Práctica 4. Identificación de palabras, frases, y documentos similares

Desarrollado por Hernández Jiménez Erick Yael
Escuela Superior de Cómputo
Ingeniería en Inteligencia Artificial

Para la materia de Tecnologías de Lenguaje Natural
Impartida por Ituriel Enrique Flores Estrada
27 de noviembre de 2024

Resumen 1

Introducción 2

Similitud de palabras 2

Synsets 3

Rapid Automatic Keyword Extraction 4

BERT 4

Modelo bert-base-uncased 5

GloVe 6

Propiedades de los embeddings de GloVe 7

Desarrollo 7

Generación de cuerpos de documentos 7

Normalización de textos 8

Similitud de palabras con synsets 9

Similitud de verbos 10

Similitud de sustantivos 12

Similitud de documentos con synsets 15

RAKE – NLTK 15

Path similarity 15

Similitud de palabras con “embedding” 16

GloVe 16

Similitud de documentos con “embedding” 17

BERT – Transformers 17

Conclusiones 18

Bibliografía 19

Extracción automática de texto.

Fórmulas

Fórmula 1. Profundidad del ancestro más cercano. 3
Fórmula 2. Mapeo de palabras en espacio vectorial. 6

Resultados

Resultado 1. Muestra de corpus normalizado. 9
Resultado 2. Muestra de verbos más frecuentes por libro. 10
Resultado 3. Similitud de verbos con método de Wu Palmer y por Path de los verbos más frecuentes. 11
Resultado 4. Muestra de sustantivos más frecuentes por libro. 13
Resultado 5. Similitud de verbos con método de Wu Palmer y por Path de los verbos más frecuentes. 14
Resultado 6. Enunciados principales de los 5 libros con RAKE. 15
Resultado 7. Similitud de frases con método de Path Similarity. 15
Resultado 8. Palabras similares obtenidas con GloVe. 16
Resultado 9. Similitud de documentos encontrados con BERT. 17

Extracción automática de texto.

Resumen

La práctica comprende la creación de un cuerpo de documentos basado en textos del portal Project Gutenberg, la normalización y etiquetado gramatical de estos textos, y el análisis de similitudes semánticas y sintácticas mediante herramientas como *WordNet*, *GloVe* y *BERT*.

Se deben identificar términos y frases similares utilizando métricas como *wup_similarity*, *Path_similarity* y similitud de coseno, así como comparar documentos mediante *embeddings* pre entrenados. Finalmente, los resultados obtenidos deben analizarse y presentarse en un reporte formal que incluya tablas, imágenes y código debidamente documentado, destacando las diferencias entre los métodos empleados.

Palabras clave: Similitud, palabras, frases, documentos, *WordNet*, *GloVe*, *BERT*, *embeddings*, normalización, *wup_similarity*, *Path_similarity*, coseno, análisis, Python.

Extracción automática de texto.

Introducción

La similitud en el procesamiento de lenguaje natural (PLN) es una técnica que mide el grado de relación o semejanza entre palabras, frases o documentos, crucial para tareas como análisis semántico, recuperación de información y traducción automática. En este contexto, las métricas como `wup_similarity` y `Path_similarity`, proporcionadas por la biblioteca NLTK, son ampliamente utilizadas. Ambas se basan en la estructura jerárquica de *WordNet*, un diccionario léxico, donde `wup_similarity` calcula la similitud considerando la profundidad del nodo más bajo compartido entre dos términos, mientras que `Path_similarity` mide la proximidad en la jerarquía mediante la longitud del camino más corto entre dos conceptos.

Por otro lado, los *embeddings* son representaciones numéricas densas de palabras o textos en un espacio vectorial, diseñadas para capturar relaciones semánticas y contextuales. Modelos como *GloVe* y *BERT* generan *embeddings* que permiten comparar palabras y documentos con mayor precisión, utilizando técnicas como la similitud de coseno para medir la alineación entre vectores. Estos métodos combinados enriquecen el análisis semántico y sintáctico en aplicaciones de PLN.

Similitud de palabras

La similitud en el procesamiento de lenguaje natural (PLN) se refiere a la capacidad de medir cuán relacionados o similares son dos elementos lingüísticos, ya sean palabras, frases o documentos. Esta similitud puede evaluarse desde un punto de vista semántico, considerando el significado, o desde un enfoque sintáctico, basado en estructuras o patrones lingüísticos. Es una herramienta esencial en tareas como clasificación de texto, resumen automático y recuperación de información, ya que permite evaluar conexiones entre términos más allá de coincidencias literales.

En el caso de *NLTK*, la similitud semántica entre palabras se implementa mediante métricas basadas en *WordNet*, una base de datos léxica jerárquica que organiza palabras según su significado en categorías llamadas *synsets*. Los métodos más destacados son:

1. **Wu-Palmer Similarity (wup_similarity)**: Este método mide la similitud entre dos palabras calculando la profundidad del ancestro común más cercano (*Lowest Common Subsumer*, LCS) en la jerarquía de *WordNet* en relación con las profundidades individuales de las palabras evaluadas. La fórmula generalizada es:

$$\mathrm{Sim}(w1, w2) = \frac{2 \times \mathrm{Depth}(LCS)}{\mathrm{Depth}(w1) + \mathrm{Depth}(w2)}$$

Fórmula 1. Profundidad del ancestro más cercano.

Esto significa que cuanto más cercano esté el LCS al nodo raíz (más general es), menor será la similitud, mientras que un LCS más profundo indica una relación más específica y, por tanto, mayor similitud.

2. **Path Similarity (path_similarity)**: Este método calcula la similitud en función de la longitud del camino más corto entre dos palabras en la jerarquía de *WordNet*. El valor es inversamente proporcional a la cantidad de pasos necesarios para conectar dos *synsets*; es decir, un camino más corto implica una similitud mayor. Su simplicidad lo hace eficiente, aunque puede ser menos preciso en casos donde la estructura jerárquica no captura perfectamente la relación semántica.

Ambos métodos permiten cuantificar relaciones semánticas entre términos, proporcionando una base sólida para tareas de análisis más complejas, especialmente cuando se combinan con enfoques modernos como los *embeddings*.

Synsets

Los **synsets** (conjuntos sinónimos) en NLTK son unidades básicas de *WordNet* que agrupan palabras con significados similares. Cada *synset* representa un concepto único o un sentido específico de una palabra, y está conectado jerárquicamente a otros *synsets* mediante relaciones semánticas, como hipónimos (más específicos), hipónimos (más generales) y merónimos (partes de un todo). Por ejemplo, los *synsets* para "*bank*" incluirían uno para "banco de un río" y otro para "institución financiera" [1].

En los métodos *Wu-Palmer* (`wup_similarity`) y *Path Similarity* (`path_similarity`), los *synsets* se usan para medir similitudes semánticas entre términos:

- ***Wu-Palmer*** calcula la similitud basada en la profundidad del ancestro común más cercano (Lowest Common Subsumer, LCS) en la jerarquía y las profundidades individuales de los synsets evaluados.
- ***Path Similarity*** mide la similitud considerando la longitud del camino más corto entre los synsets en la estructura jerárquica.

Ambos enfoques utilizan los *synsets* para aprovechar las relaciones semánticas y estructurales en *WordNet*, permitiendo una comparación más rica y precisa de palabras.

Rapid Automatic Keyword Extraction

RAKE (*Rapid Automatic Keyword Extraction*) es un algoritmo no supervisado para extraer palabras clave de un texto. Funciona al identificar frases que no contienen palabras comunes ("stop words"), asignar puntuaciones a las palabras basadas en su frecuencia y coocurrencia, y luego calcular una puntuación total para las frases, permitiendo así extraer las palabras clave más importantes.

NLTK ofrece una implementación de RAKE a través de "*rake-nltk"*, que permite tokenizar textos y asignar puntuaciones automáticamente a las palabras clave mediante técnicas similares, simplificando el proceso de análisis. [1]

BERT

BERT (*Bidirectional Encoder Representations from Transformers*) es un modelo basado en transformers que comprende el contexto bidireccional de las palabras, lo que lo hace poderoso para análisis contextual y resumen de textos. Funciona al generar representaciones ricas de frases para tareas de procesamiento de lenguaje natural [2].

Por otro lado, *BART* (*Bidirectional and Auto-Regressive Transformer*), en su implementación, utiliza *BERT* para codificar texto y genera un resumen al estilo de un decodificador autorregresivo, logrando resúmenes extractivos y abstracticos. *BART*

combina el aprendizaje bidireccional con técnicas de generación, mejorando la precisión en resúmenes [3].

BERT no solo se destaca por comprender el contexto bidireccional de las palabras en una oración, sino también por su capacidad de generar *embeddings* contextuales. Estos *embeddings* son representaciones numéricas densas de palabras, frases o documentos en un espacio vectorial, donde términos con significados similares se agrupan cercanamente. A diferencia de métodos tradicionales como Word2Vec o *GloVe*, que generan representaciones estáticas, los *embeddings* de BERT son dinámicos y dependientes del contexto, lo que significa que una misma palabra puede tener diferentes vectores dependiendo de su uso en la oración.

Para el análisis de similitud, BERT convierte las frases o documentos en embeddings mediante su arquitectura de transformadores. Una vez generados, las relaciones semánticas entre textos se evalúan mediante métricas como la similitud de coseno, que mide la alineación entre vectores en el espacio. Este enfoque permite captar matices complejos, como sinonimia, polisemia y variaciones contextuales, ofreciendo resultados superiores en comparación con técnicas más simples.

Modelo bert-base-uncased

El modelo pre entrenado `bert-base-uncased` es una variante de BERT con las siguientes características clave:

Arquitectura: Este modelo tiene 12 capas de transformadores (*encoders*), 768 dimensiones en cada capa y un total de 110 millones de parámetros. Esto lo hace adecuado para tareas generales de PLN, equilibrando precisión y eficiencia computacional.

Uncased: El modelo ignora las diferencias entre mayúsculas y minúsculas, tratándolas como equivalentes (por ejemplo, "Dog" y "dog" tendrán la misma representación).

Preentrenamiento: Fue entrenado con dos objetivos principales:

Extracción automática de texto.

- Máscara de lenguaje (Masked Language Modeling, MLM), donde se ocultan palabras en una oración y el modelo predice el término más probable basado en el contexto bidireccional.
- Predicción de la siguiente oración (Next Sentence Prediction, NSP), que permite al modelo comprender la relación entre oraciones consecutivas.

El modelo `bert-base-uncased` es ampliamente utilizado en tareas de similitud textual porque puede manejar frases y documentos con estructuras complejas y matices contextuales. Combinado con métricas como la similitud de coseno, facilita la comparación precisa entre textos, superando las limitaciones de métodos basados en *embeddings* estáticos [4].

GloVe

GloVe (*Global Vectors for Word Representation*) es un modelo de aprendizaje para generar representaciones vectoriales de palabras que captura relaciones semánticas y estadísticas entre términos. A diferencia de modelos como *Word2Vec*, que se basan en contextos locales dentro de una ventana de palabras, *GloVe* utiliza una combinación de información global del corpus completo y coocurrencias locales, logrando *embeddings* que reflejan patrones semánticos y sintácticos.

El enfoque de *GloVe* se basa en la matriz de coocurrencia: una representación donde cada entrada indica cuántas veces una palabra aparece en el contexto de otra dentro de un corpus. A partir de esta matriz, *GloVe* aprende a mapear palabras en un espacio vectorial, resolviendo la ecuación [4]:

$$w_i^T \cdot w_j + b_i + b_j = log(X_{ij})$$

Fórmula 2. Mapeo de palabras en espacio vectorial.

donde w_i y w_j son los vectores de las palabras i y j, b_i y b_j son los sesgos, y X_{ij} representa la frecuencia de coocurrencia entre las palabras i y j. Este diseño optimiza el modelo para que la relación entre los vectores resultantes preserve propiedades semánticas, como analogías.

Extracción automática de texto.

Propiedades de los embeddings de GloVe

En el espacio vectorial generado, las palabras con significados similares tienen vectores cercanos (según métricas como la similitud de coseno). Por ejemplo, "*king*" y "*queen*" estarán más cerca que "*king*" y "*car*".

Además, una característica distintiva de *GloVe* es su capacidad para capturar relaciones semánticas lineales. Por ejemplo, las relaciones de género o pluralidad pueden representarse como operaciones vectoriales:

$$\mathrm{Vec}(\mathrm{king}) - \mathrm{Vec}(\mathrm{man}) + \mathrm{Vec}(\mathrm{woman}) \approx \mathrm{Vec}(\mathrm{queen})$$

Ecuación 1. Ejemplo de operación de similitud entre palabras.

Así mismo, *GloVe* es eficiente en entrenamiento, ya que se basa en factorizaciones matriciales pre computadas de la matriz de coocurrencia. Esto lo hace adecuado para grandes corpus como *Wikipedia* o *Common Crawl* [5].

Desarrollo

A continuación, se describe a grandes rasgos el proceso que se llevó a cabo para el desarrollo de la práctica. Para más detalles de la implementación (funciones, métodos y lógica usada), consúltese el archivo de código en el cuaderno de Jupyter. Los requerimientos para construir el entorno virtual recomendado se encuentran en el archivo "requirements.txt"

Generación de cuerpos de documentos

Como parte de las indicaciones, se guardaron los primeros capítulos o introducciones de 5 libros de la plataforma de Gutenberg en archivos de texto con los nombres:

- "A_JOURNEY_TO_THE_CENTRE_OF_THE_EARTH_CHAPTER_1.txt"
- "ALL_AROUND_THE_MOON_CHAPTER_1.txt"
- "OFF_ON_A_COMET_CHAPTER_1.txt"
- "THE_MYSTERIOUS_ISLAND_CHAPTER_1.txt"
- "TWENTY_THOUSAND_LEAGUES_UNDER_THE_SEA_CHAPTER_1.txt"

Extracción automática de texto.

Estos se guardaron en las variables homónimas y se tokenizarán en enunciados para facilitar la normalización siguiente y, así, evitar problemas en la definición de las oraciones.

Esta tokenización se hizo con el método '*sent_tokenize*' de '*NLTK*'. Finalmente, se juntan los tokens en una misma cadena de texto '*corpus*'. Esto facilita la manipulación y separación de los tokens tanto por palabras como por enunciados.

Normalización de textos

Dado que se solicita que:

1. Se tokenicen por enunciado
2. Se tokenicen por palabras
3. Se les etiquete por su papel en el enunciado (POS)

Lo único que se le agregará a este flujo será:

1. Conversión a minúsculas: para que sean compatibles con los métodos de GloVe y BERT, así como para evitar redundancias.
2. Eliminado de números y caracteres especiales: ya que no aportan significado al contenido.
3. Lematización: para que el análisis de similitud se enfoque en el significado del lema y no de la palabra en sí para los métodos de Wu Palmer o por *Paths*.

Así, en conjunto, la normalización es:

1. Tokenización por enunciados
2. Tokenización y etiquetado por palabras
3. Conversión a minúsculas
4. Filtrado de caracteres especiales
5. Lematización

Con este flujo de normalización simple evitamos redundar en palabras y en aplicación de métodos para enfocarnos en el contenido de las palabras y su contexto en el caso de resumidores más complejos como el que se realiza con *BERT* o *GloVe*.

Una muestra de los resultados de este proceso es el siguiente texto:

[('uncle', 'n'), ('make', 'n'), ('great', 'n'), ('discovery', 'n'), ('looking', 'n'), ('back', 'r'), ('occur', 'v'), ('since', 'n'), ('eventful', 'a'), ('day', 'n'), ('scarcely', 'r'), ('able', 'a'), ('believe', 'v'), ('reality', 'n'), ('adventure', 'n'), ('truly', 'r'), ('wonderful', 'a'), ('even', 'r'), ('bewilder', 'v'), ('think', 'v'), ('uncle', 'n'), ('german', 'a'), ('marry', 'v'), ('mother', 'n'), ('sister', 'n'), ('englishwoman', 'n'), ('much', 'r'), ('attach', 'v'), ('fatherless', 'n'), ('nephew', 'r'), ('invite', 'v'), ('study', 'v'), ('home', 'n'), ('fatherland', 'n'), ('home', 'n'), ('large', 'a'), ('town', 'n'), ('uncle', 'n'), ('professor', 'n'), ('philosophy', 'n'), ('chemistry', 'n'), ('geology', 'n'), ('mineralogy', 'n'), ('many', 'a'), ('ology', 'n'), ('one', 'n'), ('day', 'n'), ('pass', 'v'), ('hour', 'n'), ('uncle', 'n'), ('absent', 'n'), ('suddenly', 'r'), ('felt', 'v'), ('necessity', 'n'), ('renovate', 'v'), ('hungry', 'a'), ('rouse', 'v'), ('old', 'a'), ('french', 'a'), ('cook', 'n'), ('uncle', 'n'), ('professor', 'n'), ('von', 'n'), ('hardwigg', 'n'), ('suddenly', 'r'), ('open', 'v'), ('street', 'n'), ('door', 'n'), ('come', 'v'), ('rush', 'v'), ('upstairs', 'n'), ('professor', 'n'), ('hardwigg', 'n'), ('worthy', 'a'), ('uncle', 'n'), ('mean', 'v'), ('bad', 'a'), ('sort', 'n'), ('man', 'n'), ('however', 'r'), ('choleric', 'a'), ('original', 'a'), ('bear', 'v'), ('mean', 'v'), ('obey', 'v'), ('scarcely', 'r'), ('heavy', 'a'), ('foot', 'n'), ('resound', 'v'), ('within', 'n'), ('joint', 'n'), ('domicile', 'n'),…]

Resultado 1. Muestra de corpus normalizado.

Similitud de palabras con synsets

En el programa, una vez que hemos etiquetado las palabas por verbos (‘v’), sustantivos (‘n’) y adjetivos (‘a’) y su lema.

1. Por cada libro crearemos un diccionario con los verbos y las frecuencias de cada uno.
2. Seleccionamos el verbo más común por ser el más frecuente
3. Buscamos los 5 verbos más comunes en el libro correspondiente con "wup similarity"
4. Buscamos los 5 verbos más comunes en el libro correspondiente con "path similarity".

Así mismo...:

1. Por cada libro crearemos un diccionario con los sustantivos y las frecuencias de cada uno.
2. Seleccionamos el sustantivo más común por ser el más frecuente
3. Buscamos los 5 sustantivos más comunes en el libro correspondiente con "wup similarity"
4. Buscamos los 5 sustantivos más comunes en el libro correspondiente con "path similarity"

Extracción automática de texto.

Similitud de verbos

Obteniendo los verbos ordenados por su frecuencia descendentemente, se obtienen las siguientes muestras por cada libro:

Libro 1	{'say': 7, 'know': 7, 'make': 7, 'come': 5, 'find': 5, 'mean': 4, 'cry': 4, 'declare': 4, 'tell': 3, 'learn': 3, 'give': 3, 'take': 3, 'grow': 3, 'think': 2, 'study': 2, 'open': 2, 'obey': 2, 'wait': 2, 'consume': 2, 'pore': 2, 'stammer': 2, 'resemble': 2, 'hop': 2, 'see': 2, 'go': 2, 'relate': 2, 'love': 2, 'appear': 2, 'write': 2, 'fell': 2, 'occur': 1, …}
Libro 2	{'throw': 7, 'appear': 7, 'rise': 6, 'take': 6, 'reach': 6, 'pass': 5, 'extend': 4, 'contain': 4, 'lighten': 4, 'fall': 3, 'hear': 3, 'hold': 3, 'see': 3, 'know': 3, 'save': 3, 'drive': 3, 'descend': 2, 'resound': 2, 'rag': 2, 'cover': 2, 'devastate': 2, 'cast': 2, 'leave': 2, 'carry': 2, 'traverse': 2, 'come': 2, 'reckon': 2, 'suspend': 2, 'sustain': 2, 'begin': 2, 'say': 2, 'diminish': 2, 'go': 2, 'exhaust': 2, 'collapse': 2, 'maintain': 2, 'hoist': 2, 'try': 2, 'prevent': 2, 'fell': 2, 'escape': 2, 'retain': 2, 'retard': 2, 'remain': 2, 'plunge': 2, 'mount': 2, 'make': 2, 'strike': 2, 'let': 2, 'sake': 1, …}
Libro 3	{'go': 5, 'make': 4, 'hundred': 4, 'find': 4, 'excite': 3, 'take': 3, 'admit': 3, 'become': 3, 'break': 3, 'strike': 3, 'forget': 2, 'agitate': 2, 'meet': 2, 'enter': 2, 'seem': 2, 'surpass': 2, 'exist': 2, 'produce': 2, 'think': 2, 'observe': 2, 'sail': 2, 'frequent': 2, 'grave': 2, 'circulate': 2, 'hurry': 2, 'revive': 2, 'give': 2, 'bring': 2, 'continue': 2, 'sink': 2, 'stop': 2, 'put': 2, 'signalise': 1, …}
Libro 4	{'take': 5, 'say': 5, 'make': 5, 'arrange': 3, 'reply': 3, 'love': 3, 'induce': 2, 'allow': 2, 'leave': 2, 'appoint': 2, 'meet': 2, 'part': 2, 'assign': 2, 'answer': 2, 'start': 2, 'see': 2, 'find': 2, 'keep': 2, 'know': 2, 'call': 2, 'marry': 2, 'want': 2, 'whoe': 2, 'surrender': 1, …}
Libro 5	{'take': 4, 'fasten': 2, 'look': 2, 'strike': 1, 'begin': 1, 'surround': 1, 'intend': 1, 'accompany': 1, 'deposit': 1, 'approach': 1, 'seat': 1, 'creak': 1, 'start': 1, 'dare': 1, 'disappear': 1, 'give': 1, 'come': 1, …}

Resultado 2. Muestra de verbos más frecuentes por libro.

Tomando el primer verbo de cada lista, los verbos más similares a cada uno son:

Verbo	Wu Palmer Similarity	Path Similarity
say	(**'love'**, 0.5333333333333333) (**'wish'**, 0.5333333333333333) (**'delight'**, 0.5333333333333333) (**'surprise'**, 0.5333333333333333) ('want', 0.5)	(**'love'**, 0.125) ('present', 0.125) (**'wish'**, 0.125) (**'delight'**, 0.125) (**'surprise'**, 0.125)
throw	(**'drive'**, 0.8571428571428571) (**'try'**, 0.7142857142857143) (**'escape'**, 0.7142857142857143) (**'support'**, 0.7142857142857143) (**'pass'**, 0.6666666666666666)	(**'drive'**, 0.3333333333333333) (**'try'**, 0.2) (**'escape'**, 0.2) (**'support'**, 0.2) (**'pass'**, 0.16666666666666666)
go	('grave', 0.375) ('hundred', 0.35294117647058826) ('thousand', 0.35294117647058826) (**'signal'**, 0.3076923076923077) ('set', 0.2857142857142857)	(**'signal'**, 0.1) ('forget', 0.09090909090909091) ('enter', 0.09090909090909091) ('surpass', 0.09090909090909091) ('exist', 0.09090909090909091)
take	(**'keep'**, 0.5882352941176471)	(**'keep'**, 0.125)

	(**'part'**, 0.42857142857142855) (**'form'**, 0.35294117647058826) ('come', 0.3333333333333333) ('hide', 0.3)	(**'part'**, 0.1111111111111111) ('allow', 0.08333333333333333) ('forego', 0.08333333333333333) (**'form'**, 0.08333333333333333)
take	(**'keep'**, 0.5882352941176471) (**'lease'**, 0.5) ('come', 0.3333333333333333) ('remove', 0.3333333333333333) ('pad', 0.3333333333333333)	(**'keep'**, 0.125) (**'lease'**, 0.1111111111111111) ('intend', 0.08333333333333333) ('accompany', 0.08333333333333333) ('disappear', 0.08333333333333333)

Resultado 3. Similitud de verbos con método de Wu Palmer y por Path de los verbos más frecuentes.

Estos resultados, en general, pese a ser con distintos métodos tienen verbos en común, indicando una fuerte relación semántica con el resto de los verbos en la lista obtenida en el proceso. En promedio cada verbo principal tiene 3 verbos con los que mantiene similitud en ambos métodos. La diferencia radica en cómo cada método busca la palabra más similar tal como se explicó en el marco teórico presente.

Particularmente, en el caso del tercer libro, solo hubo un verbo en común similar al principal. Esto se debió en gran parte a un proceso de etiquetado con poca precisión que, en este punto, afectó a la categorización de las palabras adecuadamente. A esto se llega debido a que el resto de las palabras no son verbos sino sustantivos para los pseudo verbos encontrados con el método de *Wu Palmer*. Mientras que con el método con *Paths* es menos evidente esta deficiencia en el etiquetado, puesto que realmente se encuentran verbos, pero estos mismos tiene muy poca relación (igual o por debajo de 10% de similitud) y la única coincidencia en ambos métodos no es verdaderamente un verbo.

Esto se podría evitar si se fuese más minucioso con el proceso de etiquetado o con un flujo de normalización distinto que no deteriore la palabra antes del etiquetado correspondiente.

Con las anotaciones anteriores, ambos métodos tienen una coincidencia del 60% (15/25 resultados) y con una similitud media de 48.3% y 12.64% con el método de *Wu Palmer* y con el de *Paths* respectivamente.

La coincidencia del 60% indica que ambos métodos llegaron a los mismos resultados en casi la mitad de los casos y esto, nuevamente, puede ser ocasionado por cómo están etiquetadas las palabras y cómo fueron manejadas como lemas y no como las palabras originales. Mientras que la similitud encontrada media es consecuencia de los “verbos” que se encontraron en el proceso de selección y probablemente debido al etiquetado por el que pasaron.

Similitud de sustantivos

Obteniendo los sustantivos ordenados por su frecuencia descendentemente, se obtienen las siguientes muestras por cada libro:

Libro 1	{'uncle': 33, 'professor': 8, 'one': 8, 'hardwigg': 6, 'time': 6, 'house': 6, 'man': 5, 'dinner': 5, 'mineralogy': 4, 'cook': 4, 'upon': 4, 'like': 4, 'event': 4, 'day': 3, 'adventure': 3, 'geology': 3, 'three': 3, 'truth': 3, 'question': 3, 'science': 3, 'would': 3, 'glass': 3, 'interest': 3, 'u': 3, 'letter': 3, 'subject': 3, 'nothing': 3, 'every': 3, 'icelandic': 3, 'language': 3, 'home': 2, 'town': 2, 'street': 2, 'door': 2, 'foot': 2, 'could': 2, 'room': 2, 'tone': 2, 'moment': 2, 'omelette': 2, 'value': 2, 'benefit': 2, 'order': 2, 'knowledge': 2, 'word': 2, 'connection': 2, 'fond': 2, 'fact': 2, 'study': 2, 'specimen': 2, 'meal': 2, 'men': 2, 'nose': 2, 'article': 2, 'presence': 2, 'half': 2, 'notion': 2, 'part': 2, 'four': 2, 'leaf': 2, 'account': 2, 'hand': 2, 'book': 2, 'work': 2, 'iceland': 2, 'idiom': 2, 'parchment': 2, 'piece': 2, 'runic': 2, 'matter': 2, 'two': 2, 'thousand': 2, 'make': 1, 'great': 1, …}
Libro 2	{'balloon': 29, 'could': 16, 'land': 14, 'sea': 12, 'car': 11, 'foot': 10, 'voyager': 9, 'air': 8, 'hour': 8, 'wave': 7, 'mile': 7, 'wind': 7, 'gas': 6, 'like': 5, 'four': 5, 'storm': 5, 'without': 5, 'might': 5, 'beneath': 5, 'passenger': 5, 'ocean': 5, 'article': 5, 'time': 5, 'must': 5, 'men': 5, 'u': 4, 'weight': 4, 'everything': 4, 'one': 4, 'fact': 4, 'round': 4, 'five': 4, 'surface': 4, 'part': 4, 'day': 4, 'two': 4, 'position': 4, 'whether': 4, 'half': 4, 'whose': 4, 'voice': 4, 'rise': 3, 'march': 3, 'distance': 3, 'water': 3, 'tempest': 3, 'movement': 3, 'rate': 3, 'midst': 3, 'atmosphere': 3, 'night': 3, 'danger': 3, 'height': 3, 'would': 3, 'region': 3, 'towards': 3, 'case': 3, 'direction': 3, 'mesh': 3, 'watery': 2, 'clock': 2, 'northeast': 2, 'parallel': 2, 'coast': 2, 'vessel': 2, 'shore': 2, 'hundred': 2, 'waterspout': 2, 'catastrophe': 2, 'space': 2, 'point': 2, 'vapor': 2, 'hung': 2, 'world': 2, 'symptom': 2, 'since': 2, 'although': 2, 'descent': 2, 'provision': 2, 'layer': 2, 'fluid': 2, 'abyss': 2, 'death': 2, 'hurricane': 2, 'breeze': 2, 'element': 2, 'continent': 2, 'island': 2, 'mercy': 2, 'crest': 2, 'cost': 2, 'effort': 2, 'southwest': 2, 'master': 2, 'possibility': 2, 'rent': 2, 'minute': 2, 'fall': 2, 'dog': 2, 'bird': 2, 'captain': 1, …}
Libro 3	{'could': 10, 'monster': 10, 'three': 9, 'two': 8, 'sea': 8, 'company': 7, 'one': 6, 'fact': 6, 'length': 6, 'ship': 6, 'scotia': 6, 'captain': 5, 'question': 5, 'day': 5, 'journal': 5, 'vessel': 4, 'time': 4, 'object': 4, 'creature': 4, 'foot': 4, 'opinion': 4, 'mile': 4, 'place': 4, 'shock': 4, 'reef': 3, 'year': 3, 'mind': 3, 'continent': 3, 'officer': 3, 'country': 3, 'whale': 3, 'science': 3,

Libro 4	'observation': 3, 'governor': 3, 'higginson': 3, 'steam': 3, 'five': 3, 'water': 3, 'ocean': 3, 'lat': 3, 'part': 3, 'yard': 3, 'board': 3, 'whose': 3, 'existence': 3, 'rock': 3, 'moravian': 3, 'passenger': 3, 'thanks': 3, 'anderson': 3, 'compartment': 3, 'phenomenon': 2, 'men': 2, 'sailor': 2, 'europe': 2, 'america': 2, 'state': 2, 'matter': 2, 'past': 2, 'movement': 2, 'apparition': 2, 'shape': 2, 'rapidity': 2, 'cetacean': 2, 'might': 2, 'dimension': 2, 'july': 2, 'navigation': 2, 'sandbank': 2, 'column': 2, 'air': 2, 'mammal': 2, 'pacific': 2, 'columbus': 2, 'another': 2, 'chart': 2, 'seven': 2, 'helvetia': 2, 'shannon': 2, 'royal': 2, 'island': 2, 'report': 2, 'line': 2, 'staff': 2, 'upon': 2, 'every': 2, 'paper': 2, 'moby': 2, 'dick': 2, 'month': 2, 'article': 2, 'institution': 2, 'nature': 2, 'like': 2, 'public': 2, 'danger': 2, 'quarter': 2, 'four': 2, 'knot': 2, 'strength': 2, 'hull': 2, 'would': 2, 'bottom': 2, 'circumstance': 2, 'half': 2, 'minute': 2, 'leak': 2, 'iron': 2, 'plate': 2, 'shifting': 1, …}
Libro 5	{'captain': 13, 'servadac': 13, 'count': 10, 'two': 10, 'mostaganem': 7, 'would': 6, 'timascheff': 5, 'way': 5, 'shelif': 5, 'one': 4, 'officer': 4, 'clock': 4, 'coast': 4, 'town': 4, 'might': 4, 'man': 4, 'sword': 3, 'second': 3, 'cause': 3, 'place': 3, 'upon': 3, 'mile': 3, 'quarter': 3, 'friend': 3, 'gourbi': 3, 'lover': 3, 'nothing': 2, 'matter': 2, 'seniority': 2, 'affair': 2, 'kind': 2, 'name': 2, 'hector': 2, 'staff': 2, 'wassili': 2, 'every': 2, 'case': 2, 'shall': 2, 'dispute': 2, 'wagner': 2, 'rossini': 2, 'smile': 2, 'another': 2, 'mouth': 2, 'month': 2, 'continent': 2, 'towards': 2, 'hour': 2, 'five': 2, 'entrance': 2, 'thousand': 2, 'harbor': 2, 'cliff': 2, 'part': 2, 'concession': 2, 'word': 2, 'four': 2, 'effort': 2, 'rondo': 2, 'composition': 2, 'poetry': 2, 'rhyme': 2, 'simplicity': 2, 'er': 2, 'love': 2, 'verse': 2, 'chapter': 1, …}
	{'one': 5, 'ardan': 3, 'projectile': 3, 'car': 3, 'may': 3, 'stony': 2, 'barbican': 2, 'three': 2, 'mouth': 2, 'time': 2, 'windlass': 2, 'interior': 2, 'light': 2, 'prison': 2, 'moment': 1, …}

Resultado 4. Muestra de sustantivos más frecuentes por libro.

Tomando el primer verbo de cada lista, los verbos más similares a cada uno son:

Sustantivo	Wu Palmer Similarity	Path Similarity
uncle	(**'person'**, 0.7058823529411765) (**'relative'**, 0.6666666666666666) (**'inhabitant'**, 0.6666666666666666) (**'polyglot'**, 0.6666666666666666) (**'man'**, 0.631578947368421)	(**'relative'**, 0.3333333333333333) (**'person'**, 0.25) (**'inhabitant'**, 0.2) (**'polyglot'**, 0.2) (**'man'**, 0.16666666666666666)
could	(**'ship'**, 0.782608695652174) (**'car'**, 0.6666666666666666) ('ammunition', 0.6) ('machine', 0.6) ('basket', 0.6)	(**'ship'**, 0.16666666666666666) (**'car'**, 0.1111111111111111) ('surface', 0.1111111111111111) ('layer', 0.1111111111111111) ('plaything', 0.1111111111111111)
captain	No hay sinsets para el sustantivo 'could'	No hay sinsets para el sustantivo 'could'

captain	(**'officer'**, 0.88) (**'orderly'**, 0.7407407407407407) (**'friend'**, 0.5454545454545454) (**'lover'**, 0.5454545454545454) (**'disputant'**, 0.5454545454545454)	(**'officer'**, 0.25) (**'orderly'**, 0.125) (**'friend'**, 0.1111111111111111) (**'lover'**, 0.1111111111111111) (**'disputant'**, 0.1111111111111111)
one	(**'three'**, 0.875) (**'two'**, 0.875) (**'hundred'**, 0.75) (**'moment'**, 0.46153846153846156) ('week', 0.42857142857142855)	(**'three'**, 0.3333333333333333) (**'two'**, 0.3333333333333333) (**'hundred'**, 0.2) (**'moment'**, 0.125) ('space', 0.1111111111111111)

Resultado 5. Similitud de verbos con método de Wu Palmer y por Path de los verbos más frecuentes.

Estos resultados, en promedio muestran 4 sustantivos en común con los que mantiene similitud en ambos métodos. La diferencia radica, igualmente, en cómo cada método busca la palabra más similar tal como se explicó en el marco teórico presente.

Particularmente, en el caso del tercer libro, no se encontraron synsets para el pseudo sustantivo ‘could’ ya que este no es un estema y, adicionalmente no es un sustantivo en sí. Esto se debió igualmente a un proceso de etiquetado con poca precisión que, en este punto, afectó a la categorización de las palabras adecuadamente.

Con las anotaciones anteriores, ambos métodos tienen una coincidencia del 80% (16/20 de los resultados encontrados, sin contar los resultados del tercer libro) y con una similitud media de 66.24% y 17.86% con el método de *Wu Palmer* y con el de *Paths* respectivamente.

La coincidencia del 80% indica que ambos métodos llegaron a los mismos resultados en más de la mitad de los casos y esto, nuevamente, puede ser ocasionado por cómo están etiquetadas las palabras y cómo fueron manejadas como lemas y no como las palabras originales. Mientras que la similitud encontrada media es consecuencia de los “sustantivos” que se encontraron en el proceso de selección y probablemente debido al etiquetado por el que pasaron.

Extracción automática de texto.

En general, tanto para los verbos como para los sustantivos, el etiquetado jugó un papel fundamental para encontrar los synsets adecuados y, en consecuencia, la similitud entre las palabras de cada libro.

Similitud de documentos con synsets

En la práctica 3 se encontraron 3 métodos con buenos resultados: Text Rank, *Rake* y Frecuencia de Palabras Normalizada. Se usará *Rake* ya que solo se requiere unir los tokens de cada libro, simplificando el proceso.

RAKE – NLTK

La implementación de este método fue hecha con la guía del contenido de Manmohan Singh [4] aplicada en la práctica 3.

1. Se importa el algoritmo Rake
2. Se guarda en una variable
3. Se aplica el algoritmo sobre el texto original
4. Se obtiene el resumen

Para ello se importa el constructor '*Rake*' de '*nltk*'. Dado que el método ya extrae las palabras clave de los enunciados, lo alimentamos con la unión de las cadenas con los enunciados que conforman el corpus y, posteriormente, obtenemos los enunciados de los 5 libros. Los resultados son:

classify six hundred different geological specimens
several thousand people crushed
immense kraken whose tentacles could entangle
two officers listened gravely enough
car soon came back empty

Resultado 6. Enunciados principales de los 5 libros con RAKE.

Path similarity

Con las frases obtenidas, se escogió deliberadamente el primer enunciado en la lista como referencia y se realiza el cálculo de similitud:

classify six hundred different geological specimens	
Enunciado	**Similitud**
several thousand people crushed	0.17
immense kraken whose tentacles could entangle	0.12
two officers listened gravely enough	0.14
car soon came back empty	0.12

Resultado 7. Similitud de frases con método de Path Similarity.

Con estos resultados, vemos que concuerda la similitud dado que las palabras encontradas en cada enunciado son, en su mayoría, distintos entre sí, teniendo poca similitud perceptible.

Similitud de palabras con “embedding”

GloVe

Tal como se indicó en las instrucciones, se usó el modelo GloVe para encontrar la similitud entre los verbos de cada libro. Los resultados obtenidos fueron los siguientes:

Verbo	Palabra encontrada	Similitud
say	believe	0.8163
	know	0.7665
	want	0.7577
	think	0.7299
	might	0.7227
throw	ball	0.5623
	catch	0.5044
	let	0.4890
	go	0.4826
	get	0.4729
go	come	0.7988
	do	0.7331
	take	0.7206
	would	0.6825
	think	0.6638
take	give	0.7691
	would	0.7523
	make	0.7344
	come	0.7306
	go	0.7206
take	give	0.7691
	would	0.7523
	come	0.7306
	could	0.7007
	turn	0.6617

Resultado 8. Palabras similares obtenidas con GloVe.

En comparación con los obtenidos en la sección anterior, los resultados son notoriamente mejores, aunque esto se debe tanto a que se considera cada palabra como un vector más robusto que proporciona el modelo de 300 dimensiones, como a

la mayor flexibilidad que se le dio al modelo para encontrar similitudes no solo con los verbos sino con los sustantivos y los adjetivos también.

Pese a lo mencionado, vemos que un modelo con un mapeo más robusto y detallado proporciona mejores resultados aún cuando se siguió un proceso similar de etiquetado.

Similitud de documentos con "embedding"

Para este método, se siguió lo indicado usando el modelo `bert-base-uncased`.

BERT – Transformers

Igualmente, al aplicarle el modelo a las frases principales obtenidas en la sección análoga con `path_similarity`, se obtuvo lo siguiente.

classify six hundred different geological specimens	
Enunciado	**Similitud**
several thousand people crushed	0.5981
immense kraken whose tentacles could entangle	0.6351
two officers listened gravely enough	0.5409
car soon came back empty	0.5192

Resultado 9. Similitud de documentos encontrados con BERT.

Con *BERT*, así como se notó una mejoría notable entre el método con *Paths* con el modelo *GloVe*, se nota también una diferencia en las similitudes encontradas, siendo que la similitud encontrada con *BERT* fue mayor que con el método de *Paths*.

En ambos casos, la diferencia en los resultados radicaría en que, en las primeras aplicaciones, la similitud era semántica (evalúa qué tan cercanos son dos elementos en cuanto a su significado o contenido conceptual) y en las últimas dos aplicaciones se logra un análisis de similitud sintáctica (mide qué tan similares son dos elementos en términos de su estructura superficial o forma escrita).

Extracción automática de texto.

Conclusiones

Con los resultados anteriores, lo primero a notar es la importancia en el proceso de etiquetado de los tokens por su papel gramatical en el enunciado siendo que, entre más minucioso y exacto sea, mejores resultados se obtendrán. Esta relación directa también se mantiene para el proceso de normalización, tal como en prácticas anteriores se ha demostrado. Estos dos factores influyen demasiado en la efectividad del análisis de similitud tanto sintáctico como semántico de las palabras y de los documentos en los corpus.

Aunado a lo anterior, podemos ver que, para el análisis semántico y tomando en cuenta las similitudes obtenidas, el análisis con el método de Wu Palmer otorga mejores resultados que con *Paths* para la similitud de palabras, pero *Paths* ofrece aun así resultados razonables como se vio para el análisis de documentos.

Por otro lado, para el análisis de similitud sintáctico, los *embeddings* ofrecen resultados notoriamente mejores, aunque es entendible debido a la diferencia de condiciones bajo los que fueron aplicados en comparación de la aplicación de los métodos de Wu Palmer y Paths. En contraste, el proceso con *embeddings* es más lento y requiere de recursos más robustos.

En conclusión, realizar un análisis de similitud sintáctico es más rápido de aplicar pero requiere de un pre procesamiento más especializado y minucioso, mientras que para realizar un análisis de similitud semántico es más lento y requiere más recursos pero que, particularmente con Python y sus bibliotecas desarrolladas, en la implementación es más fácil de realizar.

Bibliografía

[1] NLTK, «Sample usage for wordnet,» NLTK, 2024. [En línea]. Available: https://www.nltk.org/howto/wordnet.html. [Último acceso: 26 11 2024].

[2] V. Sharma, «rake-nltk 1.0.6,» pypi, 15 09 2021. [En línea]. Available: https://pypi.org/project/rake-nltk/. [Último acceso: 07 11 2024].

[3] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019.

[4] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov y L. Zettlemoyer, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, 2019.

[5] Hugging Face, «google-bert/bert-base-uncased,» HuggingFace, [En línea]. Available: https://huggingface.co/google-bert/bert-base-uncased. [Último acceso: 26 11 2024].

[6] S. Nivedh, «Introduction to GloVe Embeddings,» Medium, 28 08 2022. [En línea]. Available: https://sainivedh.medium.com/introduction-to-glove-embeddings-9f57d48d0ce4. [Último acceso: 25 11 2024].

[7] M. Singh, «Keyword Extraction process in Python with Natural Language Processing(NLP),» Medium, 03 02 2021. [En línea]. Available: https://towardsdatascience.com/keyword-extraction-process-in-python-with-natural-language-processing-nlp-d769a9069d5c. [Último acceso: 08 11 2024].