

|  |
| --- |
| **Universidad Internacional de La Rioja**  **Escuela Superior de Ingeniería y Tecnología**  **Máster Universitario en Análisis y Visualización de Datos Masivos** |
|  |
| Análisis de la Cronología y relaciones entre personajes bíblicos y eventos a través del tiempo |

**Trabajo Fin de Máster**

**Tipo de trabajo:** Piloto Experimental

**Presentado por:** González Cano, David

**Director/a:** Fernández García, Diego

**Resumen**

A través del uso de diferentes técnicas de Big Data y Visualización del Procesamiento de Lenguaje Natural (NPL), usando técnicas dirigidas al análisis de modernos análisis de texto, usando Python y herramientas de software libre como Google BERT.

La propuesta de trabajo de Fin de Master (TFM) tiene el propósito central de utilizar estas técnicas para analizar la cronología de los personajes bíblicos, los eventos y las relaciones que estos personajes tienen en los libros de la Biblia.

Para ello se utilizan diferentes datasets públicos que contienen variables como “fecha, “pasaje de la biblia”, mediante técnicas de NPL se analizan las variables y finalmente se crea una plataforma de visualización que sea accesible a cualquier usuario que lo requiera.

Para la recopilación, limpieza, depuración, análisis, clasificación y visualización de datos se utilizará Python con sus respectivas librerías.

Para analizar y trabajar con textos relacionados con la cronología bíblica se va a utilizar **NPL** (Procesamiento de Lenguaje Natural)

Para la visualización se está pensando en herramientas colaborativas como Google sheet que permita obtener tanto visualizaciones estáticas como dinámicas.

Lo que se espera tener de esta investigación es poder dejar un prototipo que utilizando el texto de la biblia permita usar las herramientas para hacer preguntas sobre la cronología tanto de eventos como de personajes y presente visualizaciones claras de lo requerido.

**Palabras Clave:** Procesamiento Natural de Lenguaje, NLP, Cronología Bíblica, Google BERT, GitHub. Google Colab, Visualización, Power Bi, Biblia,

**Abstract**

The Master's Thesis proposal (TFM) has the central purpose of using different Big Data and Data Visualization techniques to analyze the chronology of biblical characters, the events and the relationships that these characters have in the books of the Bible.

Using the different datasets that are public, with different variables and different formats.

Python with its respective libraries will be used to collect, clean, debug, analyze, classify and visualize data.

To analyze and work with texts related to biblical chronology, NPL (Natural Language Processing) will be used.

For visualization, collaborative tools such as Google Sheet are being considered that allow obtaining both static and dynamic visualizations.

What is expected from this research is to be able to leave a prototype that, using the text of the Bible, allows the use of tools to ask questions about the chronology of both events and characters and presents clear visualizations of what is required.

**Keywords:** Natural Language Processing, NLP, Biblical Chronology, Google BERT, GitHub. Google Colab, Visualization, Power Bi, Bible

Índice de Contenido

[1. Introducción 7](#_Toc154435968)

[1.1 Justificación 7](#_Toc154435969)

[1.2 Planteamiento del trabajo 8](#_Toc154435970)

[1.3 Estructura de la memoria 8](#_Toc154435971)

[2. Contexto y estado del arte 9](#_Toc154435972)

[2.1 Introducción al tema 9](#_Toc154435973)

[2.2. Desarrollo (base teórica): antecedentes, estudios actuales, autores de referencia 9](#_Toc154435974)

[2.2.1 NLP (Natural Lenguaje Processing) 10](#_Toc154435975)

[2.2.1.1 Word Embedding 12](#_Toc154435976)

[2.2.1.2 Pruebas 15](#_Toc154435977)

[2.2.1.3 Haciendo Predicciones 16](#_Toc154435978)

[2.2.2 Google BERT (Bidirectional Encoder Representations from Transformers) to 17](#_Toc154435979)

[2.2.3. Herramientas Colaborativas 17](#_Toc154435980)

[2.2.3.1 GitHub / GitLab / Bitbucket / Gitkraten / SourceGorge 17](#_Toc154435981)

[*2.2.3.2* Google *Colab* 18](#_Toc154435982)

[2.2.4. Bibliotecas Python 18](#_Toc154435983)

[2.2.5. La Biblia 19](#_Toc154435984)

[2.2.6. Dataset bíblicas 23](#_Toc154435985)

[2.3 Conclusiones (nexo de unión de lo investigado con el trabajo a realizar) 24](#_Toc154435986)

[3. Objetivos concretos y metodología de trabajo 24](#_Toc154435987)

[3.1. Objetivo general 24](#_Toc154435988)

[3.2. Objetivos específicos 24](#_Toc154435989)

[4. Desarrollo específico de la contribución 25](#_Toc154435990)

[4.1. Descripción detallada del Piloto experimental 25](#_Toc154435991)

[4.1.1 Qué tecnologías se utilizaron (incluyendo justificación de por qué se emplearon y descripciones detalladas de las mismas). 25](#_Toc154435992)

[4.1.2 Cómo se organizó el piloto 25](#_Toc154435993)

[4.1.3 Qué personas participaron (con datos demográficos) o qué técnicas de evaluación automática se emplearon. 25](#_Toc154435994)

[4.1.4. Cómo transcurrió el experimento. 25](#_Toc154435995)

[4.1.5 Qué instrumentos de seguimiento y evaluación se utilizaron. 25](#_Toc154435996)

[4.2 Descripción de los resultados 25](#_Toc154435997)

[4.3. Discusión 26](#_Toc154435998)

[5. Conclusiones y trabajo futuro 27](#_Toc154435999)

[5.1. Conclusiones 27](#_Toc154436000)

[5.2. Líneas de trabajo futuro 27](#_Toc154436001)

[Bibliografía 29](#_Toc154436002)

**Índice de figuras**

[Figura 1 Taxonomía del machine Learning dentro de la Inteligencia Artificial. Basado en Panesar, 2019 9](#_Toc154175686)

[Figura 2 Generando un modelo estadístico con el algoritmo de aprendizaje automático 13](#_Toc154175687)

[Figura 3 Ejemplo Predicción una estructura de árbol de dependencia usando un modelo estadístico 14](#_Toc154175688)

[Figura 4 Parte de la visualización de cronología de la Biblia (Aschmann, 2022) 21](#_Toc154175689)

**Tabla de Contenido**

[Tabla 1 Simplificado Espacio vectorial de palabras 13](#_Toc154435938)

[Tabla 2 Libros de la Biblia Hebrea o Antiguo testamento 20](#_Toc154435939)

[Tabla 3 Libros Antiguo testamento mayoría biblias cristianas 22](#_Toc154435940)

# 1. Introducción

## 1.1 Justificación

Meditando sobre los proyectos que dan gasolina en el día a día y que se vuelven retos ultraístas, el leer la Biblia es uno de ellos, desde su inicio hasta el final contando con más de cuarenta escritores y aproximadamente 773.746 palabras en un libro que nos lleva desde el inicio del mundo, va recorriendo con eventos, personajes los diferentes tiempos de la humanidad y ahí, en el mismo texto lleno de aventuras nos sumergimos en cual será el final de la humanidad con señales que se conectan en los 66 libros divididos en 1189 capítulos y 31.102 versículos y 3.5 35.666.480 letras, Biblia Reina Valero 1960. (Concepto y definicion Net, 2023)

La secuencia de los tiempos, los personajes, los eventos en los escritos no son tan fáciles de visualizar debido a que las fechas no están implícitas en el texto y se necesitan estudios teológicos y contextos históricos para poder entender la cronología de los eventos con sus respectivos personajes.

Para llevar a cabo esta investigación se van a utilizar diferentes técnicas, métodos y herramientas para encontrar una forma de analizar y presentar cronológicamente eventos, personajes utilizando textos que están ya colocados en la web y también si es posible desarrollar un prototipo de base de datos que nos permita ir actualizando las fechas en una línea de tiempo actualizable por los investigadores de este tema.

La necesidad de esta investigación surge de la diversidad de autores en sus escritos, las figuras literarias y estilísticas que se utilizan como metáforas, símiles, personificaciones, parábolas, hipérboles, ironías, y muchas más; la identificación de fechas en cada evento, la no existencia hasta el momento de una base de datos estructurada que coloque cada evento o personaje en una línea de tiempo.

Por todo ello se propone una herramienta a la población interesada en poder dar un contesto en el tiempo de los escritos ya sea por estudio u otro interés dando una visualización en una forma clara y utilizando herramientas de datos masivos.

En la población mundial aproximadamente de 8300 millones y con un cálculo de 2500 millones de cristianos en el mundo (EL orden Mundial EOM, 2019), tienen como base de su fe la Biblia, la cual se divide en Antiguo y nuevo testamento. Así a esta población se puede sumar el pueblo judío aproximadamente 13 millones (Embajada de Israel en España, s.f.), los cuales su base de lectura en su creencia es el Antiguo Testamento. ya que tienen una división diferente y se conoce como la biblia hebrea o Tanaj la cual está dividida en tres partes: la Torá (ley) , los Nevi’im (profetas) y los Ketuvim (escritos), además los judíos no consideran la parte de los libros del nuevo testamento como parte de las escrituras.

## 1.2 Planteamiento del trabajo

Utilizando técnicas de procesamiento de lenguaje natural (NLP) y modelos de machine Learning que permitan extraer información y procesarla cronológicamente. Creando una base de datos estructurada que manaje y permita trabajar con fechas de los personajes bíblicos y utilizar herramientas de visualización que permitan de una forma fácil presentar los eventos y personajes en forma dinámica.

## 1.3 Estructura de la memoria

La primera parte es describir las técnicas de manejo de Texto

Lo segunda parte está dirigida al análisis de la biblia como un dataset y como se logra organizar para obtener información cronológica

El tercer es el prototipo planteado de la unión de las técnicas y la estructura bíblica con herramientas de Visualización.

# 2. Contexto y estado del arte

## 2.1 Introducción al tema

Es interesante pensar que cuando hablamos de tiempo, aunque no lo sentimos material, tiene estados de pasado, presente y futuro; y cuando lo graficamos con personajes y eventos se nos convierte en una línea de tiempo, una herramienta visual para presentar cualquier información que mezcla datos y eventos formando una función de sucesión temporal.

La cronología bíblica se refiere al estudio y ordenamiento de los eventos históricos descritos en la biblia en secuencia temporal y busca determinar fechas aproximadas de eventos, así como la duración de periodos y la secuencia en la que ocurrieron.

El análisis de texto mediante el procesamiento del Lenguaje natural aplicado a el texto bíblico tiene varios temas de interés, asi como la identificación de personajes, eventos y sus relaciones en el tiempo usando diferentes técnicas.

## 2.2. Desarrollo (base teórica): antecedentes, estudios actuales, autores de referencia

El proyecto se enmarca en la Disciplina de la inteligencia artificial que, según la definición de la real academia española, Inteligencia Artificial: “es una Disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o el razonamiento lógico” (ASALE, n.d.)

Como un subcampo encontramos el procesamiento de lenguaje natural que trata el procesar y analizar los datos de lenguaje natural. Esto incluye enseñarles a máquinas para interactuar con humanos con un lenguaje que se desarrolla naturalmente a través del uso, creando algoritmos de Machine Learning que trabajan con grandes datasets desconocidas. Hoy en día se pueden usar algoritmos para observar cuyas reglas semánticas y gramaticales son bien conocidas. No es sorprendente que generar y comprender el lenguaje natural sea la tarea más prometedora y, al mismo tiempo, más desafiante, involucrada en NPL. (Vasiliev, 2020)

.

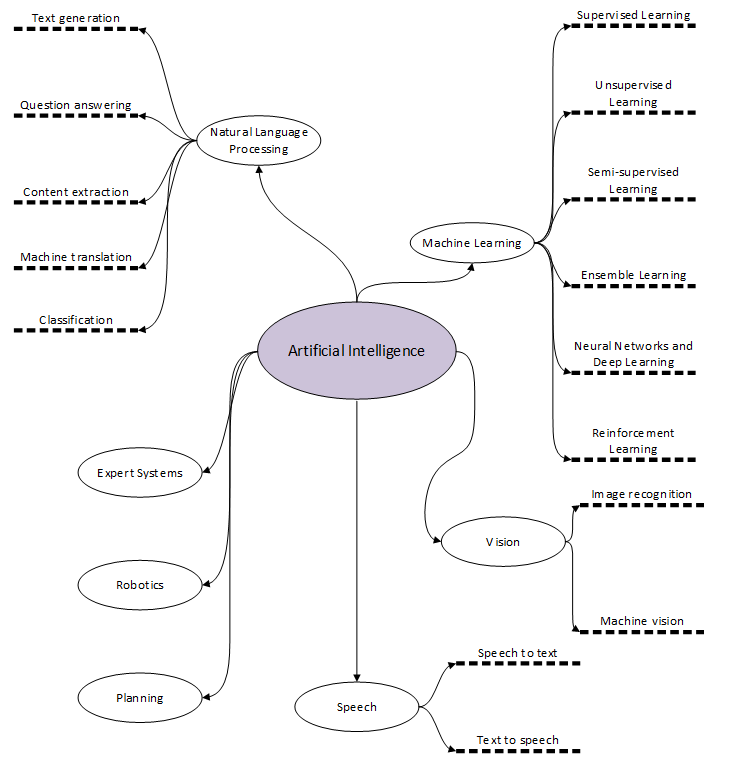


Figura 1 Taxonomía del machine Learning dentro de la Inteligencia Artificial. Basado en Panesar, 2019

.

### 2.2.1 NLP (Natural Lenguaje Processing)

“Un idioma no son solo palabras. Es una cultura, una tradición, una unificación de una comunidad, una historia completa que cruza lo que es una comunidad. Todo está incorporado en un lenguaje".

-Noam Chomsky

Es un campo entre la intersección de ciencias de computación e inteligencia artificial y logística. Es un tema de construir sistemas que procesan y entienden el lenguaje humano.

(Sowmya Vajjala, 2020)

El lenguaje natural es el primer medio de comunicación entre humanos desde los comienzos de la humanidad. Los computadores siendo objetos no emocionales se comunican en lenguaje de máquina, una codificación de 0s y 1s. La forma en que las maquinas puede entender el lenguaje es lo que se denomina Procesamiento de lenguaje natural.

Lo primero que hay que decir es que tenemos que traducir el lenguaje natural de palabras o texto a números.

El término que se maneja para la representación vectorial de los datos es “embedding” y se utiliza para diversos campos como el NPL, la recomendación de contenido y la visión por computadora. Los algoritmos comunes utilizados en diferentes contestos son

1. Word Embedding
   * Word2Vec: Desarrollado por Google, utiliza una red neuronal para aprender representaciones vectoriales de palabras
   * GloVe (Global Vector for Word Representación) Utiliza estadísticas globales de coocurrencia para generar embeddings
2. Documento Embeddings
   * Doc2Vec: Una extensión de Word2vec que aprende representaciones vectoriales para documentos.
3. Embeddings para imágenes
   * CNN (Convulutional Neural Network) Embeddings: Las capas de intermedias de una CNN pueden utilizarse como embeddings para imágenes
   * Siamese Networks: Se utilizan para aprender embeddings que representa similitudes y diferencias entre pares de imágenes.
4. Embeddings para Datos Tabulares
   * FastText: Se utiliza no solo para palabras, sino también para representar datos tabulares.
   * Entity Embeddings of Categorical Variables: Se emplea en conjuntos de datos con variables categóricas.
5. Graph Embeddings
   * Node2Vec: Extiende el concepto de Word2Vec para grafos, generando embeddings para nodos.
   * GraphSAGE (Graph Sample and Aggregated): Genera embeddings para nodos agregando información de sus vecinos.
6. Embeddings para secuencias temporales
   * LSTM (Long Short-term memory) Embeddings: Redes neuronales recurrentes (RNN) que pueden aprender representaciones con secuencias temporales.
   * Transformer Embeddings: La arquitectura transformer, utilizada en modelos como GPT, puede generar embeddings para secuencias temporales.

La elección del embedding depende del tipo de datos y del problema especifico que se aborde. Otro punto interesante es que se pueden combinar.

#### 2.2.1.1 Word Embedding

Se puede definir como una técnica de procesamiento de lenguaje natural que convierte lenguaje humano a vectores matemáticos. Se hace un mapeo de palabras a vectores de números reales, lo que significa que cada palabra está en el vector.

Una vez se tiene la matriz que mapea palabras a vector numérico, se puede realizar acciones aritméticas sobre dichos vectores. Así se puede tener similitudes semánticas (sinónimos) de palabras, frases y aun documentos enteros. Así se puede usar información para determinar programáticamente de que texto se trata.

Matemáticamente, determinar la semántica similar entre dos palabras se reduce a calcular el coseno similar entre el vector correspondiente, o calcular el coseno de el angulo entre los vectores.( Mas detalles cap 5 Working with Word vectors libro (Vasiliev, 2020))

Para hacer un análisis de cómo se trabaja con vectores de texto se utiliza Spacy que es una biblioteca de procesamiento de lenguaje natural de código abierto diseñada para realizar tareas de procesamiento de texto de manera más eficiente, que ofrece herramientas para realizar actividades como tokenización, lematización, etiquetado gramatical, reconocimiento de entidades nombradas y análisis de dependencias sintácticas.

**Trabajando con vectores de Palabra**

Los vectores de palabras son números reales que permiten que las maquinas entiendan el lenguaje natural. El cálculo de las similitudes de la semántica de diferentes textos, permite por ejemplo clasificar textos según el tópico a cubrir.

Lo primero es un vistazo conceptual en los vectores de palabra, así como una idea de cómo matemáticamente calcula las similitudes semánticas entre palabras representadas en la forma de un vector. Sé mapean palabras a vectores de números reales que reflejan similitudes semánticas de las palabras. Se puede imaginar un espacio de vector de palabras como una nube en el cual el vector de palabras con similitudes está localizado cercanamente. Por ejemplo, un vector representando la palabra “Salvador” debería estar cerca a el vector de la palabra “Mesías”. Para generar estos vectores se debe estar disponible a codificar el significado de estas palabras. Hay un enfoque para codificar el significado

**Definir significado con coordenadas.**

Una manera para generar un vector de palabras significativas es o asignar un objeto o categoría desde el mundo real a cada coordenada de un vector de palabras. Por ejemplo, para las siguientes palabras: Roma, Italia Atenas y Grecia. El vector de palabras debería matemáticamente reflejar de hecho que roma es capital de Italia y Atenas no tiene igual relación con Italia. También que Atenas y roma son ciudades capitales y además que Grecia y Italia son Países.

La Tabla siguiente muestra la matriz de este vector.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pais | Capital | Grecia | Italia |
| Italia | 1 | 0 | 0 | 1 |
| Roma | 0 | 1 | 0 | 1 |
| Grecia | 1 | 0 | 1 | 0 |
| Atenas | 0 | 1 | 1 | 0 |

Tabla 1 Simplificado Espacio vectorial de palabras

Así se distribuye el significado de cada palabra entre sus coordenadas en un espacio de cuatro dimensiones, representando las categorías “País”, “Capital”, “Grecia”, “Italia”. En este simple ejemplo una coordenada puede tomar un valor de cero o uno indicando si la palabra pertenece o no a la categoría.

Una vez se tiene el vector de números que corresponden al significado de las palabras, se puede usar el vector aritmético para entender mejor el significado de una palabra. Para descubrir de cual país es capital la ciudad de Atenas se puede usar la siguiente ecuación, donde cada valor representa su vector correspondiente y X es un vector desconocido

*Italia – Roma = X – Atenas*

Esta ecuación expresa una analogía en cual X representa el vector de palabra que tiene la misma relación para Atenas como Italia la tiene para Roma.

Para resolver para X, Se puede reescribir la ecuación como la siguiente:

*X = Italia – Roma + Atenas*

Primero se resta el vector Roma del vector Italia para restar el correspondiente elemento del vector. Así a suma de los vectores es la siguiente

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operación |  | País | Capital | Grecia | Italia |
|  | Italia | 1 | 0 | 0 | 1 |
| - | Roma | 0 | 1 | 0 | 1 |
| + | Atenas | 0 | 1 | 1 | 0 |
|  | Grecia | 1 | 0 | 1 | 0 |

Tabla 2 Operación matemática vectorial en un espacio vectorial de palabras

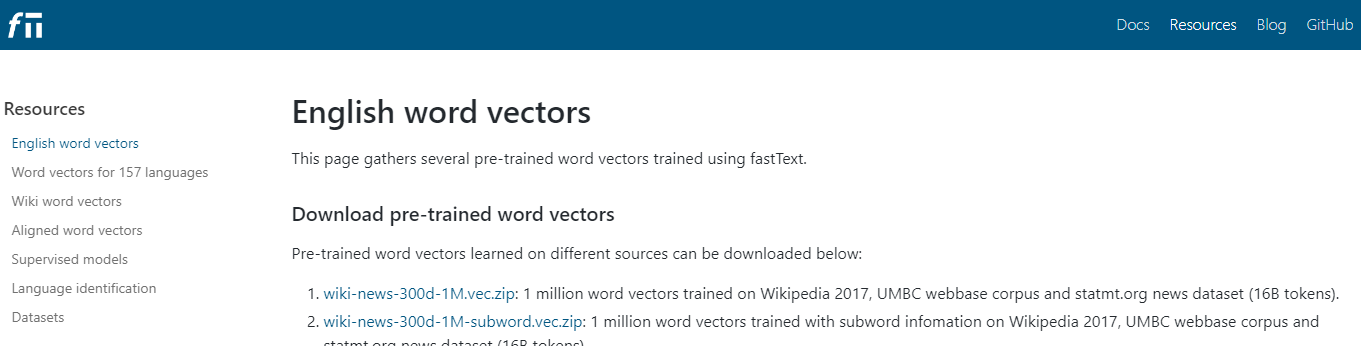
Por restar el vector de la palabra Roma de del vector de la palabra Italia y sumar el vector de la palabra Atenas, se tuvo el resultado que es igual al vector Grecia.

**Usando dimensiones para representar significados**

Con el ejemplo anterior se tomaron cuatro categorías, realmente en el mundo real de los vectores de palabras pueden llegar a ser decenas de miles. El vector de palabras de estas proporciones se vuelvo impráctico para muchas aplicaciones, porque esto requiere una matriz inmensa de Word-embedding. Por ejemplo, para trabajar unas 10.000 categorías y 1.000.000 de entidades a codificar, se necesitaría un 10.000 x 1.000.000 matriz de embedding, haciendo que las operaciones de tiempo de consumo demasiado largo. Así para lograr reducir el tamaño la matriz embedding seria reducir el número de categorías que se involucran en el espacio del vector.

En vez de utilizar las coordenadas que representan todas las categorías, se usa la distancia entre vectores para cuantificar y categorizar semánticamente las similitudes. Las dimensiones individuales no tienen el significado en vez de esto tienen las representaciones del espacio en el vector y las distancias entre los vectores indican las similitudes a el correspondiente significado de las palabras.

El siguiente es un fragmento del espacio del vector de palabras extractado de fasttest, una librería de vector de palabras, la cual se puede bajar de la siguiente dirección <https://fasttext.cc/docs/en/english-vectors.html>



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| compete | -0.0535 | -0.0207 | 0.562 |  | -0.0389 | -0.0389 |
| reviwer | 0.0337 | 0.2013 | -0.15787 |  |  |  |
| Upper |  |  |  |  |  |  |
| Mentor |  |  |  |  |  |  |
| reviewer |  |  |  |  |  |  |

Cada línea representa una palabra como un vector de números reales en multidimensionales espacios. Gráficamente, se puede representar un vector de 300 dimensiones como uno de dos o tres dimensiones proyectado. Para preparar la proyección se usan las primeras dos o tres coordenadas de el vector respectivamente.

FRAGMENTO PAG 66

Una detalle interesante es que las lineas de Grecia-Athenas y Italia-Roma, respectivamente son paralelas, lo que demuestra es que los vectores en el diagrama pais-capital tienen una relacion.

**El metodo de Similitud (similitary)**

Se puede encontrar similitud entre dos doc, token o span.

**Para entender el tema definamos varios conceptos**

**Container:**

**Token:**

**Span:**

**Doc:**

En la lbiblioteca de Spacy cada objeto del container tiene un metodo de similitud que permite que se calcule la semantica de similitud entre dos container de cualquier tipo de comparacion entre dos vectores de palabra. Para calclar la similitud de spans y documentos, lo cual no tiene sus propios vectores de palabras spaCy promedia los vectores de palabras de los tokens que ellos contienen.

Se puede calcular la similitud de semantica de dos objetos de container aun si los dos objetos son diferentes. Por ejemplo se puede comparar un objeto token con un objeto span (fragmento de un documento) , un objeto san con un documento y asi sucesivamente.

El siguiente ejemplo calcula dos objetos , una para el documento completo y otro para el fragmento especifivo y luego calcula la similitud entre ellos.

*>>> doc =nlp(‘Yo quiero una manzana verde”)*

*>>> doc.similarity(doc[2:5])*

*0.7305813588233471*

*El codigo anterior es pythhon usando la biblioteca Spacy y la expresión `doc[2:5]` se refiere a una "rebanada" (slice) de tokens en un objeto `Doc` en spaCy. Desglosemos lo que significa:*

*- `doc`: Es un objeto `Doc` que representa un documento procesado por spaCy, que contiene información lingüística sobre el texto.*

*- `[2:5]`: Es una notación de "rebanada" (slice) en Python. En este contexto, selecciona un subconjunto de tokens del objeto `Doc`. En spaCy, la indexación de tokens comienza desde 0. Entonces, `doc[2:5]` selecciona los tokens desde el tercer token hasta el quinto token (sin incluir el quinto).*

*Por ejemplo, si `doc` contiene la frase "Esto es un evento de la antigüedad", entonces `doc[2:5]` seleccionaría los tokens "un", "evento", "de".*

*La notación de rebanada en Python es `[inicio:final]`, y selecciona los elementos desde `inicio` hasta `final-1`. En este caso, selecciona los tokens desde el tercer token (índice 2) hasta el quinto token (índice 4) en el objeto `Doc`.*

Algo interesante es que cuando se realiza la similitud comparado objetos consigo mismo el resultado es un 1.

El metodo de similitud puede reconocer palabras que pertenecen a la misma o similar categoria y que frecuentemente aparecen relacionando contextos, mostrando un alto nivel de similitud.

Escogiendo plabras claves para calculo de similitudes.

El metodo de similitud calcula similitudes semanticas , pero el resultado es mas útil para el calculo escogiendo las palabras claves para comparar. Para entender el porque, considere el siguiente fragmento.

Se puede considerar clasificar el texto en una variedad de diferentes maneras dependiendo del conjunto de categorías que se van a usar. Por ejemplo, Si estamos buscando el texto acerca de las plantas más altas del planeta, la frase “Arboles altos” y “en el mundo” serán las claves. Comprando esta frase con “Plantas Altas” y “sobre el planeta” deberían tener una similitud altísima semánticamente. Se pueden extraer trozos sustantivos usando una propiedad doc.noun.chunk y así chequear la similitud de trozos sustantivos y la búsqueda de frases usando este método.

Si buscamos lugares en el mundo. “Bogotá” será la palabra clave. Realmente no sabemos de antemano cual nombre geopolíticamente puede ocurrir en el texto. Puede ser Bogotá o Amazonas. Pero sea lo que sea, esto debería ser semánticamente similar a la palabra “geografía”. La cual se puede comparar con otros sustantivos del texto. Si podemos determinar que hay un alto grado de similitud, podemos asumir que el nombre de la entidad en cuestión representa el nombre geopolítico.

===

Se pueden generar los números para poner en vector usando un algoritmo de aprendizaje automático. EL machine Learning un subcampo de la inteligencia artificial crea sistemas de computadores que pueden automáticamente aprender de datos que no han sido explícitamente programados. Los algoritmos pueden hacer predicciones acerca de nuevos datos, aprender a reconocer imágenes y discursos, clasificar fotos y documentos de texto, automatizar controles y ayudar en desarrollo de juegos.

El aprendizaje automático (Machine Learning) permite que las computadoras realicen tareas que serían difíciles, sino en muchos casos imposible de hacerlo.

Por ejemplo, para realizar un programa normal de jugar el algoritmo ajedrez, el algoritmo debe contener condiciones de si entonces (if...else) que necesitan ser definidas. Así se desarrolle exitosamente, el programa va a tener puntos débiles en su lógica que pueden tomar ventaja en el mismo juego antes que se hagan correcciones en el código del programa.

En contraste las aplicaciones construidas sobre machine Learning no confían en la predeterminada, pero si en la capacidad de aprender de experiencias del pasado. La aplicación mira las posiciones jugadas y recuerda de juegos pasados y hace el movimiento a la mejor posición. Almacena las experiencias pasadas en un modelo estadístico. (capitulo 8… Cual es el modelo estadístico en NPL (Vasiliev, 2020))

**Modelo Estadísticos**

En Spacy, además de generar vectores de palabras, permite lograr tres tareas: Análisis de dependencia semántica (determinar la relación entre las palabras en la oración), parte del etiquetado del discurso (identificar sustantivos, verbos y otros partes de la oración y reconocimiento de la entidad nombrada (organizar los sustantivos apropiados dentro de categorías como gente, organizaciones y localizaciones.

El ciclo típico del Machine Learning tiene tres pasos:

Entrenamiento Modelo

La primera fase es alimentar el algoritmo con una gran cantidad de datos. Para que sea confiable se debe proveer de una suficiente cantidad de datos en la entrada, al pensar en NPL se puede tener plataformas como Wikipedia y Google News que contiene suficiente texto para alimentar virtualmente cualquier algoritmo de aprendizaje automático (machine Learning). Para modelos específicos se debe buscar sitios que puedan ser para el caso de estudio.

Figura 2 Generando un modelo estadístico con el algoritmo de aprendizaje automático

La figura anterior proporciona una descripción de alto nivel de la etapa de entrenamiento del modelo.

El modelo procesa grandes volúmenes de datos de texto para entender cuales palabras comparten característica, entonces este crea vectores de palabra para aquellas palabras que reflejan características que comparten

Para saber que modelo estadístico se necesita Capitulo 8.

El espacio del vector de palabras no solo tiene el componente del modelo estadístico. La estructura actual es típicamente más complicada, provee una manera de extraer las características lingüísticas para cada palabra dependiendo del contexto en donde aparece.

En capítulo 10 se aprenderá como entrenar un ya existente modelo entrenado con nuevos ejemplos y uno completamente blanco desde cero.

#### 2.2.1.2 Pruebas

Una vez que el modelo esta entrenado, se realizan pruebas para saber que tan bien esta la ejecución. Para hacer pruebas se alimenta el texto que no ha sido alimentado antes y así comprobar si puede identificar con éxito las similitudes semánticas y otras características aprendidas durante el entrenamiento.

#### 2.2.1.3 Haciendo Predicciones

Con las pruebas funcionando se pasa a usar el modelo para hacer predicciones en su aplicación de PNL. Por ejemplo, puede usarlo para predecir una estructura de árbol de dependencia sobre el texto que ingresa, como se muestra en la Figura 1-2. Una estructura de árbol de dependencia representa las relaciones entre las palabras en una oración.

Arbol d

Figura 3 Ejemplo Predicción una estructura de árbol de dependencia usando un modelo estadístico

### 2.2.2 Google BERT (Bidirectional Encoder Representations from Transformers) to

Es un modelo de procesamiento de lenguaje natural desarrollado por Google. Utiliza la arquitectura de Transformers para entender el contexto de las palabras en una oración, teniendo en cuenta tanto palabras anteriores como siguientes. Esto mejora la capacidad del modelo para comprender el significado de una palabra en función de su contexto, lo que resulta en una mejor comprensión del lenguaje natural y una mejora de la calidad de las respuestas generadas por los modelos de procesamiento de lenguaje natural.

(Ravichandiran, January 2021) compartir sus proyectos de software de manera centralizada.

### 2.2.3. Herramientas Colaborativas

#### 2.2.3.1 GitHub / GitLab / Bitbucket / Gitkraten / SourceGorge

Usar una plataforma que permita la colaboración en este proyecto puede ser la base para

que analistas de datos con interés en el procesamiento natural del lenguaje, sea experimentado que puedan aportar en ideas o en el propio desarrollo pueden ser un aspecto importante para tener éxito en el proyecto que se está desarrollando.

Git es una tecnología que empezó en el 2005 …

Comandos Linux

GitHub es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git, Permite a los desarrolladores trabajar juntos en proyectos de software, facilitando el seguimiento de cambios en el código fuente, la colaboración entre equipos y la gestión eficiente de proyectos. Los usuarios pueden alojar sus repositorios de código en GitHub, lo que significa que pueden almacenar.

Algunos aspectos útiles para desarrollar el proyecto:

**Control de versiones:** Con lo que facilita un contexto de trabajo colaborativo. Lleva el control de versiones muy importante en el mundo donde una o varias personas están desarrollando un proyecto,

**Repositorios**: Es el espacio donde se almacena todo el código fuente, archivos de configuración, documentos otros recursos relacionados con un proyecto. Puede ser publico (accesible para todos) o privado (accesible solo para personas autorizadas)

**Colaboración de desarrolladores,**

Seguimiento a problemas(bugs) Los usuarios pueden informar bugs, solicitar nuevas características o discutir ideas a través del seguimiento de problemas GitHub.

**Despliegue:** Utiliza un despliegue en sitios de web y aplicaciones a través de GitHub Pages, una función que permite alojar sitios web directamente desde los repositorios de GitHub

GitHub crea un ambiente que permite almacenar código en un servidor remoto, dando la habilidad de compartir el código con otras personas y hacer más fácil que una persona agregue, modifique o borre código del mismo archivo y proyecto, mientras que guarda una fuente de verdad para este archivo. (Sarah Guthals, 2019)

#### *2.2.3.2* Google *Colab*

### 2.2.4. Bibliotecas Python

Hay varias bibliotecas en Python que son de Python son elementos centrales del lenguaje de programación y como se puede escribir, reusar y compartir código en Python.

Para el NLP procesamiento de lenguaje natural hay varias bibliotecas poderosas y populares como son:

NLTK: Es una biblioteca extensa para trabajar con datos de texto. Proporciona herramientas de tokenizacion, stemming, lematización, análisis sintético y mas. (NLTK, Consultada en 2023)

spaCy: Es una biblioteca abierta, diseñada para ser rápida y eficiente. Proporciona modelo preentrenados. Tiene muy buen rendimiento en grandes cantidades de texto.

TextBlob: Es una biblioteca fácil de usar para el procesamiento de texto basado en NLTK. Ofrece una API para tareas sencillas, como análisis de sentimiento, extracción de frases clave y clasificación de texto. (Textblob: Simplified Text Processing, Consultado en 2023)

Gensim: Esta biblioteca se centra en el modelado de temas y similitud de documentos. Es útil para la construcción y entrenamiento de modelo de vectores de palabras (Word embedddings) y modelos de tópicos. (GENSIM topic modeling for humans, Consultada en Nov 22 2023)

Transformer: Proporciona implementaciones de modelos de lenguaje preentrenados de vanguardia, como BERT, GPT-2 . (Tranformers Huggin Face, Consultado en nov 22 2023)

### 2.2.5. La Biblia

La base teórica para desarrollar este punto es extensa y a veces muy controversial, pero se planteará el ser lo más objetivos posibles, estudiando libros de historiadores, teólogos, pastores de iglesias, y revisando documentos que nos den la base en cuanto a tener un texto que sea aceptado en la mayoría.

La palabra “Biblia” tiene raíz del latín medieval y deriva del griego byblos, que significa “libros”, pero si vamos hacia su origen etimológico del termino llegamos a que byblos era una antigua ciudad Fenicia situada sobre las costas del actual Líbano. Los fenicios inventaron el alfabeto que todavía utilizamos y les enseñaron a escribir a los griegos. Desde Biblos los fenicios exportaban los papiros en los que fueron escritos los primeros libros (El papiro una planta semejante al junco era abierta en tiras que se humedecían y entretejían. Una vez secas, constituían un excelente papel en que escribir.) Aunque byblos originalmente significa “papiro” en griego, con el tiempo paso a significar “libros” Los libros \*tal como lo conocemos deben su denominación a la antigua ciudad. (Davis, 1998).

Ahora bien, el termino no es solo un libro sino la recopilación de muchos libros que fueron escritos en un tiempo aproximado de 4000 años por diferentes autores que trataron temas de leyes, poesía, filosofía e historia sabiduría. La reunión y organización de la biblia también se vuelve en un tema de estudio interesante, ya que dependiendo del grupo religioso que se esté basado su organización cambia, por ejemplo, la biblia de un judío, no es la misma de un católico y la biblia de un católico no es la de un protestante.

La Biblia se divide en Antiguo Testamento y Nuevo Testamento. Es interesante preguntarse a que se refiere la palabra testamento, que tiene varios significados. Uno es algo que mucho no quisiéramos pensar y es la la última voluntad que es un documento legal para el uso de los bienes terrenales de un difunto. Otro significado es el dejar evidencia de algo, por ejemplo …….. y la utilizada para aludir a nuestro tema en cuestión es la manera antigua de decir “pacto” que aludía a un acuerdo o contrato. O sea, el antiguo testamento era el pacto entre Dios y su pueblo y con el nuevo testamento es un nuevo pacto a través de la vida, la muerte y la resurrección de Jesús.

Para los judíos que no tienen sino el antiguo testamento ya que no creen en el nuevo testamento escrita hebreo antiguo se puede decir que es el equivalente al antiguo testamento de la biblia de los cristianos. Se divide en tres: Tora, Profetas y escrituras

|  |  |  |
| --- | --- | --- |
| **Tora** | **Profetas** | **Escrituras** |
| Génesis  Éxodo  Levítico  Números  Deuteronomio | Jesúa  Jueces  Samuel (I)  Samuel (II)  Reyes (I)  Reyes (II)  Isaías  Jeremías  Ezequiel  Oseas  Joel  Amos  Abdías  Jonás  Miqueas  Nahum  Habacuc  Sofonías  Hageo  Zacarias  Malaquías | Salmos  Proverbios  Job  Cantar de Los cantares  Ruth  Lamentaciones  Eclesiastés  Ester  Daniel  Esdras  Nehemías  Crónicas (I)  Crónicas (II) |

Tabla 3 Libros de la Biblia Hebrea o Antiguo testamento

La biblia para la mayoría de los cristianos antiguo testamento

|  |
| --- |
| Libros Antiguo Testamento en la mayoría de las biblias cristianas |
| Génesis  Éxodo  Levítico  Números  Deuteronomio  Josué  Jueces  Ruth  Samuel (I)  Samuel (II)  Reyes (I)  Reyes (II)  Crónicas (I)  Crónicas (II)  Esdras  Nehemías  Ester  Job  Salmos  Proverbios  Eclesiastés  Cantares  Isaías  Jeremías  Lamentaciones  Ezequiel  Daniel  Oseas  Joel  Amos  Abdías  Jonás  Miqueas  Nahum  Habacuc  Sofonías  Hageo  Zacarias  Malaquías |

Tabla 4 Libros Antiguo testamento mayoría biblias cristianas

Otro punto que de los autores hay controversias pero que en los mismos escritos se pueden extractar quien es el que los escribe, aunque si hay libros que siguen con diferentes teorías de quien es el autor, Un ejemplo es Deuteronomio que se le considera a Moisés el autor, pero en el último capítulo data su muerte, por lo que probablemente este fue escrito por Josué. (Jesúa en el Tora)

* La Biblia como texto base (Ya digitados o en sitios web)
  + Antiguo testamento
  + Nuevo Testamento
  + Libros que ayudan en la cronología libros judíos y históricos.
* Procesamiento de Lenguaje Natural, por sus siglas en ingles Natural Language Processing.
* El método de visualización se esta investigando con varias herramientas, Google Sheet, Google Bert, Power Bi

Un primer sitio consultado es un sitio web dedicado a la cronología bíblica realizada por Rick Aschman la cual tiene un estudio que presenta datos y una visualización que nos da una orientación hacia el uso de datos para presentar eventos, personajes y fechas con una visualización en una línea de tiempo. (Aschmann, 2022)

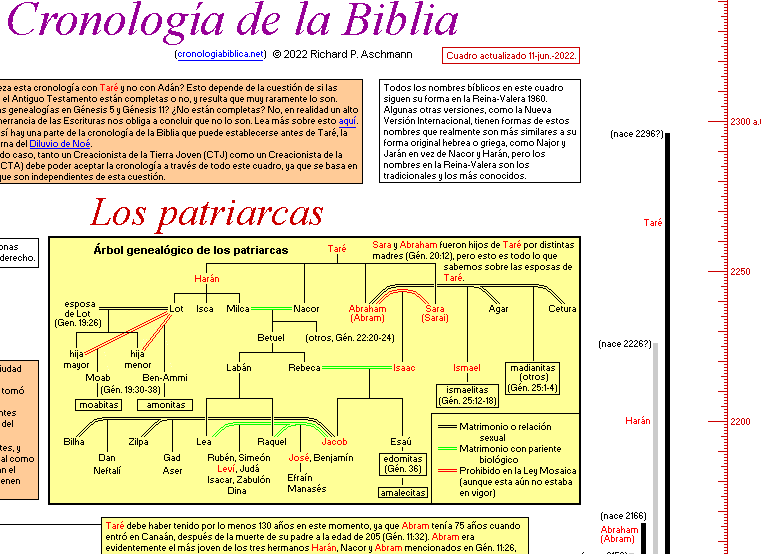


Figura 4 Parte de la visualización de cronología de la Biblia (Aschmann, 2022)

<https://aschmann.net/BibleChronology/castellano.html>

### 2.2.6. Dataset bíblicas

------------------------

Lo siento, pero no es posible proporcionar un dataset específico de ChatGPT, ya que el modelo y los datos asociados son propiedad de OpenAI y no están disponibles públicamente para su descarga o distribución. Además, la información utilizada para entrenar a ChatGPT es diversa y proviene de una variedad de fuentes en línea, por lo que no se puede atribuir a un conjunto de datos específico.

Si estás interesado en entrenar un modelo específico para comprender y generar texto relacionado con la Biblia, te recomendaría buscar datasets disponibles públicamente que contengan textos bíblicos. Puedes encontrar versiones digitales de la Biblia en diversos formatos, y algunos sitios web ofrecen datasets gratuitos o de acceso abierto que podrías utilizar para este propósito.

Algunos lugares donde podrías buscar son:

1. **Proyecto Gutenberg:** El Proyecto Gutenberg ofrece una amplia colección de libros, incluida la Biblia, en formato de texto plano.
2. **Sitios web religiosos:** Algunos sitios web religiosos pueden proporcionar acceso a textos bíblicos en formatos que podrías utilizar para entrenar tu modelo.
3. **Repositorios de datos abiertos:** Plataformas como GitHub y Kaggle pueden tener datasets relacionados con la Biblia o textos religiosos.

--------

## 2.3 Conclusiones (nexo de unión de lo investigado con el trabajo a realizar)

* Se esta trabajando un grupo de pastores en Colombia que serán parte del proceso de analizar las dificultades y serán parte de la investigación, por medio de entrevistas y soporte en el prototipo de parte del uso.

# 3. Objetivos concretos y metodología de trabajo

## 3.1. Objetivo general

El objetivo general de este piloto experimental es desarrollar un sistema de análisis de datos masivos y visualización que facilite la comprensión de la narrativa bíblica en los contextos históricos de tiempo y cada momento de la historia.

## 3.2. Objetivos específicos

* Desarrollar un prototipo de modelo NLP que permita extraer información cronológica de los textos bíblicos.
* Crear una base estructurada de las fechas de los personajes y eventos bíblicos.
* Utilizar herramientas de visualización para presentar los eventos y personajes en forma dinámica.

# 4. Desarrollo específico de la contribución

## 4.1. Descripción detallada del Piloto experimental

En el capítulo de Objetivos y Metodología del Trabajo ya habrás descrito a grandes rasgos la metodología experimental que vas a seguir. Pero si tu trabajo se centra en describir un piloto, deberás dedicar un capítulo a describir con todo detalle las características del piloto. Como mínimo querrás mencionar:

### 4.1.1 Qué tecnologías se utilizaron (incluyendo justificación de por qué se emplearon y descripciones detalladas de las mismas).

Datasets

### 4.1.2 Cómo se organizó el piloto

### 4.1.3 Qué personas participaron (con datos demográficos) o qué técnicas de evaluación automática se emplearon.

### 4.1.4. Cómo transcurrió el experimento.

### 4.1.5 Qué instrumentos de seguimiento y evaluación se utilizaron.

Qué tipo de análisis estadísticos se ha empleado (si procede).

## 4.2 Descripción de los resultados

En el siguiente capítulo deberás detallar los resultados obtenidos, con tablas de resumen, gráficas de resultados, identificación de datos relevantes, etc. Es una exposición objetiva, sin valorar los resultados ni justificarlos.

## 4.3. Discusión

Tras la presentación objetiva de los resultados, querrás aportar una discusión de los mismos. En este capítulo puedes discutir la relevancia de los resultados, presentar posibles explicaciones para los datos anómalos y resaltar aquellos datos que sean particularmente relevantes para el análisis del experimento.

El piloto experimental propuesto enfocado a contribuir …….

En este capítulo debes desarrollar la descripción de tu contribución. Es muy dependiente del tipo de trabajo concreto, y puedes contar con la ayuda de tu director para estudiar cómo comunicar los detalles de tu contribución.

# 5. Conclusiones y trabajo futuro

## 5.1. Conclusiones

Este último bloque es habitual en todos los tipos de trabajos y presenta el resumen final de tu trabajo y debe servir para informar del alcance y relevancia de tu aportación.

Suele estructurarse empezando con un resumen del problema tratado, de cómo se ha abordado y de por qué la solución sería válida.

Es recomendable que incluya también un resumen de las contribuciones del trabajo, en el que relaciones las contribuciones y los resultados obtenidos con los objetivos que habías planteado para el trabajo, discutiendo hasta qué punto has conseguido resolver los objetivos planteados.

## 5.2. Líneas de trabajo futuro

Finalmente, se suele dedicar una última sección a hablar de líneas de trabajo futuro que podrían aportar valor añadido al TFM realizado. La sección debería señalar las perspectivas de futuro que abre el trabajo desarrollado para el campo de estudio definido. En el fondo, debes justificar de qué modo puede emplearse la aportación que has desarrollado y en qué campos.

# Bibliografía

Aschmann, R. (21 de 05 de 2022). *Chronology of the Bible*. Obtenido de https://aschmann.net/BibleChronology/castellano.html

*Concepto y definicion Net*. (28 de 7 de 2023). Obtenido de https://conceptodefinicion.net/descubre-cuantas-palabras-hay-en-la-biblia/

Davis, K. C. (1998). *Que se yo de la Biblia.* (T. Arijon, Trad.) New York: Sudamericana.

*EL orden Mundial EOM*. (20 de 12 de 2019). Obtenido de El cristianismo en el mundo: https://elordenmundial.com/mapas-y-graficos/cristianismo-en-el-mundo/

*Embajada de Israel en España*. (s.f.). Obtenido de https://embassies.gov.il/madrid/AboutIsrael/AmongtheNations/Pages/ENTRE-NACIONES-Judeidad.aspx

*GENSIM topic modeling for humans*. (Consultada en Nov 22 2023). Obtenido de https://radimrehurek.com/gensim/

*NLTK*. (22 de 11 de Consultada en 2023). Obtenido de https://www.nltk.org/

Ravichandiran, S. (January 2021). *Getting Started with Goggle BERT.* Birmingham, UK: Packt Publishing Ltda.

Sarah Guthals, P. P. (2019). *GitHub dummies A wiley brand.* Hooboken, NJ, USA: John Wiley & Son, inc.

Sowmya Vajjala, B. M. (2020). *Practical Natural language Processing.* Sebastopol, CA, USA: O'Relly Media, inc.

*Textblob: Simplified Text Processing*. (Noviembre de Consultado en 2023). Obtenido de https://textblob.readthedocs.io/en/dev/

*Tranformers Huggin Face*. (Consultado en nov 22 2023). Obtenido de https://huggingface.co/docs/transformers/index