su proceso es lento debido a que se debe dar de alta cada pregunta con sus respectivas respuestas, donde el docente o director de grado define la calidad de la formulación de la pregunta y dichas respuestas a través de su criterio al momento de crear tales reactivos.

La alternativa semiautomática requiere de estar haciendo consultas directas a los exámenes de ENLACE, pero esta forma no es capaz de generar preguntas y respuestas a partir de un texto o un tema en específico; este proceso tiene una dependencia muy estrecha con los recursos de ENLACE que aporta la SEP.

1.3 Alcances y limitaciones

Alcances

- Con este proyecto se observa una mejoría en la creación de exámenes para PEC, generando preguntas y que la respuesta a la misma no se realice de forma obvia o por sentido común.
- Además, fomentar la autoevaluación, lo cual sirve para marcar un punto de referencia entre lo desconocido y el conocimiento adquirido.
- Se brinda rapidez en la creación de cuestionarios, formulados con sus respectivas preguntas y respuestas.Se pretende deducir si una oración contiene la información necesaria para crear una pregunta.
- Solo se considera la generación de preguntas de opción múltiple estilo completa la oración.
- Seleccionar las respuestas distractoras de mayor calidad. Se propone una forma de evaluar la calidad de las respuestas distractoras.
- Selección de la entidad con mayor relevancia de una oración, la cual será la respuesta que conteste correctamente.

Limitaciones

- No se toma en cuenta libros que tengan expresiones matemáticas y formulas.
- Las preguntas generadas son extraídas de texto, es decir, no se incluye un análisis de imágenes que pudieran contener texto.
- Se prueban solamente con libros de texto de primaria de la SEP, ya sea español, ciencias naturales, historia, y geografía.
- El análisis textual solamente se realiza para textos del idioma español.
- Para realizar el análisis de texto, la conexión a internet es requerida.

1.4 Justificación

El hacer manualmente el material de evaluación es lento, por lo que brindar una manera rápida y con poca intervención de una persona ayuda a la creación de cuestionarios, que a diferencia de hacerlos manualmente aporta un apoyo significante.

Los contenidos de los libros suelen cambiar por lo que se hace más necesaria la herramienta propuesta en esta tesis, ya que el docente debe actualizar sus exámenes de evaluación conforme avance el grado y numero de lecciones de un libro. De igual manera cada determinado tiempo la SEP decide actualizar o cambiar los contenidos temáticos de los libros o la renovación total de los mismos.

Se pretende generar cuestionarios para el apoyo del aprendizaje en el ámbito estudiantil, con un enfoque principalmente en la educación básica mexicana.

Con este proyecto se pretende abarcar el trabajo del docente al momento de realizar cuestionarios, lo cual le permite tomar un papel multifuncional más hábil.

La formación de una pregunta de calidad, en cuanto a estructura e información se refiere, propicia a el alumno o la persona que realice dicho cuestionario o examen, la necesidad del estudio de las sesiones de clase, tareas correspondientes o bien el conocimiento del tema a tratar dentro de los reactivos.

En la tabla 1 se muestran ejemplos de lo que se puede considerar respuestas obvias (se selecciona la respuesta correcta usando lógica) contra respuestas confusas (no es posible utilizar la lógica, debe utilizar su conocimiento adquirido) para el alumno, estás últimas apoyando a su aprendizaje de las sesiones de clase y trabajos realizados así en aula como en casa.

Ejemplo:

Seleccione la respuesta que corresponde a un tipo de árbol

El _____ es un tipo de árbol que desciende del tipo higueras.

Tabla 1. Ejemplos de respuestas obvias y respuestas confusas

Respuesta obvia	Respuesta confusa
1.-Piedra	1.-Bambú
2.-Pedernal	2.-Pino
3.-Nogal	3.-Nogal
4.-Cuarzo	4.-Sicomoro

Se busca obtener respuestas incorrectas o distractoras que sean semejantes o similares a la respuesta correcta, donde las respuestas incorrectas no sean posibles de descartar por lógica y asumir la respuesta correcta de una forma muy simple.

Se pretende agilizar y hacer más rápido la generación de estos exámenes y reducir la intervención de un usuario lo más posible que se pueda. También se propone determinar si la oración tiene información para extraer los reactivos, como primer análisis se contempla que la oración contenga más de cierto número de palabras, por lo menos un sustantivo, un verbo, un adjetivo y una entidad.

A través de la API (Interfaz de Programación de Aplicaciones) de PLN se identifican los tokens y de ellos la entidad que tenga mayor nivel de importancia, la cual servirá de respuesta correcta; así como buscar palabras homólogas que sirvan de respuestas incorrectas a través de un análisis en paralelo de dicha entidad; esto ofrece una justificación viable para las respuestas.

1.5 Objetivo

Se describe el objetivo general y cada uno de los objetivos específicos que implican para lograr este mismo.

1.5.1 Objetivo General

Desarrollar una herramienta informática para crear exámenes a partir de una lista de oraciones o un texto dado, de tal forma que este proceso sea de una manera rápida y automática. Dando lugar a que la formulación de una pregunta tenga solides en cuanto a su estructura de sintaxis; a lo igual para las respuestas, lograr una manera en que las respuestas distractoras o erróneas sean homologas a las correctas.

1.5.2 Objetivos específicos

- Automatizar la generación de cuestionarios a través de la empleabilidad de software.
- A partir de texto, realizar un análisis que brinde información de las propiedades de cada palabra que lo contengan.
- Validar si las respuestas están completas en el reactivo, mediante la evaluación del número de respuestas asignadas a una sola pregunta.
- Generar preguntas bien estructuradas y coherentes, identificando propiedades de las oraciones que en conjunto permitan proponer una solides.
- Reducir el tiempo invertido en la creación de exámenes, mediante una generación automatizada de reactivos en la formulación de preguntas con sus respectivas respuestas correctas.
- Obtener respuestas incorrectas de calidad, es decir, respuestas que sean lo más posiblemente similares a la correcta.
- Agilizar la creación de reactivos que pongan a prueba el estudio de los alumnos.
- Agilizar el trabajo del docente en la creación de exámenes.

1.6 Hipótesis

Es posible crear preguntas de calidad de forma rápida y automática partiendo de un texto, que contenga algún tema en específico. Dicha creación no requerirá la intervención de una persona o un docente para decidir cuales preguntas y cuales respuestas formaran los reactivos.

Las respuestas incorrectas se obtienen de un banco de respuestas generado de todas las entidades que existen dentro del texto analizado; esto con la finalidad de empatar dichas respuestas con la opción o entidad que conteste correctamente a la pregunta.

Respecto a la anterior, se pretende que las respuestas incorrectas sean de calidad, esto quiere decir que deben ser muy semejantes a las respuestas correctas en cuanto su contenido y contexto, por ejemplo, que ambas respuestas (correctas e incorrectas) estén en plural o singular, según sea el caso. La calidad de las respuestas generadas automáticamente será mejor que las respuestas brindadas manualmente.

A través de la empleabilidad del uso del software brindado en esta tesis, se logra reducir en más de un 70% el tiempo invertido en la creación y formulación de exámenes, conformados con sus respuestas correctas e incorrectas, así como sus respectivas preguntas.

CAPÍTULO II. ESTADO DEL ARTE

En este capítulo se describe el software existente o investigaciones realizadas tanto para la obtención de cuestionarios como alguna implementación relevante del PLN.

2.1 Quizlet de Dev-Post

La API de Quizlet (DEVPOST, 2020) es una herramienta que genera pruebas de forma automática a partir de una imagen que contenga un segmento de texto, como notas o páginas de libros. Muchas plataformas de aprendizaje en línea tienen pruebas integradas en ellas. Sin embargo, estos cuestionarios son hechos manualmente por el instructor u organización. Además, un método de estudio muy común empleado por los estudiantes es crear cuestionarios o tarjetas de estudio para ellos mismos.

Su función consiste en permitirles a los usuarios subir una foto de sus notas al servidor. A partir de ahí, los contenidos de los apuntes o notas se analizan y las preguntas relevantes se hacen con las respuestas adecuadas. Después de esto se cargan en Quizlet como conjuntos de estudio para que el alumno estudie cuando lo desee.

El propósito de esta herramienta es eficientizar el tiempo de tareas de estudio y labores cotidianas para el estudiante. Su construcción y desarrollo consiste en tres partes las cuales son Amazon Alexa, la creación de preguntas y la integración de Quizlet.

Se toma como relevancia la creación de preguntas mediante el análisis del texto extraído de las imágenes empleando la API de visión de Google.

Como punto de observación, se menciona que, los pronombres en muchas oraciones deben ser desreferenciados, ya que las oraciones seguirán el formato estándar de sujeto-verbo-objeto en el idioma inglés, y así identificar al sujeto como el primer nombre o pronombre que se produce. Cada pronombre se cambia luego con el sujeto de la oración anterior.

Luego, se identifican oraciones que contienen ciertas palabras clave para convertirlas en un formato de pregunta de relleno en blanco. Cada espacio en blanco se decide en función del valor de la relevancia de esa entidad. En este punto, se tendra una lista de lista que contiene una pregunta y su respuesta, la cual será enviada a Quizlet.

En punto crucial que se presenta en el desarrollo de Quizlet, es el determinar si Alexa Amazon es robusta, debe contar con muchos buenos ejemplos de una posible respuesta para entrenar sus redes neuronales.

2.2 Generación automática de elementos para crear pruebas de opción múltiple

Según Gierl, Mark J., Lai, H. y Turner (Gierl M. J., 2012) se desarrolló una herramienta de generación automática de elementos para crear pruebas de opción múltiple en la Universidad de Alberta en Canadá, con el objetivo de ser utilizada en el área de conocimiento médico; consistió en tener especialistas que pudieran estructurar implícitamente el conocimiento, las habilidades y el contenido necesario para expresarlo en forma de un elemento para la evaluación. Por lo que la generación no es automática, se requirió de la colaboración de expertos y en este estudio se logró obtener una base de datos de aproximadamente 20 mil preguntas de opción múltiple.

Debido a la manera de evaluar a los estudiantes del área de la medicina, surge la necesidad de crear exámenes con reactivos que contengan diversos conocimientos y áreas, por lo cual se propone una metodología para el desarrollo de reactivos de opción múltiple o también conocida como la metodología AIG (Automatic Item Generation).

Dicha metodología requiere de tres pasos como proceso, primeramente, un reactivo cognitivo como modelos, creado por el contenido de los especialistas de medicina y sus diferentes áreas. Como segundo paso, se desarrollan modelos de artículos utilizando el contenido del modelo cognitivo. En tercer lugar, los reactivos se generan a partir de reactivos modelos utilizando software de computadora.

Se menciona que, empleando esta metodología, se logra generar 1248 opciones de reactivos múltiples de un artículo.

Las cantidades de opciones para los reactivos son en este ámbito considerablemente mayores a los que comúnmente se requieren en un examen ordinario en un aula. En esta área de la medicina se pueden ocupar de cientos a miles de preguntas para conformar los exámenes que serán empleados a lo largo de la formación académica.

2.3 Generación automática de reactivos en el ámbito medicinal

El trabajo de Gierl, Mark J. (Gierl M. J., 2013) menciona que es de suma importancia generar reactivos automáticamente en el ámbito medicinal ya que las pruebas basadas en computadora permiten estudios bajo demanda, se acorta la duración del examen y resulta en una reducción dramática en el tiempo de prueba. Los modelos de reactivos se crean utilizando el contenido del modelo cognitivo, donde un modelo de artículo es como una plantilla, una representación o un molde de la evaluación.

En cuanto al modelo cognitivo incluye tres puntos principales, el primero identificar el problema, el segundo especificar las fuentes de información requeridas para diagnosticar el problema y describir las características principales para cada fuente de información necesitadas para crear los diferentes casos del problema.

Cuando el reactivo o ítem modelo está especificado, se combina la información sistemáticamente para producir nuevos reactivos, esto se complementa con el software creado por la Universidad de Alberta llamado IGOR (Item GeneratOR).

Finalmente, después de la creación de cinco diferentes modelos de reactivos, surgieron un total de 20896 items o reactivos generados.

Para lograr la AIG se precisó de dos elementos, donde el primero lo refieren a un tipo de arte, ya que se emplea juicio, expertiz y experiencia, mediante lo cual se identifica el conocimiento y habilidades requeridas para resolver los problemas, organizar la información dentro de un modelo cognitivo y diseñar los reactivos modelos.

Después viene el segundo elemento asociado con ciencia, donde la tecnología de la computación es requerida para la generación sistemática entre la información en cada uno de los reactivos modelos.

2.4 GenerEx

Según Ferreyra (Ferreyra & Backhoff-Escudero, 2016) en base a la idea de la generación automática de reactivos, crearon GenerEx el cual es un generador automático de reactivos para el Examen de Competencias Básicas (Excoba). Dicho trabajo tuvo el propósito de describir una propuesta para analizar la estructura interna y equivalencia psicométrica de los exámenes generados con el GenerEx, así como describir el tipo de resultados que se obtienen para lograr este cometido.

Dicho trabajo tiene como propósito el presentar una propuesta que analice y valide la gran cantidad de reactivos y exámenes generados para la selección de estudiantes que aspiran a ingresar a la Educación Media Superior.

Tal propuesta está fundamentada en la manera y forma en que se seleccionan las muestras de los reactivos modelos, teniendo como base que los ítems deben ser equivalentes psicométricamente.

Donde a quien se le aplica el examen debe seleccionar una respuesta u opción de varias alternativas propuestas, como, por ejemplo, soluciones numéricas o algebraicas, otra alternativa es la reubicación de elementos conceptuales en mapas, graficas esquemas, planos, incluso la selección múltiple en base a la construcción de la respuesta seleccionando tres o más opciones.

2.5 Producción de bibliotecas e identificación de oraciones

Los code contracts o especificaciones del código son aquellos términos que son requeridos y que se espera después de su ejecución, a pesar de su importancia, esta tarea en la mayoría de los casos no se lleva a cabo en la práctica.

Según menciona Xiao (Pandita, y otros, 2012) la reutilización de las librerías de software ayuda en las entregas a tiempo de proyectos; para determinar qué y cómo reutilizarlo, las especificaciones de las bibliotecas de software reutilizables juegan un papel importante. En ausencia de especificaciones, los desarrolladores pueden escribir código que es inconsistente con el uso adecuado de estas bibliotecas. Como resultado, no solo dicho código es de inferior calidad y contiene fallas, el costo adicional de la depuración y corregir dicho código defectuoso también podría frustrar el propósito de reutilizar el software. Y mediante en el uso del PLN se propone un nuevo enfoque para inferir especificaciones formales de las bibliotecas e identificar oraciones que describen los code contracts.

Para abordar esta problemática se propone un manejo del PLN para inferir automáticamente en las especificaciones deseadas, presentándose como desafíos la ambigüedad, palabras claves de programación, la equivalencia semántica, en donde para la última, su propuesta para manejar la semántica propone una nueva técnica llamada análisis de equivalencia basado en identificar la estructura de la gramática de los sustantivos y verbos principales de una oración.

El articulo resalta algunos aportes en los que se mencionan los más sobresalientes como lo es, una técnica que efectivamente identifique con lenguaje natural, las oraciones o texto donde contengan especificaciones para código y una técnica para inferir en las mismas; también se ofrece la implementación de un prototipo que con la utilización de Stanford Parser se deriva la estructura gramatical del texto, esto debido a la herramienta empleada.

2.6 Watson

Aunque Watson (IBM, 2020) no haga uso directamente de la nube de Google, es una API reciente que puede usar PLN para interpretar los diferentes comandos de usuario y transmitir estos comandos a diferentes piezas de software inteligente dentro del hogar, como la iluminación controlada por aplicaciones. Watson tiene documentación Swagger interactiva donde puede probar solicitudes como encender las luces, el clima y demás. Watson está diseñada para un uso más completo de PLN, se centra en facilitar la comunicación entre

humanos y el texto o discurso. Se basa en la tecnología internet de las cosas, donde el PLN se puede utilizar como el medio perfecto para comunicarse con un automóvil inteligente, o una casa.

En 2007, IBM asumió el gran desafío de construir un sistema informático que pudiera competir con los campeones en el juego de Jeopardy. En 2011, el sistema de preguntas y respuestas de dominio abierto denominado Watson venció a los dos jugadores mejor clasificados en Jeopardy en dos juegos televisados a nivel nacional. Esta cifra proporciona una descripción técnica profunda de las ideas y logros en construir Watson y finalmente lograr el cometido. Watson cubre áreas como la lingüística computacional, recuperación de información, representación del conocimiento y razonamiento, y machine learning.

2.7 Chatbots

Según Sjoera (Roggeman, 2017) en su artículo The linguistics behind chatbots, el PLN está detrás de esta tecnología que recientemente ha hecho una gran aparición con los chatbots. Matt Schlicht, fundador de Chatbot Magazine, describe un chatbot de la siguiente manera: Un chatbot es un servicio, impulsado por reglas y, a veces, Inteligencia Artificial (IA), con el que interactúas a través de una interfaz de chat.

Los chatbots funcionan con una interfaz conversacional, que comúnmente está ligada a un chat, como ejemplo se hace alusión a Madison Reed la cual es una compañía que vende productos para teñir el cabello empleado desde casa. Su chatbot le ayuda a elegir el tono de color adecuado analizando su color de cabello actual en función de una imagen y preguntándole qué color desea lograr, además de algunos consejos sobre su tinte deseado.

Hay dos tipos de chatbots los que se basan en reglas, por ejemplo, el usuario indica algo y el chatbot solo responde a comandos básicos, y por otro lado están los chatbots basados en IA los cuales son capaces de analizar el lenguaje natural del usuario, no solo los comandos. También aprende continuamente de las conversaciones que tiene con las personas, por lo que se vuelve más inteligente en su transcurso.

CAPÍTULO III. MARCO TEÓRICO

En este capítulo se muestra aquellos conceptos importantes que son de consideración para un mejor entendimiento respecto a temas como el PLN y el porqué de la elección de la API que ofrece Google.

3.1 Alternativas de herramientas para PLN

En el post de Tanya (Thakur, s.f.) se expuso un tema relacionado a la utilidad del PLN con la API de GCP (Google Cloud Platform) llamado la Comparación de los servicios de Machine Learning (ML) de varios proveedores de servicios Cloud, se presentan varios puntos de comparación entre las tecnologías existentes respecto a ML, dentro del análisis de tres aspectos clave que se muestra en el tema, como lo son la ejecución, la salida del código y el precio, se observa el nivel sobresaliente de GCP respecto al PLN, dicha comparación se muestra en la figura 1.

Features	Amazon Comprehend	Google Cloud Natural Language	Microsoft Azure Text Analytics	IBM Watson NLU
Entity Extraction	✓	✓	✓	✓
Key Phrase Extraction	✓	✓	✓	✓
Sentiment Analysis	✓	✓	✓	✓
Syntax Analysis	✗	✓	✓	✗
Topic Modeling	✓	✓	✗	✗
Multiple Language Support	100+	110+	120	✓
Parts of Speech	✗	✓	✓	✗

Figura 1. Comparativa de APIs para el PLN.

A continuación, se muestran algunas alternativas para el PLN

3.1.1 API Natural Language de Google

La API Natural Language de GCP, descubre la estructura y el significado del texto mediante modelos de aprendizaje automático en una API REST fácil de usar. Se emplea para extraer información sobre personas, lugares, eventos y muchos elementos más que se mencionen en documentos de texto, artículos de noticias o entradas de blogs.

En la figura 2 se muestra un ejemplo del análisis que el API de Google realiza, se observa las propiedades de cada palabra de la oración.

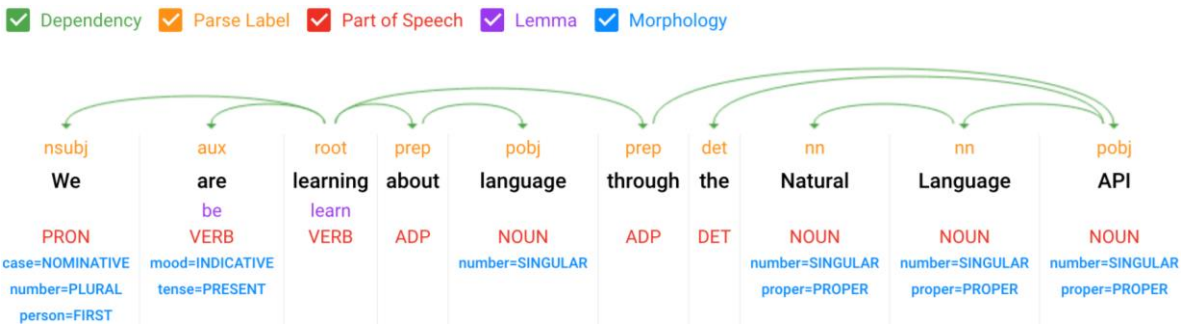

Figura 2. Ejemplo de análisis de lenguaje natural.

3.1.2 Stanford CoreNLP

Se observa que existen distintas herramientas para el PLN como es el caso del software Stanford CoreNLP (Manning, y otros, 2014) que tienen en The Stanford Natural Language Processing Group el cual proporciona un PLN en el ámbito estadístico, PLN de aprendizaje profundo y herramientas basadas en las reglas de la lingüística computacional, donde el software es de código abierto.

Puede dar las formas básicas de palabras, las partes de la oración ya sean nombres de compañías, personas, etc., normalizar fechas, horas y cantidades numéricas, marcar la estructura de oraciones en términos de frases y dependencias sintácticas, indicar qué frases nominales se refieren a las mismas entidades, indican sentimiento, extraen relaciones particulares o de clase abierta entre menciones de entidades, obtienen las citas que las personas dicen, etc.

Stanford CoreNLP integra muchas de las herramientas de PLN de Stanford, incluido el etiquetador de categorías gramaticales (POS), el reconocedor de entidades nombrado (NER), el analizador, el sistema de resolución de correferencia, el análisis de sentimientos, el aprendizaje de patrones bootstrapped y la extracción de información abierta herramientas.

El tokenizador se encarga de dividir las palabras del texto en tokens, haciendo uso de muy buenas heurísticas por lo que generalmente puede decidir cuándo las comillas simples son partes de palabras, cuándo los períodos no implican límites de oraciones, etc.

Una oración finaliza cuando el carácter que termina la oración (! o ?) no se encuentra agrupado con otros caracteres en un token (como una abreviatura o número), aunque puede incluir algunos tokens que pueden seguir a un carácter que termina la oración como parte de la misma oración (como comillas y corchetes).

El reconocimiento de las entidades que ofrece esta herramienta de Stanford se puede observar en la figura 3 con una oración como ejemplo.

Figura 3. Ejemplo de reconocimiento de entidad nombrada.

La correferencia se muestra en la figura 4 donde se observa la referencia de las palabras de dos o más expresiones lingüísticas.

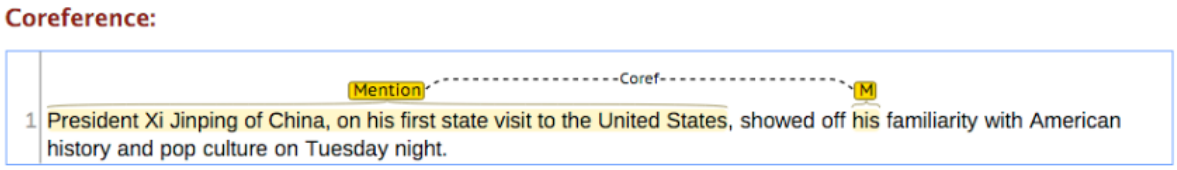

Figura 4. Ejemplo de correferencia.

En la figura 5 se observa la dependencia de las palabras para tener una adecuada estructura de la respectiva morfología, la cual muestra el orden y sentido a la oración.

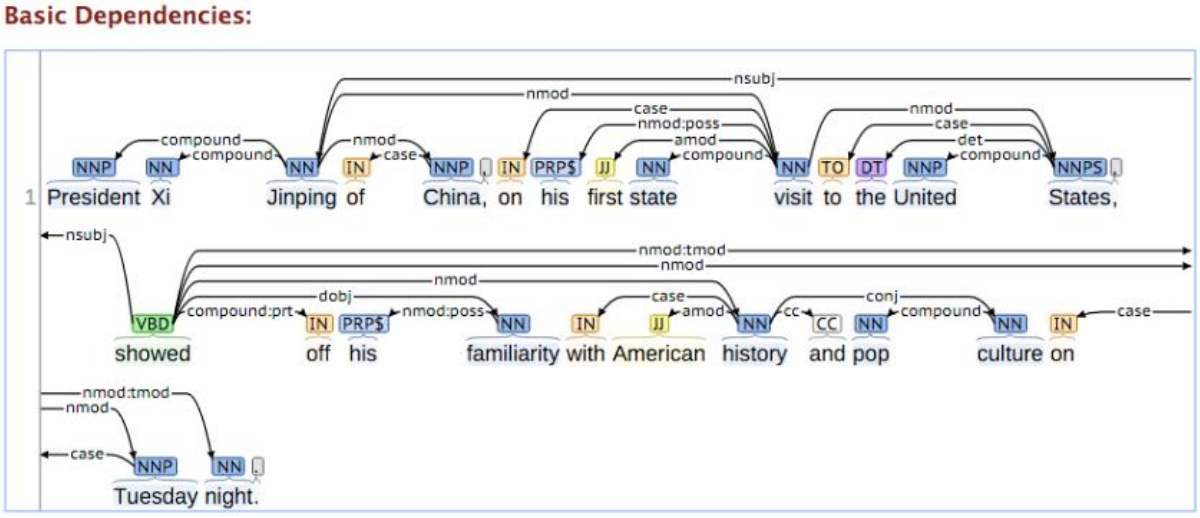

Figura 5. Ejemplo de dependencias básicas.

3.1.3 Apache OpenNLP

Apache OpenNLP (Apache OpenNLP, 2017) es otra herramienta, la cual realiza un análisis de opinión en la red social Twitter. El PLN de Apache es un conjunto de herramientas basado en el aprendizaje automático para el procesamiento de texto en lenguaje natural. Admite las tareas PLN más comunes, como la tokenización, la segmentación de oraciones, el etiquetado de categoría gramatical parcial, la extracción de entidad nombrada, la fragmentación, el

análisis sintáctico y la resolución de correferencia. Estas tareas generalmente se requieren para crear servicios de procesamiento de texto más avanzados. OpenNLP también incluyó la máxima entropía y aprendizaje automático basado en perceptrón.

Esta herramienta es utilizada actualmente por Twitter, para analizar los sentimientos detrás de los tweets que son generados día con día, dando como referencia una columna que contiene 0 o 1, donde 0 indica negativo y 1 indica positivo.

OpenNLP de Apache contiene varios componentes, que permite construir el PLN completo. Estos componentes incluyen: detector de oraciones, tokenizador, buscador de nombres, categorizador de documento, etiquetador de categorías gramaticales, chunker, analizador sintáctico, resolución de correferencia. Los componentes contienen partes que le permiten a uno ejecutar la tarea de PLN respectivo, entrenar un modelo y también evaluar un modelo. Cada una de estas instalaciones es accesible a través de su interfaz de programa de aplicación (API). Además, se proporciona una interfaz de línea de comando (CLI) para facilitar los experimentos y la capacitación.

3.1.4 Cogito Discover

Desarrollada por Expert System (Discover, 2018), ayuda a evaluar y analizar una gran cantidad de información de páginas web y redes sociales, se basa principalmente en extraer el contenido que a simple vista no se puede conocer en cuanto a la semántica de los textos.

Se utiliza para el procesamiento de lenguaje semántico y natural completo como minería de texto (con razonamiento semántico y entidades inferenciales), categorización, etiquetado semántico, sentimiento, entidad y extracción de relaciones, incluyendo personas, lugares, geografía, URL, correo electrónico, números de teléfono, moneda, y mucho más.

Cogito API ofrece una API orientada para medios y publicaciones, publicidad, finanzas y moda, incluidas taxonomías detalladas y bases de datos de entidades listas para usar, para proporcionar resultados más detallados y precisos.

Mediante la identificación de entidades y razonamiento semántico Cogito Discover es una API empleada por algunas empresas de renombre con el fin de solucionar ciertas demandas existentes en cuanto a la manipulación de los datos se refiere.

Esta API hace uso de Google Platform, la herramienta con la cual esta tesis se desarrolla cabe mencionar que dicha empresa menciona algunos de los beneficios de utilizar esta plataforma de Google, los cuales se considera importantes para el proyecto de tesis, entre ellos se encuentra:

- Desambiguar y comprender el contenido para organizar y clasificar.
- Aplicar el razonamiento semántico donde el contexto ofrecido por el lenguaje supera cualquier intento de inferencia.
- Adquirir y asociar automáticamente contenido vinculado externamente basado en la relevancia basada en el contenido, simplemente definiendo la fuente o fuentes externas.

Cogito combina el poder de una API de PLN con tecnología semántica para analizar y procesar grandes volúmenes de contenido no estructurado a la velocidad de la tecnología de vanguardia.

Dicha empresa hace alusión a la forma rápida y automática con la que se puede extraer el conocimiento e información acerca de un texto, además de ser capaz de manipular grandes volúmenes de contenido.

3.1.5 Cogito Studio

En la página web de Expert System (Expert System, 2020) se aplica la interpretación semántica al lenguaje y la lógica programática para lograr una comprensión precisa de algún contenido. Esta poderosa capacidad brinda control total de cómo acceder y usar el contenido que más importa.

Cogito Studio brinda eficiencia y precisión al flujo de trabajo de gestión del conocimiento y admite el desarrollo de funciones basadas en la comprensión del lenguaje natural. Cogito Studio permite aprovechar las ricas características semánticas integradas en el.

Este software cuenta con escritura avanzada de reglas lingüísticas para apoyar reglas de extracción efectiva y categorización automática. Cuando sea necesario, simplifica y automatiza el proceso de escritura de reglas a través de la integración con algoritmos de aprendizaje automático.

Cuenta con capacidades avanzadas para aumentar y enriquecer el conocimiento gráfico personalizado a través de la adquisición de Subject Matter Expert Knowledge, así como del aprendizaje automático.

Así mismo cuenta con sofisticadas herramientas de prueba y generación de informes para validar las reglas de proyectos contra una colección de documentos. Esto permite medir regresiones y mejoras para precisión y recuperación. Como resultado, los resultados se pueden mejorar rápidamente para coincidir con las expectativas de calidad.

3.1.6 Amazon Comprehend

Amazon desarrolló un servicio del PLN (Amazon, 2020), mediante la extracción de palabras claves, como lugares, personas, marcas, logra comprender si el texto tiene un estado positivo o negativo.

La aplicación de esta API lanzada por Amazon permite detectar el significado y las relaciones del texto respecto del servicio de atención al cliente, búsqueda inteligente de texto.

Hacer uso de esta tecnología permite analizar una colección de documentos al mismo tiempo, como se menciona en las redes sociales para conocer la opinión de la gente, que es lo que se dice acerca de tal empresa, aprende de las descripciones de productos y opiniones de clientes de Amazon; dicho ejemplo se ilustra en la figura 6.

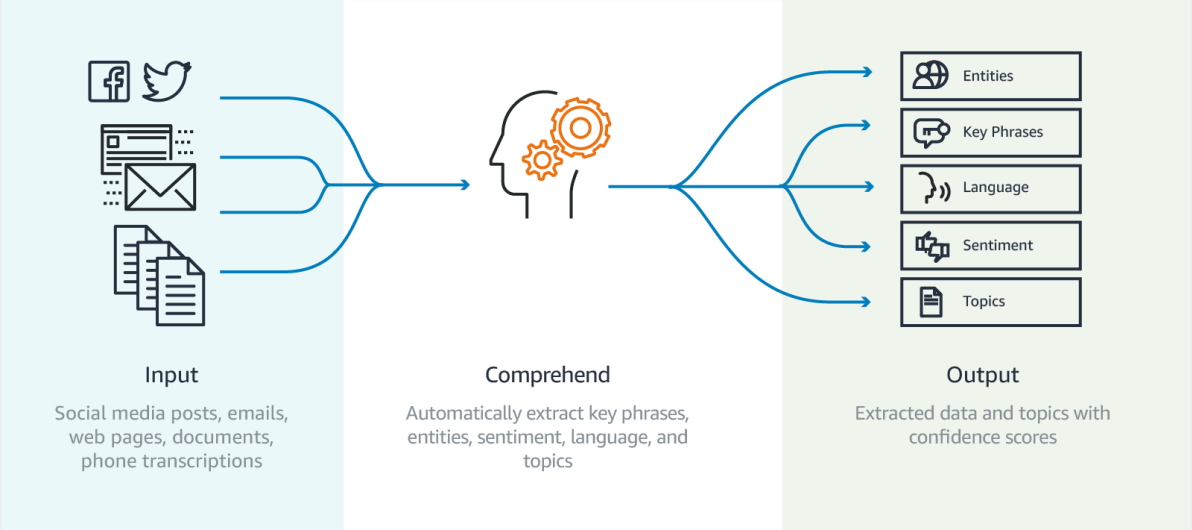

Figura 6. Ejemplo de funcionalidad del API de Amazon Comprehend.

3.1.7 FreeLing

El software FreeLing (Drupal, 2020) es una biblioteca en C++ el cual permite el análisis morfológico, detección de entidades nombradas, etiquetado PoS, análisis sintáctico, desambiguación de sentido de palabras, etiquetado de funciones semánticas haciendo uso de diversos idiomas.

El diseño está disponible para ser empleado en una biblioteca externa desde cualquier aplicación que necesite de los servicios que esta ofrece, además proporciona un programa principal de línea de comandos.

Los resultados que muestra FreeLing son análisis lingüísticos en una estructura de datos, incluye el analizador, lo cual permite al usuario final de habilidades de programación para análisis de un texto.

El módulo de tokens o tokenizer, convierte texto plano en un vector de objeto de palabras, de acuerdo al conjunto de reglas de tokenización.

Una de las importantes funciones de esta API es el poder distinguir el lenguaje de entrada. Compara el texto dado con los modelos disponibles para diferentes idiomas y devuelve el idioma más probable en el que está escrito el texto.

3.1.8 TextRazor

El software de TextRazor (TextRazor, 2019) utiliza técnicas de PLN e inteligencia artificial de última generación para analizar y extraer metadatos semánticos de su contenido, permite extraer toda la información que necesita en una sola solicitud, vinculando los metadatos semánticos extraídos para facilitar la identificación de patrones complejos.

Escrito en C ++ altamente optimizado, TextRazor es capaz de procesar miles de palabras por segundo, permite agregar nombres de productos, personas, empresas, reglas de clasificación personalizadas y patrones lingüísticos avanzados. El motor Prolog integrado permite combinar rápidamente los resultados de TextRazor con una sólida lógica específica de dominio personalizado.

TextRazor logra un rendimiento de reconocimiento de entidades líder en la industria mediante el aprovechamiento de una gran base de conocimiento de los detalles de la entidad extraídos de varias fuentes web, incluyendo Wikipedia, DBPedia y Wikidata.

El cual ha construido un diccionario de millones de diferentes entidades posibles, que se puede buscar rápidamente en un texto utilizando el motor de correspondencia.

Se emplea un etiquetador estadístico para identificar personas, lugares y empresas que nunca se han mencionado antes, y expresiones regulares para detectar las menciones menos ambiguas, como direcciones de correo electrónico y sitios web.

Se menciona que el reconocimiento de entidades es una tarea difícil debido a la ambigüedad del lenguaje escrito, cabe destacar que es la primera API en la que se encuentra esta característica de dificultad. Se muestra un ejemplo de la ambigüedad existente en la palabra "Lincoln" que por ejemplo podría usarse para referirse al presidente, un tipo de automóvil, un lugar en Inglaterra, un lugar en los Estados Unidos, etc. Tal situación TextRazor la aborda

de una manera única ya esta situación es muy común, y la única forma de hacerlo correctamente para lograr la desambigüedad es con un modelo integral de conocimiento del mundo real de cada entidad. Ejemplo en la figura 7.

```
entities: [
        0: {type:[Country, Place, owl#Thing, PopulatedPlace], matchingTokens:[0], entity
Id:Spain,}
        1: {type:[Organization, Company, Organisation, owl#Thing], matchingTokens:[3], e
ntityId:Bankia,}
        2: {matchingTokens:[10], entityId:Portfolio (finance), confidenceScore:1.11858,}
        3: {matchingTokens:[20, 21], entityId:Parent company, confidenceScore:1.24656,}
        4: {type:[owl#Thing, Company, Airline, Organisation, Organization], matchingToke
ns:[23, 24],}
        5: {matchingTokens:[26], entityId:Iberia (airline),}
]
```

Figura 7. Etiquetador estático TextRazor.

3.1.9 AYLIEN

La API de Aylien (Aylien, 2020) cuenta con un PLN basado en la nube para realizar una variedad de tareas en documentos, revisiones, comentarios sociales o cualquier otro tipo de texto.

Esta herramienta permite procesar URLs, analizar en tiempo real, se extrae lo importante de documentos, páginas web, blogs, tweets y reseñas, o cualquier contenido de texto. Se entiende qué o quién menciona un documento, qué temas trata y cuál es el sentimiento de un texto.

3.2 Elección del API de PLN Google

Observaciones importantes para la selección de la API son resultados de la comparación que realiza Brian (Jackson, 2019).

Mejor precio que el de la competencia

Google factura en incrementos de minutos (con un cargo mínimo de 10 minutos), por lo que sólo paga por el tiempo que realmente utiliza. Y una gran ventaja es que se dan precios con

descuento para las cargas de trabajo de largo plazo sin ningún compromiso inicial requerido. Simplemente utilice las máquinas virtuales durante un mes y obtendrá un descuento. Esto lo hace perfecto para las nuevas empresas y para las TI de la empresa para reducir los costos. AWS, por ejemplo, requiere prepagos en forma de Instancias reservadas para ser elegible para descuentos. Y Azure sólo ofrece un 5% de descuento en caso de un prepago de 12 meses.

AWS vs Google Cloud

Cuando se trata de GCP básicamente obtiene más IOPS (Input/Output Operations Per Second) por menos de un tercio del costo. En la figura 8 se presenta la comparativa con AWS, se paga $1,102.50/mes con un contrato de 3 años, contra los $470.64/mes de Google Cloud. IOPS es una medida para las operaciones de entrada/salida por segundo y la frecuencia con la que un dispositivo puede realizar tareas de E/S. En la mayoría de los casos, cuanto mayor sea el número de IOPS, mejor será el rendimiento.

```
AWS - Virginia                      GCP - South Carolina (East)

c4.4xlarge                          Custom
  - 16 CPUs                           - 16 CPU
  - 30 GB RAM                         - 30 GB RAM
======================              ======================
+ $613.42 instance                  + $357.25 instance (sustained-use)

500GB SSD EBS Volume                667GB SSD PD Volume
======================              ======================
+ $62.50 disk                       + $113.39 disk
+ $1040.00 IOPS (16K IOPS)          + $000.00 IOPS (20K IOPS)

======================              ======================
= $1715.92/month on-demand          = $470.64/month
= $1102.50/month 3yr ($8,580.00)
```

Figura 8. Precios de AWS vs Google Cloud.

Azure vs Google Cloud

Como Sandeep (Dinesh, 2016) señala, se tiene que adjuntar almacenamiento SSD Premium para obtener el mismo almacenamiento de red persistente conectado. Pero una vez más, cuando se haga una comparación entre los dos, el costo es menor en un 33% utilizando GCP. En la figura 9, con Azure tiene un costo de $ 1.602,68/mes en comparación con Google Cloud $532.82/mes.

```
Azure – East                        GCP – South Carolina (East)

D5 V2                                 Custom
  – 16 CPUs                           – 16 CPU
  – 56 GB RAM                         – 56 GB RAM
=======================             =======================
+ $870.48 instance                  + $419.43 instance (sustained-use)

512GB Premium Volume                667GB SSD PD Volume
=======================             =======================
+ $73.22 disk (2.3K IOPS)           + $113.39 disk
X 8 RAID0 IOPS (18.4K IOPS)         + $000.00 IOPS (20K IOPS)

=======================             =======================
= $1602.68/month                    = $532.82/month
```

Figura 9. Precios de Azure vs Google Cloud.

Como Christopher (Alghini, 2018) menciona, Google es pionero en el mundo de la informática en la nube, el análisis y la inteligencia artificial, el cual ha adoptado su propio enfoque en el mundo del PLN.

El servicio de lenguaje natural analiza el significado y la estructura del texto a través de los modelos de aprendizaje de máquina REST API fáciles de usar. Se puede utilizar esta solución para extraer información importante de documentos de texto, artículos y textos académicos, e incluso llegar al fondo del sentimiento del cliente.

Mediante el PLN se puede descubrir el contexto detrás de lo que dice la gente en las redes sociales y determinar la intención de las conversaciones con los consumidores también. Esencialmente, GCP del PLN, accede a las soluciones de aprendizaje automático utilizadas por el Asistente de Google y la Búsqueda de Google, y le proporciona esa potencia para que pueda realizar un análisis sintáctico y de sentimiento completo.

Algunas de las características sobresalientes en comparación a otras APIs se listan a continuación:

- Análisis y segmentación de oraciones: los sistemas de PLN pueden analizar partes de una oración para ofrecer una mejor comprensión de la oración completa.
- Análisis profundo: los sistemas PLN pueden aplicar técnicas de procesamiento avanzadas a información extra específica de conjuntos de datos de múltiples fuentes.
- Extracción de entidad con nombre: en un proceso de minería de datos, una entidad con nombre puede ser descubierta y analizada por sistemas de PLN.
- Traducción automática: las soluciones PLN se utilizan cada vez más para los programas de traducción, en los que un idioma se traduce automáticamente a otro, por ejemplo, cuando se usa la función de traducción de Google.
- Resumen automático: el PLN se puede usar para crear un resumen de una enorme pieza de texto.
- El análisis de los conocimientos del cliente: se extrae información útil del texto sobre cómo se sienten sus clientes sobre su marca y qué esperan de ella.
- Clasificación de contenido: puede construir gráficos de relación examinando cómo las personas en su red responden a diferentes tipos de contenido, desde noticias hasta artículos.
- Aprendizaje automático: la API de PLN de Google utiliza la misma tecnología en aprendizaje automático que potencia la solución de búsqueda de Google y el Asistente de Google.

Lenguaje natural

“Un lenguaje natural es aquel que ha evolucionado con el tiempo para fines de comunicación humana. El lenguaje continúa su evolución sin considerar la gramática, cualquier regla se desarrolla después del desarrollo de este. En contraste, los lenguajes formales están definidos por reglas preestablecidas, y por tanto se rigen con todo rigor a ellas.” (Vásquez, Quispe, & Huayna, 2009)

Los lenguajes formales se utilizan para transferir información, sin dejar lugar a ambigüedades, como por ejemplo Java, PHP y C. En la actualidad las computadoras pueden procesar los lenguajes formales sin problemas, pero uno de sus principales retos es entender el lenguaje natural. Con este propósito, hay un área informática dedicada a la interacción entre ordenadores y a las lenguas habladas por los humanos la cual es el PLN.

El lenguaje natural se utiliza a diario como medio de comunicación entre humanos. El castellano, el inglés o el francés son ejemplos de lenguaje natural entendidas como lenguas las cuales poseen una sintaxis y una gramática.

Procesamiento de lenguaje natural (PLN)

El PLN según en la Revista de investigación de Sistemas e Informática (Cortez Vásquez, Vega Huerta, Quispe Pariona, & Huayna, 2014) se emplea el uso de un lenguaje natural para lograr la comunicación con la computadora, para que esta pueda entender las oraciones que se le proporcionan, los lenguajes naturales hacen más fácil el desarrollo de programas que realizan tareas y ayudan a comprender los mecanismos humanos relacionados con el lenguaje.

El PLN es una parte esencial de la IA que investiga y formula mecanismos computacionalmente efectivos que faciliten la interrelación hombre-máquina y permitan una comunicación mucho más fluida y menos rígida que los lenguajes formales y sistemas de menús utilizados tradicionalmente.

Todo sistema de PLN intenta simular un comportamiento lingüístico humano; para ello debe tomar conciencia tanto de las estructuras propias del lenguaje, como de un conocimiento general acerca del universo de la oración. De esta forma, una persona que participe en un dialogo sabe cómo pueden combinar las palabras para formar una oración, conoce los significados de estas, sabe cómo éstas afectan el significado global de la oración y poseen un conocimiento del mundo en general que permite participar de la conversación.

Las lenguas humanas pueden expresarse por escrito (texto), oralmente (voz) y también mediante signos. Naturalmente, el PLN está más avanzado en el tratamiento de textos, donde hay muchos más datos y son más fáciles de conseguir en formato electrónico.

Los lingüistas recogen colecciones de ejemplos y datos (corpus) y a partir de ellos se calculan las frecuencias de diferentes unidades lingüísticas (letras, palabras, oraciones) y su probabilidad de aparecer en un contexto determinado. Calculando esta probabilidad, se puede predecir cuál será la siguiente unidad en un contexto dado, sin necesidad de recurrir a reglas gramaticales explícitas.

3.3 Formulación de una pregunta

Dentro del español existen 4 categorías que agrupan las palabras interrogativas (Paitamala, 2016). Su categorización de puede observar en la tabla 2.

Tabla 2. Categorización de palabras interrogativas

	Qué	Cuánto	Quién	Cuál	Dónde	Cómo	Cuándo	Proposición +interrogativo
Determinantes Interrogativos								
Pronombres interrogativos								
Proformas interrogativas								
Formas compuestas								

Ejemplos de interrogativas:

1. Determinantes interrogativos ("adjetivos interrogativos")

    Qué: ¿Qué batallas se libraron?

    Cuánto, -a, -os, -as: ¿Cuántas guerras se lucharon ahí?

2. Pronombres interrogativos

    Quién (personal): ¿Quién dijo eso?

    Qué (general, indefinido): Qué batalla aconteció?

    Cuál (general, definido): ¿Cuál falta?

3. Proformas interrogativas ("adverbios interrogativos")

    Dónde (lugar): ¿Dónde sucedió?

Cuándo (tiempo): ¿Cuándo sucedió?

Cómo (manera): ¿Cómo te vas?

Cuánto (cantidad): ¿Cuánto es mucho?

4. Formas compuestas

Preposición + interrogativo: ¿Con quién lucho?, ¿Desde qué lugar?, ¿Hasta cuándo sucedió?

Las funciones principales más comunes de una pregunta son evaluar el conocimiento y provocar un pensamiento (S. Bloom, 1956) en esto se pretende entender el propósito evaluativo y asegurar que la persona conteste, realmente logre dicho pensamiento, si no, se considera una mala pregunta.

Dentro de la formulación de una pregunta nos encontramos con dicha taxonomía la cual muestra seis tipos de pregunta.

1. Preguntas de conocimiento (hechos, definiciones).

- Ejemplo: ¿Quién hizo el primer libro de paco el chato? ¿Qué ocurrió el 5 de mayo 1862?

2. Preguntas de comprensión (ideas principales, comparaciones).

- Ejemplo: ¿Cuál es el tema principal de este tema? ¿Cuál es la diferencia entre una tesis y una ponencia?

3. Preguntas de aplicación (aplicación de conocimientos, reglas y normas).

- Ejemplo: Resuelve el siguiente espacio en blanco. De acuerdo con las siguientes instrucciones.

4. Preguntas de análisis (motivos, causas y consecuencias).

- Ejemplo: ¿Cuál es la opinión de la gente respecto a la economía actual? ¿Por qué las nuevas generaciones prefieren la tecnología desde infantes?

5. Preguntas de síntesis (generalizaciones, predicciones, nuevas soluciones).

- Ejemplo: ¿Qué pasaría en el mundo si no existiera pobreza? ¿Cómo podríamos erradicar la inseguridad?

6. Preguntas de evaluación (opiniones, valoraciones, juicios).

- Ejemplo: ¿Qué punto de vista tienes al respecto de esta sesión? ¿Cuál IDE consideras es el mejor y por qué?

"La Pedagogía de la Pregunta es un componente de la Educación Nueva, que implica no sólo innovar programas, libros, estructuras escolares, sino también rescatar el papel crítico y constructivo de la pregunta. Las preguntas constituyen un instrumento fundamental en la formación del carácter, el desarrollo de la inteligencia y el cultivo de las relaciones de afecto y mutuo respeto de maestros y alumnos. Es difícil imaginar una materia o situación pedagógica donde no pueda aplicarse este método, que ofrece la posibilidad de participación creativa a los estudiantes." (Habed López, 2012)

En el documento de *Método para la extracción de información estructurada desde textos* (Rodríguez Blanco & Simón Cuevas, 2013) se menciona una similitud en cuanto a esta tesis, ya que dicho documento trata la extracción de información del texto de una forma semejante a la que el PLN lo hace, ellos mencionan un método de construcción de mapas conceptuales a partir de texto en el que se incluyen las siguientes tareas:

1. Extracción de Texto Plano,
2. Segmentación del Texto
3. Extracción de tokens
4. POS Tagging (etiquetado de PartOfSpeech)
5. Reconocimiento de elementos Centrales Candidatos (conceptos y enlaces candidatos)

6. Intérprete de dependencias
7. Constructor del MC

Como se puede observar la similitud de procesos es muy parecida, pero lo que este tema aporta a la tesis presente es justamente como tener un correcto método de extracción de texto, el cual, en suma, de funciones importantes del API de Google, sirve para establecer un conjunto de criterios más claro de cómo se debe estructurar una correcta pregunta que sea adecuada con enfoque solido a un cuestionamiento.

El cómo se estructura una pregunta es muy importante, debido a que como se menciona al inicio de este apartado, se propicia al buen pensamiento de la persona que realiza el cuestionario, y para esto es importante un sentido conceptual.

La extracción de información del texto de Blanco y Cuevas (Rodríguez Blanco & Simón Cuevas, 2013) presenta un modelo extraído de *How humans summarize text using concept maps* (Montenegro, 2010) para la identificación de conceptos, en este caso de cada token que se sustrajo del análisis del texto en cuanto a los tokens se refiere, dentro del procesamiento.

Dicha extracción arroja sustantivo común y propio, adjetivo, adverbio, verbo, determinante, entre otros valores que la API de PLN de Google nos proporciona.

En casi la mitad de los lenguajes (Tomlin, 2014) tales como sánscrito, hindi, griego antiguo, latín, japonés, coreano, sujeto-objeto-verbo es la secuencia o el orden para predicados, pero para la otra mitad donde se encuentra el español, sujeto-verbo-objeto es la secuencia empleada para obtener el predicado, el cual expresa lo que la oración dice del sujeto.

Como menciona en el libro *Natural Lenguage Processing - IJCNLP 2005* (Robert Dale, 2005) la forma de extraer una pregunta del texto se basa en un conjunto de reglas.

De una pregunta, las palabras “quién", "qué", "cuándo", "dónde", "cómo" o "por qué", están etiquetadas como determinante o palabras de pregunta adverbial.

De acuerdo con el resultado de etiquetado POS y fragmentación de frase o tokenización, también decidimos el verbo principal y la voz de la pregunta (pasiva o activa).

Un verbo infinitivo, es un verbo que tiene un pequeño contenido semántico y forma un predicado con alguna expresión adicional, que generalmente es un sustantivo, forma un predicado en expresiones regulares, sustantivo, dar, hacer, verbo que solo tiene un significado en sí mismo.

Luego, se aplican las siguientes reglas ampliadas de extracción de palabras para formar patrones de preguntas:

Regla 1: Palabra de pregunta en un fragmento de la longitud más de uno.

> Pregunta = palabra de pregunta + palabra principal en el mismo fragmento

Regla 2: Palabra de pregunta seguida de un verbo ligero y un sustantivo frase o un fragmento preposicional frase.

> Pregunta = palabra de pregunta + verbo infinitivo + palabra principal en el siguiente fragmento de sustantivo o preposición.

Regla 3: Palabra de pregunta seguida inmediatamente por un verbo

> Pregunta = palabra de pregunta + palabra principal en el siguiente fragmento (verbo frase o sustantivo)

Regla 4: Palabra de pregunta seguida de un verbo pasivo

> Pregunta = palabra de pregunta + es, ser, será, fue, etc. + palabra clave en el fragmento de verbo pasivo

Regla 5: Palabra de pregunta seguida por el conjugante de es, ser, será, fue, etc y un sustantivo de frase

> Pregunta = palabra de pregunta + es, ser, será, fue, etc. + palabra clave en el siguiente fragmento de sustantivo

Regla 6.- Si ninguna de las reglas anteriores es aplicable, el patrón de pregunta es la pregunta.

Mediante la extracción de la información lingüística de POS y fragmentos. Se puede formar el patrón de pregunta. Estas reglas heurísticas son intuitivas y fáciles de entender. Además, el hecho de que estos patrones que tienden a repetirse, implican que son generales y fácil de recopilar datos de entrenamiento en consecuencia. Estos patrones de preguntas también indican una preferencia para que la respuesta se clasifique con un tipo de sustantivos propios.

3.4 Metodología para el PLN

Se muestra las bases de la metodología para el procesamiento del texto.

Según menciona Sjoera, colaboradora de ICAPPS (Roggeman, 2017) empresa dedica al desarrollo de software, el proceso de PLN consiste en aproximadamente 5 pasos.

1. El primer paso es el análisis léxico. El léxico de un idioma es, simplemente, una colección de palabras y frases en un texto. Como primer paso, la computadora analizará el texto y lo dividirá en párrafos, oraciones y palabras.
2. El segundo paso es el análisis sintáctico: la computadora analiza el rol gramatical de cada palabra en una oración e identifica la relación entre cada palabra. Busca tema de la oración y predicado.
3. En el tercer paso, el análisis semántico, la computadora comprueba el significado intrínseco de las palabras, lo que significa buscar el significado de las palabras. Una palabra puede tener varios significados, por lo que la computadora también necesita asignar esto con las estructuras sintácticas analizadas en el paso anterior para derivar el significado correcto.
4. El cuarto paso es la integración del discurso, lo que significa mirar el significado de una oración en comparación con la oración que viene antes. Podemos suponer que hay cohesión entre las diferentes oraciones en un texto, por lo que el PLN también debe tener esto en cuenta.
5. Finalmente, está el análisis pragmático, que también es el paso más difícil para una computadora. El análisis pragmático implica reinterpretar lo que se dice como lo que

realmente se quiere decir. Esto implica tomar en cuenta el conocimiento del mundo real porque, como humanos, lo que decimos no siempre es lo que queremos decir. Tomemos por ejemplo la oración: Hay cerveza en la nevera. Si le dices esto a un invitado que ingresa a tu casa, no estás simplemente describiendo el contenido de tu refrigerador, en realidad estás ofreciendo una bebida. Esta ambigüedad es difícil de manejar para una computadora.

Para usar PLN en la práctica, estos cinco pasos se vierten en algoritmos informáticos. A través de los algoritmos, el lenguaje del usuario se tokeniza y etiqueta, primero, cada palabra y signo de puntuación se define como un token, después, cada token se etiqueta para indicar qué tipo de token es, por ejemplo, un sustantivo, adjetivo, verbo, etc. Los tokens se analizan para definir su función en la oración, que corresponde al análisis sintáctico.

3.5 Metodología de vida del software "Desarrollo Iterativo"

La metodología en cascada propuesta por Winston (Winston, 1987) es un modelo lineal de diseño de software que emplea un proceso de diseño secuencial. Se requiere que antes de cualquier desarrollo se cuente con la claridad de la visión y el plan. El desarrollo fluye secuencialmente desde el punto inicial hasta el punto final, con varias etapas diferentes: análisis, diseño, construcción, pruebas, implementación y mantenimiento.

El modelo Iterativo (Sommerville, 2005), también llamado Incremental, permite crear una iteración o un ciclo en el que se repite cada una de las etapas del modelo en cascada hasta lograr una versión mejorada del proyecto.

Cada iteración es un ciclo de perfeccionamiento, y no necesariamente se refiere a una iteración entregable, la idea es que en cada etapa o ciclo se tenga como resultado una versión más completa e integra respecto al producto final deseado.

En la figura 10 se muestra un ejemplo del proceso de desarrollo iterativo, el cual consta de figuras semejantes a las de un diagrama de flujo para ilustrar el seguimiento de dicho proceso.

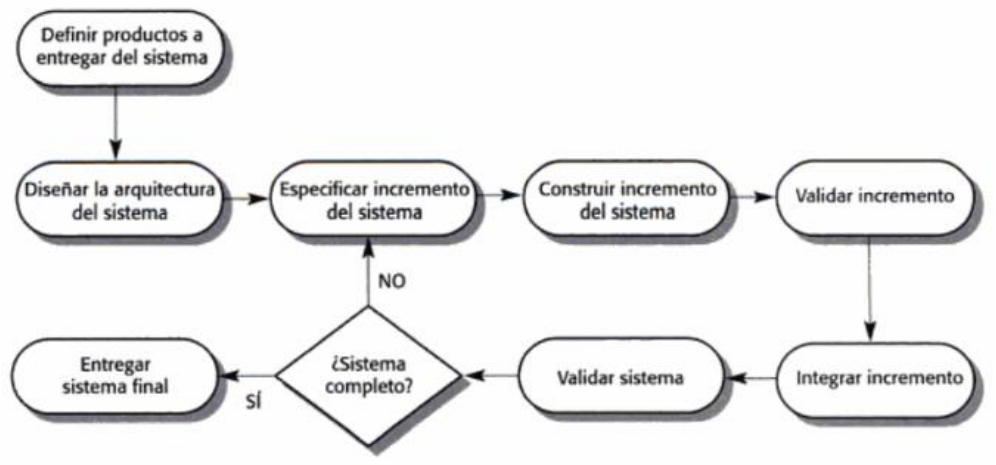

Figura 10. Ejemplo de Proceso de desarrollo Iterativo.

Específicamente el desarrollo iterativo pretende repetir las siguientes etapas en forma secuencial hasta lograr la versión final deseada.

Fase de requisitos: en la fase de requisitos del desarrollo de software, se recopila y analiza la información relacionada con el sistema. Los requisitos recopilados se planifican en consecuencia para desarrollar el sistema.

Fase de diseño: para esta fase, la solución de software está preparada para satisfacer las necesidades del diseño. El diseño del sistema puede ser uno nuevo o la extensión de uno anterior.

Implementación y prueba: en esta etapa, el sistema se desarrolla codificando y construyendo la interfaz de usuario y los módulos que luego se incorporan y prueban.

Fase de revisión: La fase de revisión es donde el software se estima y verifica según el requisito actual o lo deseado. Luego, se revisan también los requisitos adicionales para proponer una actualización en la próxima iteración.

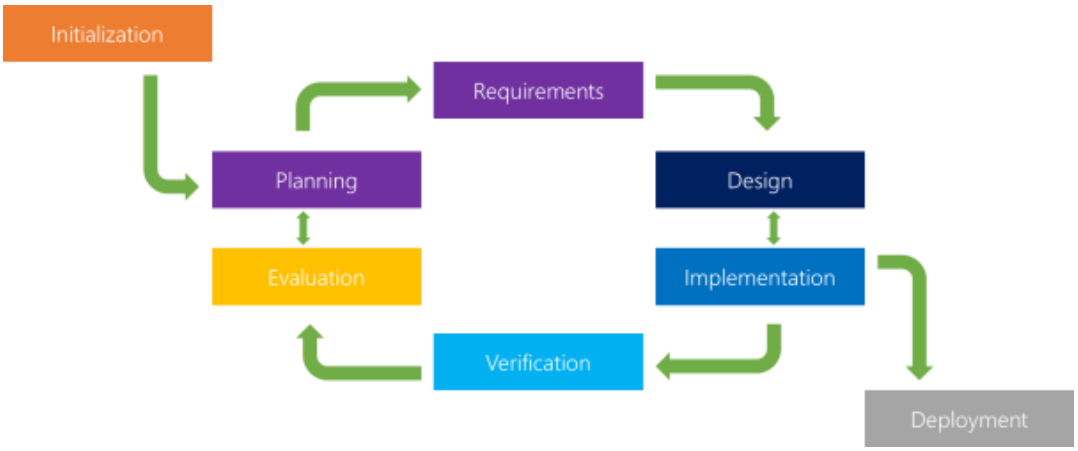

Figura 11. Modelo Iterativo.

CAPÍTULO IV. DESARROLLO

En este capítulo se muestra el desarrollo del software y los pasos previos para trabajar con la API de Google del PLN.

4.1 Selección de herramientas

El proyecto se realizó con PLN y se empleó la API que ofrece Google para tal procesamiento. Se efectuaron pruebas de la librería con una muestra de los contenidos de los libros de primaria de la SEP.

El entorno de desarrollo que se utilizó es el IDE de NetBeans (Apache NetBeans, 2020), brinda el lenguaje Java empleado en el proyecto, el cual es leído por un intérprete, permite la modulación, además de contar con la herramienta Maven (Apache Maven, 2020) para la gestión y construcción de proyectos en dicho lenguaje.

4.2 Registro para la API de PLN que ofrece Google

Primeramente, se obtuvo una cuenta Google Cloud para después configurar un proyecto en Cloud Platform, una vez que se cuenta con las credenciales de la cuenta se procedió a seguir los pasos del punto número dos que se muestra en la figura 12.

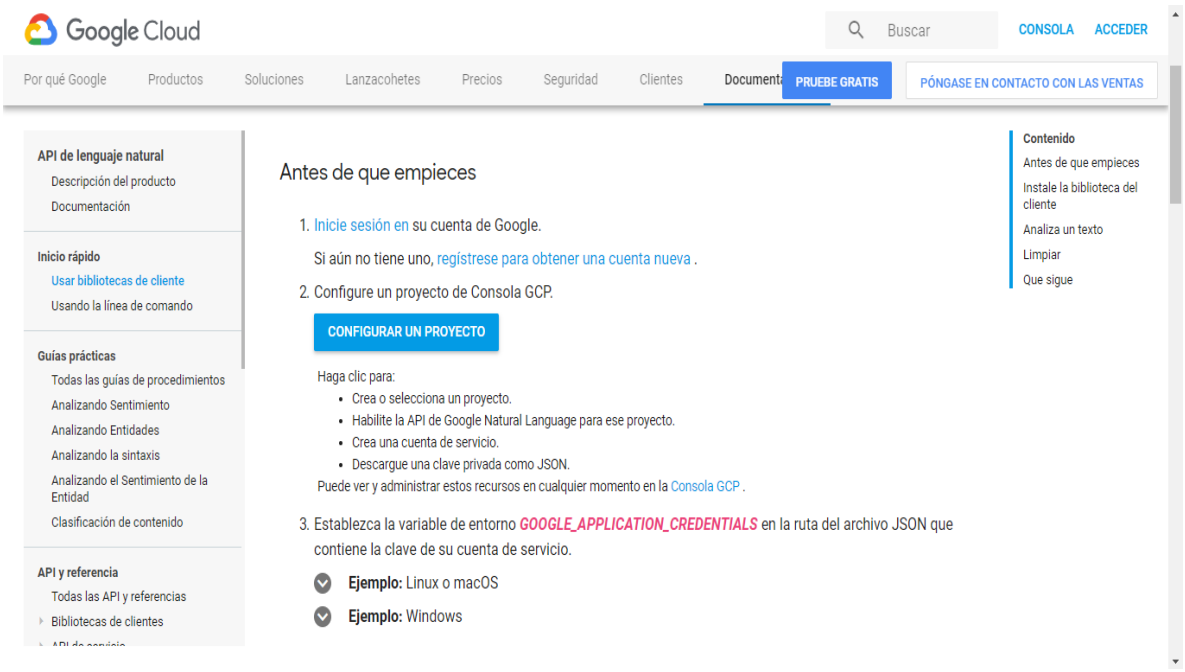

Figura 12. Configuraciones para proyecto de Google Cloud.

Una vez configurado el proyecto se descargó la clave privada en formato JSON, tal como se muestra en la figura 13.

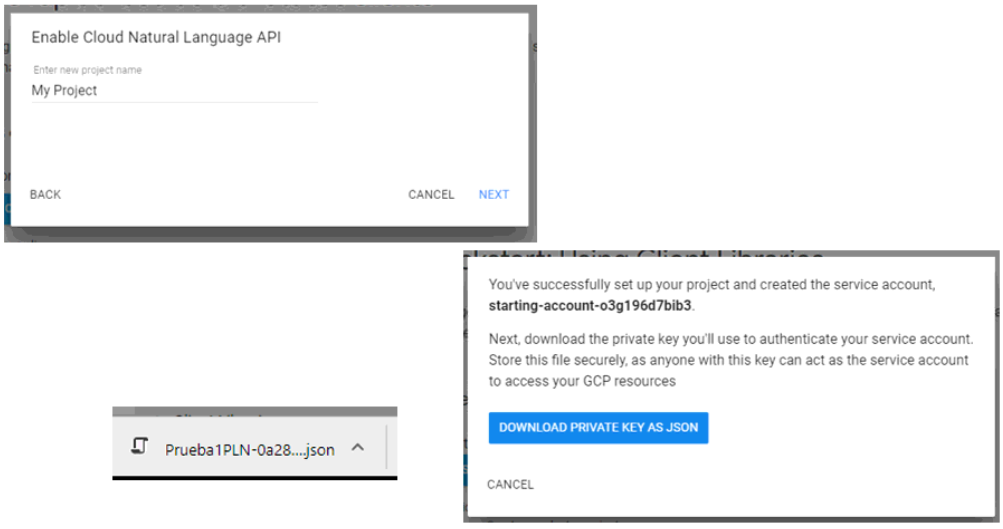

Figura 13. Generación de proyecto y llave privada.

Dicha llave se agregó como variable del sistema para que el proyecto se comunique con la API de Google; se muestra en la figura 14.

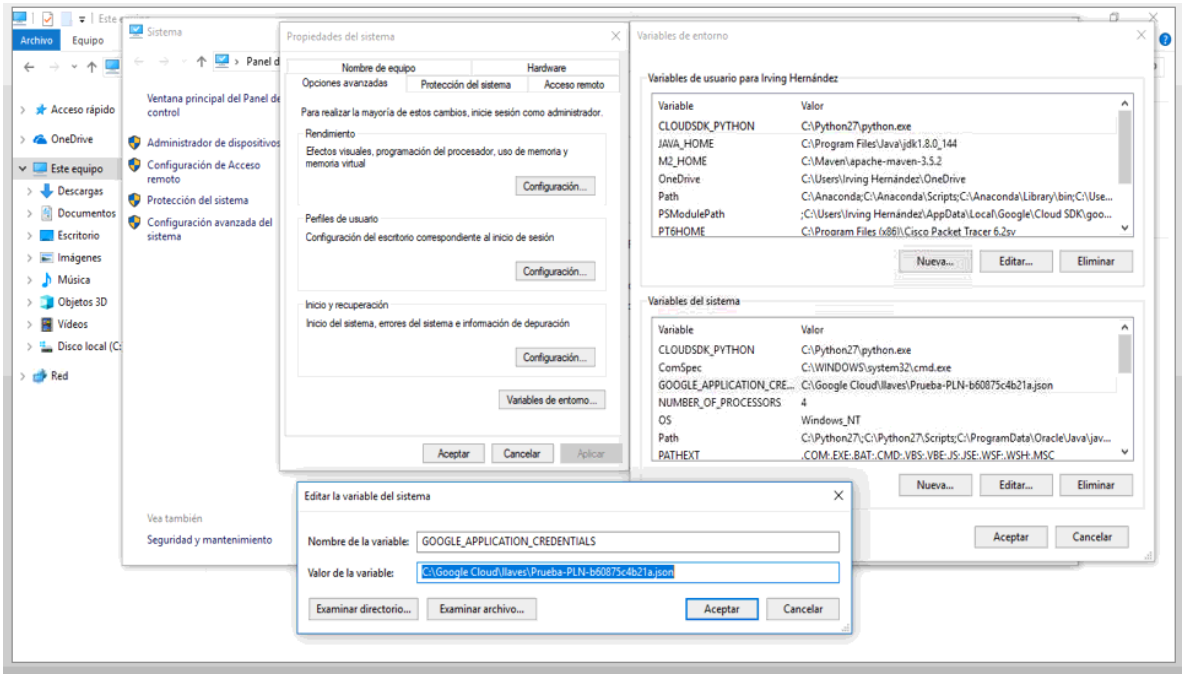

Figura 14. Llave privada como variable de entorno.

4.3 Desarrollo Iterativo o Incremental para el proyecto

Para este proyecto se implementó la metodología incremental (Sommerville, 2005) ya que el proceso de desarrollo es una situación variable en la que se presentan cambios constantes, dichas modificaciones surgen con la necesidad que el software trabaje de la mejor manera posible, satisfaciendo las expectativas deseadas.

4.3.1 Inicialización

Se considera como planificación preliminar al desarrollo del proyecto. Aquí establecimos las necesidades que se consideran sean satisfechas, y a través de una manera global, visualizar los puntos de cada parte del proyecto.

4.3.2 Planificación

Para esta etapa se consideró realizar un estudio previo para el enriquecimiento de los temas que aborda este proyecto, así como establecer los objetivos específicos que se debieron realizar para el éxito de este.

Se consideraron algunos puntos, como lo es la cantidad de respuestas generadas por cada pregunta, los criterios que fueron tomados para establecer la relación de las respuestas correctas con sus homologas, la consideración que el análisis excluyera imágenes y formulas o ecuaciones matemáticas, también los criterios que debía tener una oración para poder extraer la pregunta de dicho texto.

4.3.3 Requisitos

Esta etapa fue importante para determinar las herramientas y el conocimiento que debíamos cumplir para poder culminar el proyecto.

En cuanto a herramientas se refiere, destaca la utilización del API de PLN ofrecida por Google para el análisis del texto, el lenguaje de programación basado en Java para poder hacer uso de Maven y de la documentación existente del PLN provista en la página oficial de Google.

Así como considerar todo lo necesario conforme fuera surgiendo la necesidad de realizar cambios en el código y en la forma en la que se estuvo trabajando referente a la correcta interpretación del análisis del texto.

4.3.4 Diseño

Aquí se contempló la relación que debía existir entre las clases Java, sus métodos y propiedades. Para esto se diseñó un diagrama de clases y una representación de flujo del proyecto.

4.3.5 Implementación

La implementación permitió distinguir aquellos puntos que aun debían de trabajarse hasta lograr una versión aceptable y correctamente funcional del proyecto.

4.3.6 Despliegue

Es una etapa ramificada a la implementación, este punto nos permitió ver el comportamiento del proyecto, por ejemplo, los resultados que se obtenían al introducir texto con temas de historia, ciencias y demás.

4.3.7 Pruebas o verificación

A través de las pruebas y distintas ejecuciones con parámetros diferentes, se logró llegar a resultados esperados y aceptables.

4.3.8 Evaluación

En esta etapa o iteración final se decide si se cuenta con una versión estable del proyecto, de lo contrario se continúa haciendo iteraciones desde el comienzo de este tipo de metodología de desarrollo de software, donde se parte desde la planificación, requerimientos, etc. Esto para cumplir con las expectativas establecidas en la inicialización o planificación preliminar.

4.4 API de PLN Google

Se optó por seleccionar una API de Google para el proyecto, la cual hace uso del manejo del PLN, mediante esta misma se logró tener un mejor control entre las variaciones semánticas que existen dentro de un texto.

A la fecha la API de Google brinda las herramientas para el desarrollo del software pensado. Dichas herramientas permiten una adecuada manipulación en el lenguaje de programación Java.

Tal API realiza un trabajo semejante respecto a las demás en el mercado, incluso como se observa en el marco teórico en la figura 1 de la página 22 de esta tesis, la API de Google cuenta con ciertas ventajas sobre las demás APIs.

Google Cloud Platform es una herramienta de gran magnitud, debido al procesamiento de datos que existen dentro de sus servidores a nivel mundial.

Este mismo alberga una gran cantidad de datos e información que puede ser accesada mediante su API de PLN, y es por ello que este proyecto emplea tal herramienta.

Nuestro enfoque es en la implementación de la API de Cloud Natural Language de Google, porque permite que el párrafo extraído, se convierta en objetos de palabras, incluidos los descriptores o atributos de su parte de la oración, conocido como PartOfSpeech (POS). Luego, las entidades se obtuvieron utilizando la capacidad de extracción de entidad de Google NLP. Esto permite identificar las partes más importantes de la oración (salience), información sobre su tipo de palabra (persona, lugar, etc.) e incluso tener información sobre su página directa de Wikipedia en donde se puede obtener información de tal palabra extraída de la oración.

Esta tesis hace uso de la API de PLN que ofrece Google para tokenizar y realizar un análisis del texto, tanto en sintaxis, entidades y tokens, esto con el propósito de identificar cada palabra para realizar una pregunta sólida.

4.5 Generación de banco de respuestas distractoras

El siguiente tema trata la manera propuesta en la que se extrajo las respuestas distractoras a la respuesta correcta de la pregunta.

4.5.1 Diagrama de clase

A continuación, se explica las clases del diagrama de clases, así como los métodos o funciones utilizadas dentro de cada una de ellas, empleadas para la extracción de las respuestas distractoras; tal como muestra la figura 15.

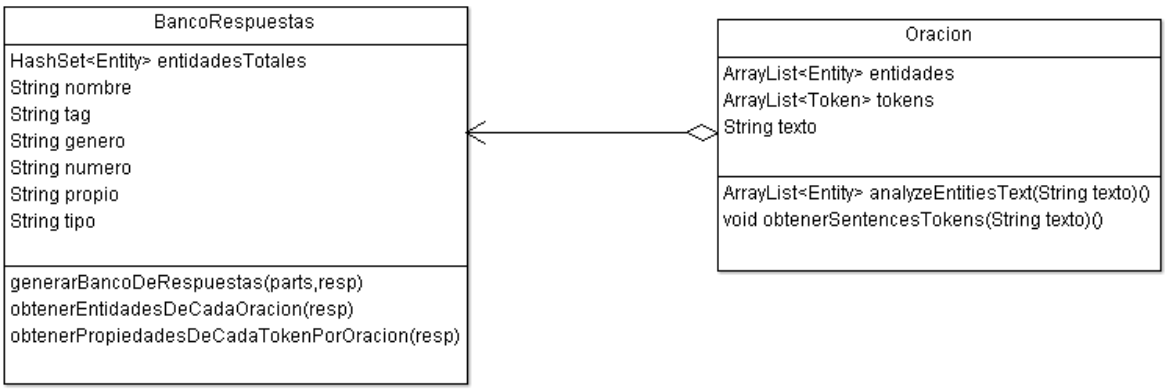

Figura 15. Diagrama de clase.

4.5.1.1 Oración.java

Es la clase que contiene las oraciones dadas a partir de un texto las cuales contienen las entidades y los tokens que se obtuvieron del análisis; son almacenados en objetos.

Mediante el uso de la API de Google de PLN se obtiene a través de las funciones correspondientes para entidades y tokens, los valores de cada una de las palabras y caracteres de cada oración, para que posteriormente la clase BancoRespuestas pueda acceder a la información de cada token y entidad.

El método analyzeEntitiesText analiza el texto completo e identifica cada palabra que es una entidad, cada una de ellas es almacenada en una lista.

El método obtenerSentencesTokens obtiene de todo el texto los valores y atributos de cada palabra, letra, carácter y símbolo. De igual forma que las entidades, son almacenados en una lista.

4.5.1.2 BancoRespuestas.java

En esta clase es donde se genera el banco de respuestas, contiene un método generarBancoDeRespuestas invocado desde la aplicación principal manda a llamar a analyzeEntitiesText y obtenerSentencesTokens de la clase Oracion.

El método obtenerEntidadesDeCadaOracion, realiza iteraciones dependiendo de cuantas entidades sean encontradas, donde, se define un objeto de tipo Entity, dicho objeto almacenara cada oración con su método Oracion.getEntidades. Oracion.analyzeEntitiesText inspecciona el texto dado y almacena exactamente el nombre de la entidad (Name) y el tipo (Type) de entidad que se analiza, nombres propios como figuras públicas, lugares, bienes de consumo, obra de arte; dichos datos son útiles para conocer el tipo información de cada entidad.

Una vez guardado el total de entidades del texto analizado se eliminaron las repeticiones, por lo que después se buscó la información de tokens para completar los atributos en el banco de respuestas.

El método Oracion.obtenerPropiedadesDeCadaTokenPorOracion, obtiene de la propiedad Tag el valor NOUN (sustantivo) y se descartaron los siguientes valores VERB (verbo), PRON (pronombre), ADJ (adjetivo) y DET (determinante), ya que una entidad es un ente o un ser; de la propiedad Gender se obtiene el género, ya sea masculino o femenino; de Number si la palabra está en plural o singular; y de la propiedad Proper si se trata de un nombre propio o impropio.

Dichos criterios son necesarios para que el banco de respuestas sea utilizado para extraer respuestas incorrectas comparadas con la respuesta correcta que conteste a cada pregunta, tales palabras tienen el rol de respuestas distractoras o respuestas homologas que sirven para formar 3 opciones de respuesta a la pregunta generada, consideradas por estas propiedades como homólogas.

4.5.2 Fragmento ejemplo de banco de respuestas generado

Como se puede observar la tabla 3 cuenta con cinco respuestas, en la cuales se muestran los datos correspondientes de Name, Type, Tag, Gender, Number y Proper.

De la propiedad TAG todos los valores son NOUN, debido a que una entidad es un ente o un ser, no puede ser verbo.

Tabla 3. Valores de la entidad

Análisis entidades		Análisis tokens			
NAME	TYPE	TAG	GENDER	NUMBER	PROPER
HISTORIA	OTHER	NOUN	FEMENINE	SINGULAR	NOT_PROPER
MÉXICO	LOCATION	NOUN	GENDER_UNKNOWN	SINGULAR	PROPER
ARI	PERSON	NOUN	MASCULINE	SINGULAR	PROPER
VEGETALES	CONSUMER_GOOD	NOUN	MASCULINE	PLURAL	NOT_PROPER
LLUVIAS	EVENT	NOUN	FEMENINE	PLURAL	NOT_PROPER

4.6 Generación de Preguntas

En esta sección se propone como generar preguntas a partir del texto dado mediante el análisis previo de cada símbolo, carácter, letra y palabra que conformen una oración.

Se analiza el diagrama de clase correspondiente a el análisis del texto para la generación de preguntas, se puede observar en la figura 16.

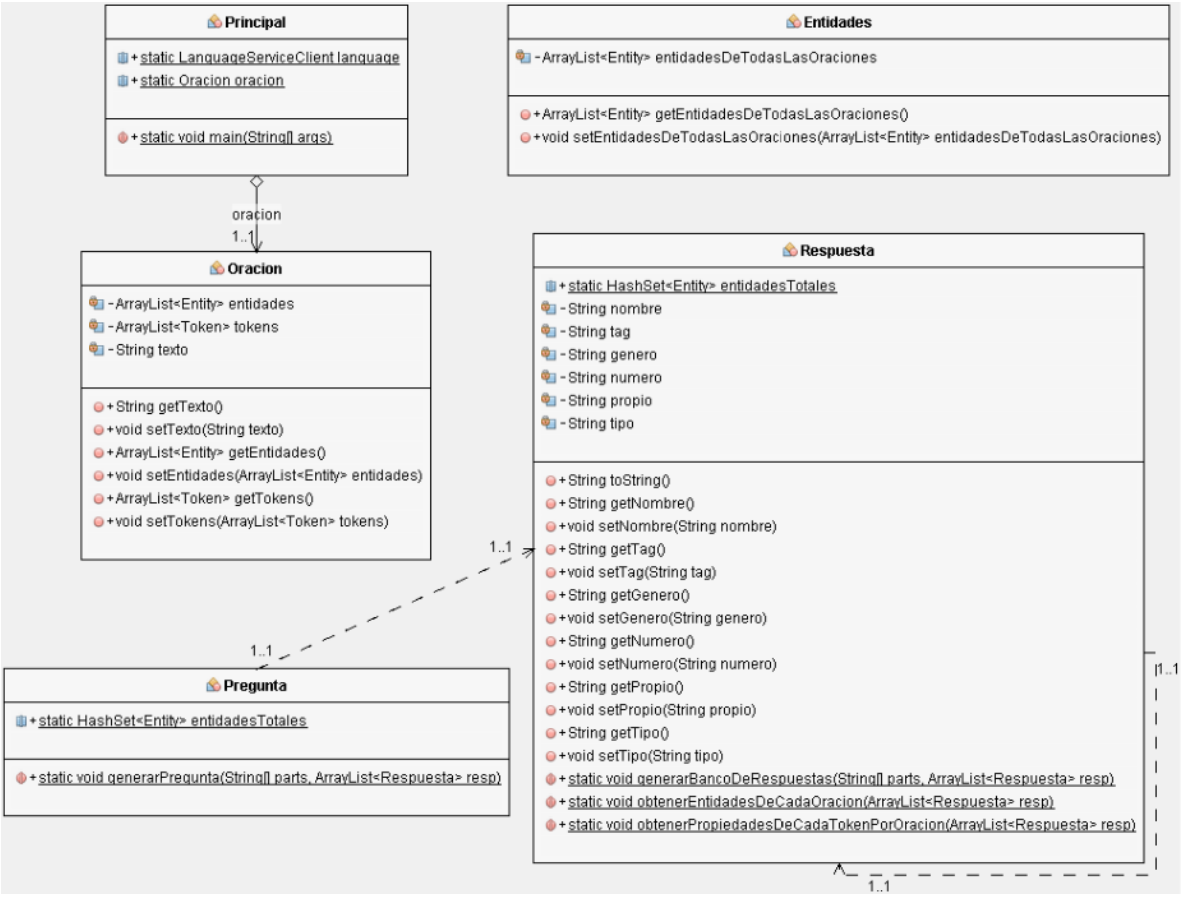

Figura 16. Diagrama de Clase Análisis de Texto.

Dentro de la clase principal se encuentra un método llamado Pregunta.generarPregunta(parts, resp) el cual pasa como parámetros parts que corresponde a la cadena del texto leído y resp que corresponde al nombre de variable que se creó para el ArrayList de tipo Respuesta.

Una vez en la clase Pregunta.java, se declara un HashSet de tipo entidades que recibe el nombre de entidadesTotales, el cual contendrá todas las entidades encontradas en el texto.

Después se encuentra la función generarPregunta que recibe el arreglo de cadena parts y el arreglo de tipo Respuesta resp, dentro se emplea un ciclo for donde el texto se analiza y se identifican o extraen los token y entidades mediante los métodos de la clase Analisis.java Analisis.obtenerSentencesTokens(part) y Analisis.analyzeEntitiesText(part) los cuales reciben como parámetro part que es el texto dado. La figura 17 muestra lo anterior descrito.

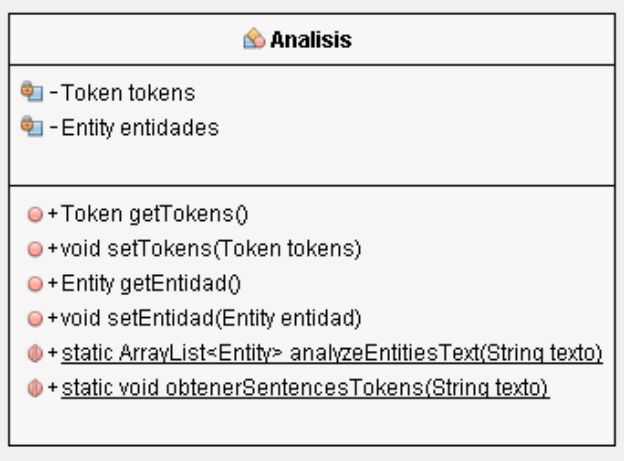

Figura 17. Diagrama de Clases Análisis.

Después de llamar los métodos, se inicializan dos HashSet de tipo cadena, uno para almacenar la propiedad tag y el otro para almacenar la propiedad label, ambas provenientes del análisis de tokens.

Recordando que el implementar HashSet evita la duplicidad de elementos guardados en una lista, ya que tanto tokens como entidades pueden tener datos repetidos en un texto dado.

Dichas propiedades, como ya se mencionó, permiten conocer del token analizado si se trata de un verbo, pronombre, sustantivo y demás valores de la propiedad TAG, y si se trata de

una preposición, un objeto directo, objeto de una preposición, entre demás valores de la propiedad LABEL.

Dentro del ciclo for actual, se anida un nuevo ciclo for, esto con la finalidad de guardar, de cada oración los tokens con sus respectivos valores de las propiedades tag y label.

Justamente después de cerrar el último ciclo for anidado, se crean validaciones, en las cuales se evalúa si el contenido de la oración es válido para formular una pregunta de tipo laguna. Como propuesta se establece que la oración debe contener las propiedades de sustantivo (NOUN), pronombre (PRON), verbo (VERB), preposiciones (PREP) con el propósito de conocer la vinculación entre palabras, objeto directo (DOBJ) el cual es la persona o cosa afectada directamente por el verbo y objeto de una preposición (POBJ) el cual se refiere un sustantivo o un pronombre que completa su significado, por ejemplo Benito dio sus votos a la nación.

Entonces si la oración cumple con los valores propuestos, se establece el valor de una bandera en valor verdadero, dicha bandera se inicializa antes de las validaciones como falsa, después de esto se guarda en el objeto oración.

4.7 Obtención de respuestas correctas

Para la generación e identificación de la respuesta correcta a la pregunta, se realiza un primer análisis del texto, y mediante la API de Google de PLN, se identifican todas las entidades del texto, donde la entidad de mayor relevancia (SALIENCE) sirve como respuesta correcta o palabra que responde correctamente a la pregunta.

Después de la validación del texto para extraer la pregunta, se procede a inicializar las variables que son de ayuda para manipular los datos y la información percibida del análisis de texto para la obtención de las respuestas correctas.

Se inicializa las variables de tipo cadena, respuestaTipo, para los tipos de entidades, respuesta para la palabra que conteste correctamente y respuestaNumber para saber si la entidad es plural o singular, algo que es muy importante al momento de empatar las respuestas distractoras a la correcta; ya que se debe contemplar que, si la respuesta correcta está en singular, las respuestas distractoras también deberán estarlo.

Se crea e inicializa una variable que permite albergar en un solo arreglo la respuesta y el tipo de respuesta proveniente del Banco de Respuestas por Oración (BancoRespuestasOracion.obtenerBancoDeRespuestasPorOracion); dicha función obtiene el peso de la entidad (SALIENCE) y su tipo (TYPE) excluyendo los tipos de OTHER, UNKNOWN y EVENT.

Para proponer buenas palabras que funjan como respuestas, es necesario tomar las propiedades de tokens y entidades, pero cada análisis devuelve distintas propiedades y valores, ya que entidades solo retorna peso y tipo, se debe complementar con los valores que provienen del análisis tokens, los cuales son number, tag y proper.

Se propone una iteración de todos los tokens y que cada uno de ellos se compare con la entidad respuesta de la oración; si el token resulta ser igual a la entidad (dicha entidad ya cuenta con los valores de peso y tipo) se le asignara los valores correspondientes al análisis de tokens (number, tag y proper).

4.8 Obtención de respuestas incorrectas homólogas a la correcta

Por cada pregunta generada existe solamente una respuesta correcta, donde dicha respuesta es comparada con el banco de respuestas generado del análisis del texto, con la finalidad de comparar los valores provenientes del análisis de tokens y entidades, con el propósito de seleccionar aquellas respuestas distractoras a la correcta.

Una vez realizado la obtención de respuestas correctas del subtema anterior (4.7 Obtención de respuestas correctas). La respuesta con los valores de entidades y tokens se guarda en un arreglo llamado tokenRespuesta, donde se encuentran el nombre de todas las entidades que sean semejantes a la respuesta correcta.

Después se seleccionan tres respuestas distractoras de ese arreglo tokenRespuesta para empatar a la respuesta correcta, logrando tener así cuatro opciones de respuestas donde solo una es la correcta. Como tales respuestas distractoras son semejantemente iguales en valores a la respuesta correcta, son seleccionadas por posición del arreglo y este posicionamiento surge de ordenar de mayor a menor, a través del valor dado por el peso de la entidad (SALIENCE).

4.9 Estructura de la oración

La oración es un enunciado que contiene una forma verbal. En todas las oraciones hay dos partes: el sintagma nominal sujeto y el sintagma verbal predicado.

- El sujeto: de una oración se refiere a la persona, animal, cosa, idea, sentimiento, de quien se está haciendo la oración.
- El sujeto nominal tiene por núcleo o por palabra más importante, el sustantivo.
- El predicado es lo que se menciona del sujeto dentro de una oración.

Una de las herramientas CASE para el proyecto de tesis a utilizar es Argo UML, pero solo es un auxiliar para generar un aspecto grafico del actual problema a solucionar, se puede observar con detalle en la figura 18.

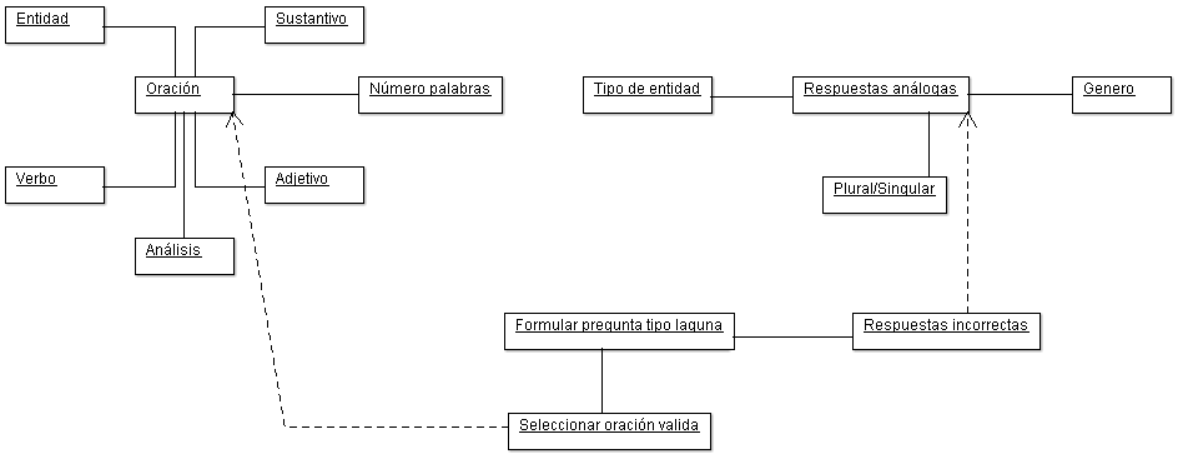

Figura 18. Secuencia del proyecto.

Para generar una buena pregunta, hay que plantear el contenido de esta, si contiene algunas características y propiedades que son importantes para que se logre el objetivo y propósito; sin la formulación de una buena pregunta, se puede dar lugar a quien conteste dicha serie de preguntas o cuestionarios se confunda y no se logre concretar el cometido.

4.10 Análisis de entidades

Devuelve un conjunto de entidades detectadas y parámetros asociados con esas entidades, como el tipo de entidad, la relevancia de la entidad para el texto general y las ubicaciones en el texto que se refieren a la misma entidad. Las entidades se devuelven en el orden (de mayor a menor) de sus puntajes de salience, lo que refleja su relevancia para el texto general.

El análisis de la entidad es útil para la desambiguación de entidades similares como es en el caso de la palabra Lawrence, dado el ejemplo con Lawrence of Arabia la cual es una película y Thomas Edward Lawrence que es una persona.

Este análisis inspecciona el texto dado para las entidades conocidas (nombres propios como figuras públicas o personas famosas, puntos de referencia, etc.) y proporciona información sobre esas entidades.

Los campos utilizados para almacenar los parámetros de la entidad se listan a continuación:

- Type: Indica el tipo de esta entidad (por ejemplo, si la entidad es una persona, ubicación, bien de consumo, etc.) Esta información ayuda a distinguir o desambiguar entidades, y se puede usar para escribir patrones o extraer información.
- PartOfSpeech: Dentro de este campo de la respuesta se devuelve una solicitud sintáctica, la información morfológica y su categoría gramatical. Este campo contiene un conjunto de subcampos con información de dicha categoría (POS) así como información morfológica más explícita.
- Metadata: Contiene información fuente sobre el repositorio de conocimiento de la entidad. Repositorios adicionales pueden estar expuestos en el futuro.
- Salience: Indica la importancia o relevancia de esta entidad para todo el texto del documento. Este puntaje puede ayudar a recuperar y resumir información al priorizar entidades destacadas.

Para que se logre una buena calidad en la generación de preguntas y respuestas, se debe contar con un banco de estas mismas que vaya proporcionando un estándar de calidad para una correcta formulación del cuestionario con sus respectivas respuestas correctas y homologas distractoras.

Tal como se menciona en Automation of question generation from sentences (Ali, 2010), dependiendo de la oración, en su naturaleza, puede presentar alguna dificultad de interpretación o de contexto, como por ejemplo los elementos que componen la oración o su propia longitud; se pretende generar preguntas elementales a partir de oraciones que presenten o no dicha adversidad.

Para resolver la situación de una buena calidad en la generación de preguntas y respuestas, se utiliza el análisis sintáctico, el análisis de entidades, y un etiquetador POS, de lo cual se extrae el sujeto (ya sea sustantivo, pronombre o verbo en infinitivo), verbo, objeto y preposición, para determinar si la oración es viable para generar una pregunta de calidad.

Para llevar una secuencia correcta se realiza el análisis sintáctico, se extrae los tokens, se emplea la propiedad POS de la cual la etiqueta tag indica si tiene sustantivo y verbo.

Dentro del análisis de tokens, se utiliza la etiqueta label, la cual mediante el valor PREP se extrae la preposición.

También de dicha clase token se obtiene el objeto directo a través de la etiqueta label con el valor DOBJ y para el objeto de la preposición el valor de POBJ.

4.11 Etiquetado PoS

Por sus siglas en ingles Part of Speech que en español es Categoría Gramatical, lee el texto y etiqueta o asigna categorías a cada palabra, verbo, adjetivo, etc. Esto en función del contexto de la oración.

Dentro de una solicitud sintáctica, la información morfológica y categoría gramatical se devuelve dentro del campo partOfSpeech de la respuesta. El campo partOfSpeech contiene un conjunto de subcampos con información de la categoría (POS) así como información morfológica más explícita.

Dentro de la morfología que existe en el análisis del texto que brinda partOfSpeech, existen campos que reflejan información como los siguientes:

Tag. - Denota las partes de la oración, usando una etiqueta POS como (SUSTANTIVO, VERBO, etc.) y proporciona información de sintaxis del nivel superior. Las etiquetas POS son útiles para crear patrones y reducir la ambigüedad para el posterior análisis del lenguaje (por ejemplo, "tren" etiquetado como un SUSTANTIVO frente a un VERBO).

Number. - Denota el número gramatical de una palabra que indica su distinción de recuento. En inglés, el sufijo "s" se usa generalmente para distinguir formas plurales de sustantivos de singular, por ejemplo. Algunos idiomas, como el árabe, tienen la noción de un número dual también. Este campo puede contener los siguientes valores:

- SINGULAR denota una cantidad.
- PLURAL denota más de una cantidad.
- DUAL denota precisamente dos cantidades.

Tanto el artículo en la Revista Cubana de Ciencias Informáticas (Rodríguez Blanco & Simón Cuevas, 2013) como en la conferencia de Automation of question generation from sentences (Ali, 2010), realizan el análisis del texto, mediante el cual comparten la extracción del sujeto, sustantivo, pronombre, verbo, objeto directo, objeto de preposición, se tomara tal manera de extraer dichos elementos para generar una pregunta sólida en este proyecto.

Como menciona Daniel Alfaro (Paitamala, 2016) precisa de herramientas como Ixa-pipe-otk e Ixa-pipe-parse para analizar el texto sintácticamente y extraer los tokens para su correcta separación del texto en fragmentos.

Empleando los criterios de number (plural/singular), gender (femenino, masculino o neutro), tipo de entidad, entre otros subcampos que retorna partOfSpeech perteneciente a la clase de entidades dentro de la API.

CAPÍTULO V. RESULTADOS Y DISCUSIÓN

En este capítulo se trata los resultados obtenidos, así como si existe lugar a un tema de discusión.

5.1 Resultados

Una vez completado todo el análisis ya descrito en esta tesis, se generan las preguntas validas con sus respectivas respuestas distractoras y su respuesta que conteste correctamente.

El proceso de generación de reactivos (preguntas y respuestas) se realiza de una manera rápida que minimiza en más de un 80% el tiempo invertido en la formulación de preguntas y respuestas que conformen un examen de algún tema deseado. Esto significa la creación de cuestionarios en minutos, para ejemplificar tiempos, de un total de 54 oraciones, se invierte alrededor de 3 minutos en la generación de preguntas y respuestas.

El resultado final de las preguntas generadas es afectado directamente por los criterios (sustantivo, pronombre, verbo, preposiciones, objeto directo y objeto de la preposición) que se toman en cuenta para decidir si la oración de donde se extrae la pregunta cuenta con información valida que permita dicha formulación.

A continuación, se muestra el resultado de procesar el texto dado para la generación de preguntas con sus respectivas respuestas distractoras y correctas.

5.2 Obtención del banco (universo) de respuestas del texto dado

En la figura 19 de la siguiente página, se muestra un total de 54 oraciones encontradas en el Bloque 1 del libro Historia de cuarto grado para primaria de la SEP.

```
Número de oraciones: 54

TAMAÑO: 357
RESPUESTAS: [Historia, NOUN, FEMININE, SINGULAR, NOT_PROPER, OTHER
, grado, NOUN, MASCULINE, SINGULAR, NOT_PROPER, OTHER
, bloque, NOUN, MASCULINE, SINGULAR, NOT_PROPER, OTHER
, Panorama del periodo, null, null, null, null, WORK_OF_ART
, Ubicación, NOUN, FEMININE, SINGULAR, NOT_PROPER, OTHER
, Aridoamérica, NOUN, GENDER_UNKNOWN, SINGULAR, PROPER, LOCATION
, Oasisamérica, NOUN, FEMININE, SINGULAR, PROPER, LOCATION
, Mesoamérica, NOUN, GENDER_UNKNOWN, SINGULAR, PROPER, LOCATION
, investigadores, NOUN, MASCULINE, PLURAL, NOT_PROPER, PERSON
, áreas, NOUN, FEMININE, PLURAL, NOT_PROPER, OTHER
, Mesoamérica, null, null, null, null, LOCATION
, México, NOUN, GENDER_UNKNOWN, SINGULAR, PROPER, LOCATION
, Aridoamérica, null, null, null, null, CONSUMER_GOOD
, Oasisamérica, null, null, null, null, LOCATION
, características, NOUN, FEMININE, PLURAL, NOT_PROPER, OTHER
```

Figura 19. Listado de entidades respuesta con sus valores.

5.3 Oración válida

Oración analizada y catalogada como válida: En algunas regiones del actual territorio mexicano, como Aguascalientes, Morelos, Guerrero y principalmente Baja California, varios grupos nómadas hicieron pinturas rupestres, llamadas así porque fueron pintadas sobre superficies rocosas.

5.4 Respuesta correcta a la pregunta

De toda la oración se extrae y se guarda la entidad con mayor peso (valor SALIENCE) y de tipo (TYPE) diferente a OTHER, se puede observar los resultados en la figura 20.

```
LO QUE GUARDO: regiones
INFO ENTIDAD: regiones PESO: 0.18356754 TIPO: LOCATION
INFO ENTIDAD: territorio PESO: 0.14621614 TIPO: LOCATION
INFO ENTIDAD: Guerrero PESO: 0.109405205 TIPO: LOCATION
INFO ENTIDAD: Baja California PESO: 0.109405205 TIPO: LOCATION
INFO ENTIDAD: mexicano PESO: 0.103192754 TIPO: LOCATION
INFO ENTIDAD: superficies PESO: 0.08595099 TIPO: LOCATION
INFO ENTIDAD: grupos PESO: 0.07698167 TIPO: PERSON
INFO ENTIDAD: pinturas PESO: 0.07698167 TIPO: WORK_OF_ART
INFO ENTIDAD: Aguascalientes PESO: 0.054149408 TIPO: LOCATION
INFO ENTIDAD: Morelos PESO: 0.054149408 TIPO: LOCATION
```

Figura 20. Entidades con mayor peso.

5.5 Respuestas incorrectas homologas a la correcta por pregunta

La sinergia de SALIENCE y TYPE permite ubicar de todo el universo de entidades previamente identificadas en el texto, el conjunto de entidades que formaran parte de las respuestas distractoras a la respuesta correcta.

En la figura 21, 22 y 23 se muestra el resultado de comparar todas las entidades del universo con la respuesta correcta, donde se agregan a la lista de respuestas homogéneas a la respuesta correcta (tokenRespuesta) y se deben seleccionar tres de ellas para la función de distractoras.

```
AGREGADO A LA LISTA DE RESPUESTAS: tierras
AGREGADO A LA LISTA DE RESPUESTAS: ciudades
AGREGADO A LA LISTA DE RESPUESTAS: viviendas
AGREGADO A LA LISTA DE RESPUESTAS: zonas
AGREGADO A LA LISTA DE RESPUESTAS: montañas
AGREGADO A LA LISTA DE RESPUESTAS: mesetas
AGREGADO A LA LISTA DE RESPUESTAS: superficies
```

Figura 21. Lista de respuestas homogéneas.

Más resultados de ejemplo

```
INFO ENTIDAD: minerales PESO: 0.44809586 TIPO: OTHER
INFO ENTIDAD: colores PESO: 0.20126674 TIPO: OTHER
INFO ENTIDAD: árboles PESO: 0.09539096 TIPO: OTHER
INFO ENTIDAD: agua PESO: 0.09539096 TIPO: OTHER
INFO ENTIDAD: grasa PESO: 0.08553268 TIPO: OTHER
INFO ENTIDAD: resina PESO: 0.074322805 TIPO: OTHER
```

Figura 22. Lista de respuestas homogéneas 1.

```
INFO ENTIDAD: información PESO: 0.3350836 TIPO: OTHER
LO QUE GUARDO: integrantes
INFO ENTIDAD: integrantes PESO: 0.24567455 TIPO: PERSON
INFO ENTIDAD: restos PESO: 0.13670692 TIPO: OTHER
INFO ENTIDAD: plantas PESO: 0.06233982 TIPO: OTHER
INFO ENTIDAD: frutos PESO: 0.06233982 TIPO: OTHER
INFO ENTIDAD: época PESO: 0.056856595 TIPO: OTHER
INFO ENTIDAD: animales PESO: 0.055700906 TIPO: OTHER
INFO ENTIDAD: pobladores PESO: 0.04529777 TIPO: PERSON
```

Figura 23. Lista de respuestas homogéneas 2.

5.6 Reemplazo de la respuesta correcta por la laguna

Se toma la oración válida para formular una pregunta, se reimprime y a continuación se reemplaza la entidad más importante respecto su nivel SALIENCE con una laguna o espacio en blanco.

En la figura 24 se muestra un ejemplo al momento de realizar tal reemplazo.

```
Entidad analizada para quitar de oración: regiones


PREGUNTA DE LUGAR

 En algunas __________ del actual territorio mexicano, como Aguascalientes, Morelos,
 Guerrero y principalmente Baja California, varios grupos nómadas hicieron pinturas
rupestres, llamadas así porque fueron pintadas sobre superficies rocosas.
```

Figura 24. Pregunta.

5.7 Preguntas generadas con sus respectivas respuestas

En la figura 25 se muestran las preguntas generadas con su respuesta que contesta correctamente y las distractoras homogéneas.

```
 En algunas __________ del actual territorio mexicano, como Aguascalientes, Morelos,
 Guerrero y principalmente Baja California, varios grupos nómadas hicieron pinturas
rupestres, llamadas así porque fueron pintadas sobre superficies rocosas.

Seleccione la respuesta correcta:
1.- zonas
2.- viviendas
3.- montañas
4.- regiones
```

Figura 25. Reactivo.

5.8 Base de datos MySQL

Para el almacenamiento de las preguntas con sus respectivas respuestas correctas e incorrectas, se propone emplear el motor de base de datos que ofrece MySQL con una relación uno a muchos entre la tabla que contendrá las preguntas y la tabla que contendrá las respuestas (correctas e incorrectas).

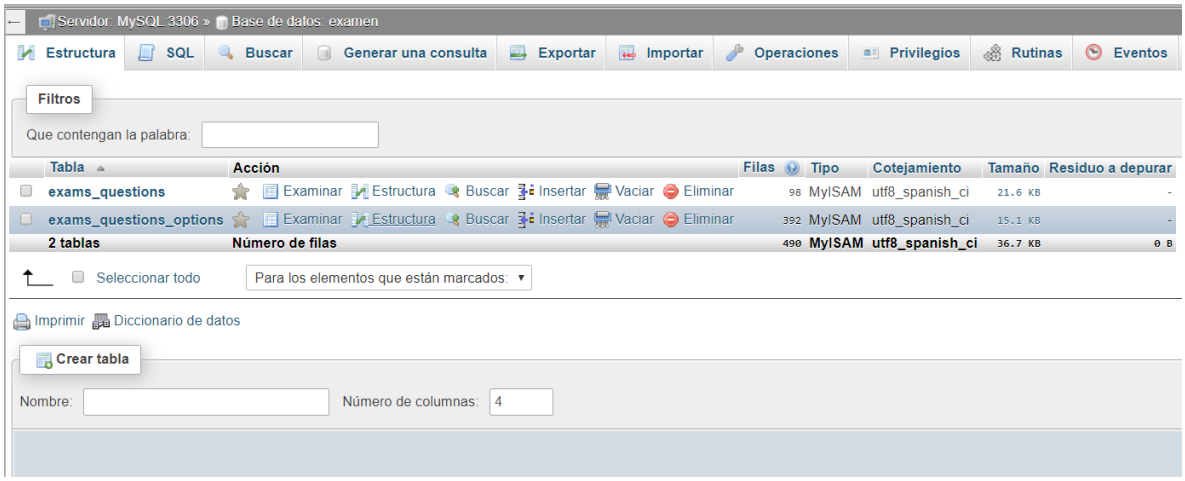

Figura 26. Tablas BD.

En la figura 27 se muestra la tabla “exams_questions” la cual tiene se encarga del almacenamiento de las preguntas, en las cuales se sustituye la respuesta correcta por una laguna o un espacio en blanco. Como se puede observar cada pregunta está identificada con un “id” único.

Mostrando filas 0 - 10 (total de 11, La consulta tardó 0,0000 segundos.)

```
SELECT * FROM `exams_questions`
```

Perfilando [E

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

+ Opciones

	id	exam_id	section_id	text	hint
Editar Copiar Borrar	1	2	3	Estas _____ poseían características geográfic...	áreas
Editar Copiar Borrar	2	2	3	Aridoamérica abarcó parte del _____ actual de...	territorio
Editar Copiar Borrar	3	2	3	_____ cubría gran parte de la superficie de l...	oasisamérica
Editar Copiar Borrar	4	2	3	Esta _____ posee un territorio semiárido, de ...	región
Editar Copiar Borrar	5	2	3	Esta área tiene varios tipos de clima y cuenta con...	zona
Editar Copiar Borrar	6	2	3	Hace miles de años, la _____ experimentó una ...	tierra
Editar Copiar Borrar	7	2	3	Las glaciaciones provocaron cambios en las plantas...	mares
Editar Copiar Borrar	8	2	3	Con ello se formó un _____, el cual permitió ...	puente
Editar Copiar Borrar	9	2	3	Se cree que los primeros pobladores de _____ ...	américa
Editar Copiar Borrar	10	2	3	Una de las razones fue que los grupos humanos eran...	zona
Editar Copiar Borrar	11	2	3	La información que aportan estos restos materiales...	integrantes

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Figura 27. Tabla "exams_questions".

En la figura 28 de la tabla "exams_questions_options" se muestran las respuestas que corresponden a cada pregunta. Cuatro respuestas corresponden a una pregunta, son identificadas por el campo “question_id”, cada respuesta es identificada por un identificador único, además para resaltar la respuesta correcta en el campo “is_correct” colocamos un 1. Las respuestas que tienen 0, corresponden a las respuestas distractoras, que son homologas en sus propiedades.

+ Opciones

	id	question_id	text	is_correct
Editar Copiar Borrar	1	1	zonas	0
Editar Copiar Borrar	2	1	gracias	0
Editar Copiar Borrar	3	1	áreas	1
Editar Copiar Borrar	4	1	cuerdas	0
Editar Copiar Borrar	5	2	sur	0
Editar Copiar Borrar	6	2	desierto	0
Editar Copiar Borrar	7	2	puente	0
Editar Copiar Borrar	8	2	territorio	1
Editar Copiar Borrar	9	3	oasisamérica	1
Editar Copiar Borrar	10	3	región	0
Editar Copiar Borrar	11	3	península	0
Editar Copiar Borrar	12	3	región	0
Editar Copiar Borrar	13	4	región	1
Editar Copiar Borrar	14	4	oasisamérica	0
Editar Copiar Borrar	15	4	península	0
Editar Copiar Borrar	16	4	oasisamérica	0
Editar Copiar Borrar	17	5	península	0
Editar Copiar Borrar	18	5	zona	1
Editar Copiar Borrar	19	5	oasisamérica	0
Editar Copiar Borrar	20	5	región	0
Editar Copiar Borrar	21	6	región	0
Editar Copiar Borrar	22	6	oasisamérica	0
Editar Copiar Borrar	23	6	península	0
Editar Copiar Borrar	24	6	tierra	1
Editar Copiar Borrar	25	7	mares	1

Figura 28. Tabla "exams_questions_options".

CAPÍTULO VI. CONCLUSIONES

Para finalizar este trabajo, se crea este apartado final, en el que se discute los resultados obtenidos, el aprendizaje obtenido durante el tiempo transcurrido de este cometido, la experiencia adquirida a través de cada situación presentada en el desarrollo de este, y dar a conocer el posible trabajo futuro.

En conclusión, se observa una mejoría en la creación de exámenes para PEC, generando preguntas de una manera más rápida, además estas mismas muestran una estructura sólida para el alumno, y que la respuesta a la misma no se realice de forma obvia o por sentido común.

Tal como se propuso, existe una automatización en la creación de los reactivos que conformaran los cuestionarios, de esta manera se excluye la manipulación externa de una persona para decidir el contenido de tales reactivos (que preguntas y respuestas se incluyen en los mismos).

Como se planeó, el extraer preguntas y respuestas de un texto de un tema en específico, es posible gracias al PLN y al análisis sintáctico y morfológico que brinda el API de Google, y se determina si el texto de un párrafo es válido para extraer preguntas.

Mediante la comparación de las propiedades en las entidades del texto se logra una relación homogénea de las palabras que tendrán el papel de respuestas correctas e incorrectas, de esta forma se pueden relacionar entidades que sean iguales en un sentido plural y singular, iguales en un sentido masculino y femenino (por ejemplo, carro y autobús, para masculino; y playa, costa, para femenino), un mismo tipo de entidad o por decir de otra manera, un mismo tipo de categoría, por ejemplo el hecho de no mezclar contextos diferentes, como por ejemplo, lápiz y carro, o celular con mesa, ya que carro corresponde a un medio de transporte y lápiz a un instrumento de escritura; mediante tipo de entidad se agrupan las categorías semejantes, como por ejemplo, costa con playa y mares con ríos.

El alcance del análisis es basado exclusivamente en forma textual, no se analiza alguna fórmula matemática o alguna ecuación, tampoco se analizan imágenes que pudieran contener alguna temática con texto.

La conectividad a internet es necesaria para la conexión a la API de PLN ofrecida por Google, la cual permite realizar el debido análisis para el texto, del cual se extraen las propiedades de cada palabra, cada carácter y sus debidos análisis sintácticos y morfológicos.

El presente trabajo queda abierto a la sugerencia de mejoras futuras para complementar y enriquecer aún más lo realizado, ya sea en una nueva tesis y proyecto que tome como base este trabajo o sea de referencia para nuevas aplicaciones de lo aquí realizado e investigado.

Me quedo con la satisfacción de haber conocido un área de la tecnología que a la fecha comienza a ser desarrollada en muchas partes del mundo y con un sinfín de aplicaciones, pude adentrarme a conocer más respecto a la utilidad que se le puede dar a las herramientas de software y ver que la gente puede resultar beneficiada de ello.

El presentar este trabajo frente a personas conocedoras del tema, profesionistas y público en general en el 10º Congreso Internacional de Investigación Científica Multidisciplinaria en las instalaciones del Instituto Tecnológico de Monterrey, fue una experiencia realmente retadora, algo que me llevo a trabajar en la correcta explicación de los temas desarrollados en este trabajo, ya que justamente el reto está en cómo transmitirle a la gente de una manera simple y entendible lo que con ayuda del comité de tesis y el trabajo personal se logró.

La participación en dicho congreso y en cada coloquio que se organizó en el Tecnológico sede durante cada semestre, me permitió relacionarme con gente de la cual aprendí demasiado, que dejo en mi un pensamiento reflexivo respecto a los avances tecnológicos y científicos que en la actualidad se viven.

CAPÍTULO VII. BIBLIOGRAFÍA

Alghini, C. (Febrero de 2018). *Lectura entre líneas: análisis de texto con Google Cloud Natural Language Processing*. Obtenido de http://www.coolheadtech.com/blog/text-analysis-with-google-cloud-natural-language-processing

Ali, H. C. (June de 2010). Automation of question generation from sentences. *In Proceedings of QG2010: The Third Workshop on Question Generation*, (págs. 58-67).

Amazon. (Marzo de 2020). *Amazon Web Services*. Obtenido de https://aws.amazon.com/es/comprehend/

Apache Maven. (Marzo de 2020). *Maven*. Obtenido de https://maven.apache.org/

Apache NetBeans. (Marzo de 2020). *NetBeans*. Obtenido de https://netbeans.org/

Apache OpenNLP. (2017). *OpenNLP*. Obtenido de https://opennlp.apache.org/

Aylien. (Marzo de 2020). *Aylien*. Obtenido de https://aylien.com/

Butler, A., & Roediger, H. (2008). Feedback enhances the positive effects and reduces the negative effects of multiple-choice testing. *Memory & Cognition*, 36, 604-616.

Cortez Vásquez, A., Vega Huerta, H., Quispe Pariona, J., & Huayna, A. (2014). Procesamiento de lenguaje natural. *Revista de investigación de Sistemas e Informática*, 6(2), 45 - 54. Obtenido de Cortez Vásquez, A., Vega huerta, H., Pariona Quispe, J., & Huayna, A. (2014). Procesamiento de lenguaje natural. Revista de investigación de Sistemas e Informática, 6(2), 45 - 54. Recuperado de http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/ar

DEVPOST. (Marzo de 2020). Obtenido de Quiz It: https://devpost.com/software/quiz-it

Dinesh, S. (Agosto de 2016). *Medium*. Obtenido de https://medium.com/google-cloud/new-google-cloud-ssds-have-amazing-price-to-performance-2a58e7d9b433

Discover, C. (2018). *Expert System*. Obtenido de https://www.expertsystem.com/es/productos/cogito-discover-mineria-de-textos/

Drupal. (Marzo de 2020). *FreeLing*. Obtenido de http://nlp.lsi.upc.edu/freeling/node/1

Expert System. (Marzo de 2020). *Cogito-studio*. Obtenido de https://www.expertsystem.com/products/cogito-studio/

Ferrández, O., Kozareva, Z., Montoyo, A., & Muñoz, R. (2005). NERUA: sistema de detección y clasificación de entidades utilizando aprendizaje automático. *Redalyc*, 35(37-44), 94.

Ferreyra, M. F., & Backhoff-Escudero, E. (2016). Validez del Generador Automático de Ítems del Examen de Competencias Básicas (Excoba). *RELIEVE-Revista Electrónica de Investigación y Evaluación Educativa*, 22(1).

Gierl, M. J. (2012). Using automatic item generation to create multiple-choice test items. *Medical Education*, 46: 757-765.

Gierl, M. J. (2013). *Advances in Automatic Item Generation with Demonstration. Centre for Research in Applied Measurement and Evaluation.* Edmonton: University of Alberta.

Gierl, M. J. (2018). Using Automatic Item Generation to Create Solutions and Rationales for Computerized Formative Testing. *Applied psychological measurement*, 42-57.

Habed López, N. (Octubre de 2012). Hacia una pedagogía de la pregunta. *El Nuevo Diario*.

IBM. (Enero de 2020). *IBM Watson Studio*. Obtenido de https://dataplatform.cloud.ibm.com/docs/content/DO/WML_Deployment/DeployModelRest.html

Jackson, B. (Diciembre de 2019). *Las 7 Ventajas Principales de Escoger Google Cloud Hosting*. Obtenido de https://kinsta.com/es/blog/google-cloud-hosting/

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (Junio de 2014). The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55-60.

Milindjagre. (Agosto de 2016). *milindjagre.co*. Obtenido de https://milindjagre.co/2016/08/26/twitter-sentiment-analysis-using-opennlp-java-api/

Montenegro, J. V. (2010). Analysis of a gold standard for concept map mining – How humans summarize text using concept maps. Viña del Mar, Chile.

Paitamala, D. A. (2016). *Generación de preguntas sobre un texto.* Lejona.

Pandita, R., Xiao, X., Zhong, H., Xie, T., Oney, S., & Paradkar, A. (2012). Inferring method specifications from natural language API descriptions. *In Software Engineering (ICSE), 2012 34th International Conference* (págs. 815-825). Zúrich: IEEE Press.

Rivas, L. A. (Enero de 2016). *mugsnoticias*. Obtenido de http://www.mugsnoticias.com.mx/noticias-del-dia/a-quien-debemos-dar-gracias-por-paco-el-chato-y-el-libro-espanol-lecturas-de-primer-grado/

Robert Dale, K.-F. W. (2005). *Natural Lenguage Processing - IJCNLP 2005.* Korea: Springer.

Rodríguez Blanco, A., & Simón Cuevas, A. (2013). Method to extract structured information from texts. *Revista Cubana de Ciencias Informáticas*, 55-67.

Roggeman, S. (Febrero de 2017). *ICAPPS*. Obtenido de https://www.icapps.com/blog/linguistics-behind-chatbots

S. Bloom, B. (1956). *Taxonomy of Educational Objectives.* Chicago: Longmans.

Secretaría de Educación Pública. (Marzo de 2020). *CONALITEG*. Obtenido de https://libros.conaliteg.gob.mx/content/common/consulta-libros-gb/

Sommerville, I. (2005). *Ingeniería de software.* Madrid: Pearson.

TextRazor. (Noviembre de 2019). *Named Entity Recognition*. Obtenido de https://www.textrazor.com/named_entity_recognition

Thakur, T. (s.f.). *Medium*. Obtenido de https://medium.com/kontikilabs/comparing-machine-learning-ml-services-from-various-cloud-ml-service-providers-63c8a2626cb6

Tomlin, R. S. (2014). *Basic Word Order (RLE Linguistics B: Grammar): Functional Principles.* London: Routledge.

Vásquez, A., Quispe, J., & Huayna, A. (2009). Procesamiento de lenguaje natural. *Revista de investigación de Sistemas e Informática*, 6(2), 45-54.

Winston, W. W. (1987). Managing the development of large software systems: concepts and techniques. *Proceedings of the 9th international conference on Software Engineering*, 328-338.

ANEXOS

Se muestran términos que son empleados en la tesis, los cuales son de utilidad para la mejor comprensión del documento.

Glosario

Ambigüedad

Hay muchas palabras que tienen más de un significado (polisemia) y también hay oraciones que se prestan a más de una interpretación. Cuando por el contexto no podemos determinar el significado de una palabra o de una oración, decimos que los signos se están usando con ambigüedad.

Análisis de sentimiento de la entidad

Combina el análisis de la entidad y el análisis del sentimiento e intenta determinar el sentimiento (positivo o negativo) expresado sobre las entidades dentro del texto. El sentimiento de la entidad está representado por el puntaje numérico y los valores de magnitud, y se determina para cada mención de una entidad. Esos puntajes luego se agregan a un puntaje de sentimiento general y magnitud para una entidad.

Análisis morfológico

Consiste en el análisis interno de las palabras que forman oraciones y así extrae lemas, rasgos flexivos. Es esencial para la información básica como lo es la categoría sintáctica y significado léxico.

Resuelve problemas mediante el análisis de sus partes que lo conforman, se emplea para la generación de ideas rápidamente.

Análisis semántico

Proporciona la interpretación de las oraciones, una vez eliminadas las ambigüedades morfosintácticas.

Análisis pragmático

Incorpora el análisis del contexto de uso a la interpretación final del texto. Dentro de este se incluye o incorpora el tratamiento del lenguaje figurado, metáfora e ironía, como el conocimiento del mundo específico necesario para entender un texto especializado.

Análisis de sentimiento

Inspecciona el texto dado e identifica la opinión emocional prevaleciente dentro del texto, especialmente para determinar la actitud del escritor como positiva, negativa o neutral.

De tal análisis se desprende dos campos, magnitude que representa la magnitud absoluta del sentimiento independientemente de la puntuación (positiva o negativa) y score que es puntaje de sentimiento entre -1.0 (sentimiento negativo) y 1.0 (sentimiento positivo).

API

Una API es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. Las siglas API vienen del inglés Application Programming Interface. En español sería Interfaz de Programación de Aplicaciones.

Una API nos permite implementar las funciones y procedimientos que engloba en nuestro proyecto sin la necesidad de programarlas de nuevo. En términos de programación, es una capa de abstracción.

API REST

Buscando una definición sencilla, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad, pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como REST.

Componentes del procesamiento del lenguaje natural

Se muestran algunos de los componentes del PLN. No todos los análisis que se describen se aplican en cualquier tarea de dicho procesamiento del lenguaje, sino que depende del objetivo de la aplicación.

- Análisis morfológico o léxico. Consiste en el análisis interno de las palabras que forman oraciones para extraer lemas, rasgos flexivos, unidades léxicas compuestas. Es esencial para la información básica: categoría sintáctica y significado léxico.
- Análisis sintáctico. Consiste en el análisis de la estructura de las oraciones de acuerdo con el modelo gramatical empleado (lógico o estadístico).
- Análisis semántico. Proporciona la interpretación de las oraciones, una vez eliminadas las ambigüedades morfosintácticas.
- Análisis pragmático. Incorpora el análisis del contexto de uso a la interpretación final. Aquí se incluye el tratamiento del lenguaje figurado (metáfora e ironía) como el conocimiento del mundo específico necesario para entender un texto especializado.

Capa de abstracción

La capa de abstracción (en inglés Hardware Abstraction Layer o HAL), es un conjunto de funcionalidades de un sistema operativo que permiten a los programadores acceder, de forma fácil y transparente a la base de datos del sistema. Es por tanto muy crítica, ya que la mayoría de aplicaciones utiliza accesos a la base de datos para mostrar la información. La HAL funciona como una interfaz entre el software y el hardware del sistema, proveyendo una plataforma de hardware consistente sobre la cual correr las aplicaciones.

Chatbots

Son agentes de máquina que sirven como interfaces de usuario de lenguaje natural para proveedores de datos y servicios.

Correferencia

La correferencia es el conjunto de mecanismos de la lengua para establecer relaciones entre los enunciados de un texto. Su adecuada comprensión y aplicación contribuyen tanto a la comprensión como a la producción de textos.

Entidad

Representa una frase en el texto. Tiene una o más propiedades como nombre, una persona, una organización o ubicación este tiene una o más propiedades con nombre. Las entidades en este proyecto sirven para determinar si una palabra (token) es un pronombre, verbo, determinador.

Etiquetadas según su tipo: persona, organización, ubicación, evento, producto o medio.

En términos generales, las entidades se dividen en dos categorías:

Nombres propios que se asignan a entidades únicas (personas específicas, lugares, etc.) o sustantivos comunes (también llamados "nominales" en el PLN). Una buena práctica general a seguir es que, si algo es un sustantivo, se califica como una "entidad". Las entidades se devuelven como compensaciones indexadas en el texto original.

Desambiguar

La definición de desambiguar en el diccionario castellano es efectuar las operaciones necesarias para que una palabra, frase o texto pierdan su ambigüedad. Otro significado de desambiguar en el diccionario es también averiguar.

Expresiones regulares

Las expresiones regulares son patrones utilizados para encontrar una determinada combinación de caracteres dentro de una cadena de texto.

GAI

La Generación Automática de Ítems (GAI) se define como el proceso para diseñar y elaborar reactivos de una prueba que son conceptual y estadísticamente equivalentes y que se desarrollan con el apoyo de sistemas informáticos (Gierl & Lai, 2012).

Gramática

Es el estudio de reglas y principios que rigen y regulan el uso de las lenguas y a como las palabras deben estar organizadas dentro de una oración.

La gramática tiene como objetivos describir las construcciones gramaticales propias del español general, el uso adecuado de las variantes fónicas, morfológicas y sintácticas.

Está dividida en cuatro niveles: nivel fonético-fonológico, sintáctico-morfológico, léxico semántico y pragmático.

Guía para análisis sintáctico:

1. Leer con profundidad la oración, hasta entender su significado.
2. Buscar el verbo, la acción que se realiza, si no hay verbo, no hay oración.
3. Buscar el sujeto, es la persona, animal, objeto, etc. que realiza la acción del verbo.
4. Buscar el predicado., acciones realizadas por el sujeto.
5. Analizar los elementos del sujeto.
6. Determinar el tipo de predicado, predicado nominal o predicado verbal.
7. Buscar el complemento directo, preguntando al texto ¿Qué? ¿A quién?
8. Buscar el complemento indirecto, preguntando al texto ¿A quién? ¿Para quién?
9. Buscar los complementos circunstanciales, expresan circunstancias de tiempo, lugar, cantidad, etc. Son preguntas como: ¿De dónde? ¿Hacia dónde? ¿Para dónde?

Pasos del análisis morfológico:

1. Especificar el problema u objetivo.
2. Analizar que atributos lo componen. Estos pueden ser: partes físicas, procesos, funciones, aspectos estéticos, etc. Se determina si un atributo es lo suficientemente relevante para añadirlo, contestando la pregunta siguiente, ¿Seguiría existiendo el problema sin este atributo?
3. Analizar las variantes o alternativas posibles de cada atributo.
4. Combinaciones: hacer cuantas combinaciones sean posibles, tomando cada vez una variante de cada atributo. El número de atributos y variaciones determinará la complejidad de la matriz. El producto de combinaciones posibles se denomina producto morfológico.
5. Búsqueda morfológica: consiste en analizar combinaciones y ver sus posibilidades.

Sentencia

Contiene una lista de las oraciones extraídas del documento original.

Sintagma

Es una palabra o un grupo de palabras conectadas entre sí que forman una estructura con sentido y desempeñan la misma función sintáctica en la oración. Los sintagmas pueden ser de diferentes tipos y relacionarse de modo diverso en la oración.

Sintaxis

Es la parte de la gramática que observa la disposición de las palabras dentro de una frase u oración, la función que cumplen en la estructura del texto, así como sus reglas de combinación.

Es el estudio de cómo las palabras individuales en una oración se relacionan entre sí. La sintaxis y la morfología funcionan juntas para transmitir relaciones gramaticales, con diferentes lenguajes que dividen el trabajo entre ellas de manera diferente.

Morfología

Es el estudio de la estructura interna de las palabras y cómo se forman y modifican. La morfología se centra en cómo los componentes dentro de una palabra (tallos, raíz de palabras, prefijos, sufijos, etc.) se arreglan o modifican para crear diferentes significados.

El método de análisis de sintaxis (analyzeSyntax) devuelve detalles sobre la estructura lingüística del texto dado. Para cada token en el texto, la API de lenguaje natural proporciona información sobre su estructura interna (morfología) y su función en la oración (sintaxis).

Morfosintáctico

Es un parámetro gramatical que debe tenerse en cuenta al momento de escribir un texto, se puede afirmar que es la combinación entre la morfología y la sintaxis.

Permite que el texto tenga la correcta orientación o sentido.

Morfosintaxis

Evita que exista ambigüedad y no se logre obtener la secuencia real al momento de redactar un texto.

Polisemia

La polisemia es la propiedad que tiene una misma palabra para representar varios significados, que pueden generarse por ampliación o restricción del significado original.

Tokenización

El método análisis de sintaxis (analyzeSyntax) transforma el texto en una serie de tokens, que corresponden a los diferentes elementos textuales (límites de palabras) del contenido. El proceso por el cual la API de lenguaje natural desarrolla este conjunto de tokens se conoce como tokenización.

Una vez que se extraen estos tokens, la API de lenguaje natural los procesa para determinar su parte de la oración asociada (incluida la información morfológica) y el lema. Además, los tokens se evalúan y se colocan dentro de un árbol de dependencias, lo que le permite determinar el significado sintáctico de los tokens e ilustrar la relación de los tokens entre sí y las oraciones que los contienen. La información sintáctica y morfológica asociada con estos tokens es útil para comprender la estructura sintáctica de las oraciones dentro de la API de lenguaje natural.

Token

Un token es una instancia de una secuencia de caracteres en algún documento particular que se agrupan como una unidad semántica útil para el procesamiento. Un tipo es la clase de todos los tokens que contiene la misma secuencia de caracteres.

El token representa el bloque sintáctico más pequeño del texto, el cual puede emplear algunos métodos para obtener alguna parte del texto, la etiqueta, aspecto, caso, genero, estado anímico, número, persona, tiempo, entre más.

Es producto de la tokenización que divide la secuencia de texto en una serie de tokens, y cada token generalmente corresponde a una sola palabra, letra o carácter.