ISSN 2007-9737

Métricas de similitud para la desambiguación automática del sentido de los verbos sobre un corpus basado en WordNet

César Jesús Núñez-Prado, Grigori Sidorov*, Irina Gelbukh,
Liliana Chanona-Hernández

Instituto Politécnico Nacional,
Centro de Investigación en Computación, Ciudad de México,
México

cnunezp@ipn.mx, sidorov@cic.ipn.mx, ir.gelbukh@gmail.com

Resumen. Esta investigación explora la desambiguación del sentido de los verbos polisémicos en las definiciones de un diccionario digital (*WordNet*) a través de dos métricas de similitud: Similitud Coseno y el Algoritmo de Lesk Simplificado Modificado. Estos métodos fueron aplicados con el objetivo de identificar los tres sentidos de mayor correspondencia para cada verbo, facilitando así la selección del sentido más adecuado según el contexto en el que aparece. La desambiguación de sentido en verbos es una tarea fundamental en el campo del Procesamiento del Lenguaje Natural (PLN), con amplias aplicaciones que incluyen traducción automática, recuperación de información y análisis semántico, y ha sido evaluada en competiciones como SensEval y SemEval. En esta investigación, *WordNet* fue empleado para construir un conjunto de datos que incluye verbos con múltiples definiciones, representando un escenario común y desafiante en la desambiguación léxica. Para validar los resultados, se utilizó una evaluación manual por expertos, permitiendo establecer una referencia confiable sobre el sentido correcto de cada verbo. Los resultados de precisión fueron del $82{,}78\,\%$ para el método de Similitud Coseno y del $73{,}12\,\%$ para el Algoritmo de Lesk Simplificado Modificado, evidenciando la eficacia relativa de cada método y proporcionando un punto de partida para mejorar modelos de desambiguación en tareas avanzadas de PLN.

Palabras clave. Procesamiento del lenguaje natural, desambiguación del sentido de los verbos, similitud coseno, algoritmo de Lesk simplificado modificado, WordNet.

Similarity Metrics for Automatic Verb Sense Disambiguation Using a Corpus based on WordNet

Abstract. This research explores the sense disambiguation of polysemous verbs in the definitions of a digital dictionary (*WordNet*) through two similarity metrics: Cosine Similarity and the Modified Simplified Lesk Algorithm. These methods were applied with the aim of identifying the three senses of greatest correspondence for each verb, thus facilitating the selection of the most appropriate sense according to the context in which it appears. Verb sense disambiguation is a fundamental task in the field of Natural Language Processing (NLP), with broad applications including automatic translation, information retrieval and semantic analysis, and it has been evaluated in competitions such as SensEval and SemEval. In this research, *WordNet* was employed to build a dataset that includes verbs with multiple definitions, representing a common and challenging scenario in lexical disambiguation. To validate the results, a manual evaluation by experts was used, allowing to establish a reliable reference on the correct meaning of each verb. Accuracy results were 82.78% for the Cosine Similarity method and 73.12% for the Modified Simplified Lesk Algorithm, evidencing the relative effectiveness of each method and providing a starting point for improving disambiguation models in advanced PLN tasks.

Keywords. Natural language processing, verb sense disambiguation, cosine similarity, modified simplified Lesk algorithm, WordNet.

1. Introducción

Al sistema empleado por los seres humanos para llevar a cabo el proceso de comunicación se le denota como lenguaje natural. Este sistema es estudiado desde diversas disciplinas, tales como la filosofía, la lingüística y la lingüística computacional y se caracteriza por presentar alta ambigüedad, variabilidad y una fuerte dependencia al contexto. Estas ciencias analizan el lenguaje natural a través de su estructura gramatical, la capacidad de generar un número infinito de expresiones y los retos que plantea para su comprensión automática.

Noam Chomsky en su obra [4] lo define como un sistema generativo que utiliza un conjunto finito de reglas gramaticales para producir un número infinito de oraciones, incluyendo oraciones que no han sido pronunciadas antes. Estas reglas describen cómo las oraciones se pueden estructurar en un nivel sintáctico, sin necesariamente preocuparse por el significado.

Por otra parte, en [25] Steven Pinker define al lenguaje natural como un instinto biológico único de la especie humana, el cual surge de manera natural en el cerebro, en donde a través de reglas gramaticales innatas, los seres humanos adquieren y utilizan el lenguaje de manera espontánea para comunicarse, sin la necesidad de un aprendizaje formal extenso.

De manera general, en las diversas fuentes y disciplinas que tratan el lenguaje natural, coinciden en que este es un sistema de comunicación desarrollado espontáneamente por los seres humanos [7] en donde las palabras cuentan con múltiples significados y el contexto tiene un papel clave para la desambiguación semántica [31].

Entre sus principales características se encuentran la variabilidad, la ambigüedad y la polisemia y la comprensión automática de dicho lenguaje se puede realizar mediante modelos estadísticos, de aprendizaje automático y aprendizaje profundo [13]. En su conjunto, es posible considerar al lenguaje natural como una característica distintiva de los seres humanos desarrollada para facilitar la comunicación, el aprendizaje, la transmisión del conocimiento y la expresión de la cultura.

En términos simples, se puede considerar a este tipo de lenguajes como un sistema desarrollado con la principal finalidad de comunicarse entre sí. Sin embargo, este lenguaje humano posee algunas características que lo convierten en un sistema complejo debido a ciertos factores linguísticos, tales como la sinonimia (palabras con significados similares o idénticos), antonimia (palabras con significados opuestos), polisemia (palabras con múltiples significados) y la homonimia (palabras con la misma escritura pero con significados diferentes), por mencionar algunos.

Una de las disciplinas encargadas del estudio del lenguaje natural es el Procesamiento de Lenguaje Natural (PLN) el cual es un campo interdisciplinario que abarca áreas como la lingüística, la inteligencia artificial y las ciencias de la computación, cuyo objetivo principal es el desarrollo de sistemas que tengan la capacidad de comprender y reproducir el lenguaje humano de la misma manera en que los seres humanos lo hacen.

Dentro del conjunto de técnicas aplicadas se han empleado modelos estadísticos y de probabilidad o introducido conceptos como la entropía en la interpretación de palabras para abordar la ambigüedad y variabilidad [28]. En los últimos años, los modelos basados en redes neuronales y aprendizaje profundo, han revolucionado el campo del PLN al mejorar significativamente la precisión de tareas como la traducción automática, el análisis de sentimientos y el reconocimiento del habla [11].

La introducción de modelos pre-entrenados como *BERT* de Google y *GPT* de OpenAI ha impulsado el desarrollo de sistemas de PLN que tienen la capacidad de comprender y generar lenguaje con una fluidez y precisión nunca antes alcanzada [6].

Hoy en día, el PLN es un área en rápido crecimiento, con aplicaciones en asistentes virtuales, sistemas de recomendación y chatbots, los cuales buscan alcanzar una comprensión completa del lenguaje y el contexto humano para proporcionar respuestas cada vez más precisas y contextualmente adecuadas [19].

Tabla 1. Estadísticas de *WordNet*

Etiqueta	Número de synsets
Sustantivo	82,115
Verbo	13,767
Adjetivo	18,156
Adverbio	3,621

Dentro del amplio conjunto de tareas que aborda el PLN se encuentra una fase intermedia recurrente y fundamental para la comprensión del lenguaje humano y es denominada como desambiguación del sentido de las palabras (*WSD*, por sus siglas en inglés *Word Sense Disambiguation*), la cual busca encontrar el sentido correcto de cada palabra de acuerdo con el contexto en el que aparece [30].

De manera general, el ser humano resuelve esta tarea apoyándose de un diccionario y considerando de entre las definiciones, la mejor de ellas de acuerdo al contexto en el que se presentan. La ambigüedad léxica es un desafío para aplicaciones en PLN, tales como la traducción automática, análisis semántico, sistemas de respuestas a preguntas o generación de resúmenes ya que se requiere una comprensión profunda del contexto y una interpretación correcta de los sentidos léxicos [23].

En esta investigación buscamos calcular la similitud semántica entre los verbos polisémicos contenidos en las definiciones de un diccionario explicativo digital (*WordNet*) y sus mismas definiciones apoyándonos del contexto proporcionado para la elección del mejor sentido.

La estructura de este artículo es como sigue: en la sección 2, describimos las investigaciones actualizadas referentes a la desambiguación automática de los sentidos de las palabras. En la sección 3, abordamos algunos conceptos relevantes para la investigación. En la sección 4, describimos el conjunto de datos utilizado. En la sección 5, detallamos la metodología empleada. Y finalmente, en la sección 6, se presentan los resultados.

2. Trabajos relacionados

La desambiguación del sentido de las palabras es una tarea de gran relevancia en el ámbito del PLN, que consiste en seleccionar la acepción correcta de una palabra en un contexto dado. Este problema ha sido extensamente estudiado por parte de investigadores en el área y también ha sido objeto de evaluación en diversos eventos y competiciones (*SemEval* y *SensEval*) [17] [18], debido a su impacto en aplicaciones como la traducción automática, la recuperación de información y el análisis semántico.

La desambiguación de sentidos no sólo afecta a sustantivos y adjetivos, sino también a verbos y preposiciones, cuyas interpretaciones están ligadas fuertemente al contexto [23]. En cuanto a las investigaciones que abordan la desambiguación del sentido de las palabras utilizando el contexto en el que aparecen, se puede encontrar un método basado en diccionarios, en donde emplean un enfoque de superposición de definiciones para la desambiguación léxica. Utilizan un conjunto de datos basado en diccionarios tradicionales y pruebas en textos específicos de dominio limitado.

El método de *Lesk* alcanzó una precisión media del 50 % en conjuntos de prueba restringidos a palabras de alta ambigüedad, demostrando limitaciones en textos de mayor variabilidad contextual [14]. Por otra parte, con el desarrollo de *WordNet*, propusieron un marco léxico para la tarea de desambiguación, organizando palabras en conjuntos de sinónimos y jerarquías semánticas.

Los investigadores aplicaron este marco a dicha tarea sobre un corpus de pruebas derivado de *WordNet*, alcanzando un 72 % de precisión en la asignación de palabras a sentidos correctos en un corpus de noticias. Este resultado mostró la utilidad de *WordNet* para la desambiguación en textos de lenguaje formal [22].

En el trabajo en el que se presentó a *Word2Vec*, se utilizó un modelo de redes neuronales para aprender representaciones vectoriales a partir de un corpus extenso como *Google News*, el cual contiene aproximadamente 100 mil millones de palabras.

Tabla 2. Distribución de verbos polisémicos en cada archivo *JSON*

Archivo	Número de verbos
1	132
2	124
3	134
4	131
5	126
6	122
7	144
8	121
9	130
10	128

Los modelos *Skip-gram* y *CBOW* lograron resultados sólidos en tareas de similitud semántica, con una precisión del 75 % en pruebas de similitud y analogía semántica. Aunque el modelo no fue diseñado específicamente para la desambiguación del sentido de las palabras, permitió capturar relaciones semánticas, abriendo el camino para su aplicación en la desambiguación léxica [20].

Continuando con los modelos de aprendizaje automático, en [27] propusieron una arquitectura basada en modelos de redes neuronales recurrentes (*RNNs* por sus siglas en inglés *Recurrent Neural Networks*) y atención, utilizando un conjunto de datos de evaluación consolidado a partir de *SemCor*, *SensEval* y *SemEval*.

Utilizaron la representación de vectores incrustados de palabras (*word embeddings*) y atención para captar relaciones contextuales complejas. En este trabajo obtuvieron una precisión del 78 % en *SemEval* y un 82 % en *Senseval-2*, mostrando que los modelos enfocados en atención presentan una mejora en la precisión de la tarea de la desambiguación, particularmente en la desambiguación de verbos y preposiciones.

Por último, en el trabajo en el cual fue introducido *BERT* (por sus siglás en inglés *Bidirectional Encoder Representations from Transformers*), emplearon un modelo de transformers bidireccional preentrenado en el corpus *BookCorpus* y *Wikipedia*, utilizando aproximadamente 3 mil millones de palabras. El modelo fue afinado para la desambiguación del sentido de las palabras utilizando el conjunto de datos de *SemEval*, logrando una precisión del 91 % en los datos de prueba, superando a los modelos previos en esta tarea. Además, se logró un 92 % de precisión en el corpus *Senseval-2*, mostrando la efectividad de *BERT* para capturar dependencias contextuales y resolver ambigüedades de manera precisa [6].

En conjunto, estos trabajos subrayan la evolución y la diversidad de enfoques en la desambiguación del sentido de las palabras, desde la incorporación de heurísticas basadas en relaciones hasta metodologías de aprendizaje automático y aprendizaje profundo. Cada una de dichas metodologías proporciona una aportación a la comprensión automática del lenguaje a esta tarea fundamental en el procesamiento del lenguaje natural.

3. Conceptos relevantes

En esta sección se abordarán de manera general algunos temas de importancia utilizados en esta investigación.

3.1. WordNet

WordNet es una base de datos léxica desarrollada en el idioma inglés por la Universidad de Princeton, cuyo objetivo principal es modelar relaciones semánticas entre palabras para facilitar aplicaciones del procesamiento de lenguaje natural [22, 9]. Agrupa palabras en conjuntos de sinónimos llamados *synsets*, que abarcan sustantivos, verbos, adjetivos y adverbios. Las relaciones que conectan estos elementos son de tipo conceptual-semántico y léxico. Aunque existen diversas relaciones, la agrupación predominante es la de sinonimia.

Este diccionario digital está disponible de forma gratuita a través de la biblioteca *Natural Language Toolkit* (*NLTK* por sus siglas en inglés), lo que facilita su acceso y uso en aplicaciones del campo del procesamiento del lenguaje natural.

La Tabla 1 presenta las estadísticas generales de la versión de *WordNet* utilizada en esta investigación. Esta base de datos proporciona un total de 117,659 synsets, distribuidos en sus cuatro categorías principales.

3.2. Word2Vec

La conversión de palabras en representaciones vectoriales (*word embeddings*) responde a la necesidad de realizar operaciones matemáticas significativas entre palabras, una tarea difícil cuando se consideran simplemente como cadenas de caracteres.

Además, la mayoría de los algoritmos de inteligencia artificial están diseñados para trabajar con entradas numéricas en lugar de cadenas de caracteres. En este contexto, la transformación de palabras en vectores es esencial.

Word2Vec, desarrollado por [20, 21], emplea una red neuronal para aprender representaciones vectoriales de palabras a partir de un corpus extenso de texto. Su proceso de entrenamiento utiliza modelos de aprendizaje como *Skip-gram* y *CBOW* (*Continuous Bag of Words*), que capturan relaciones semánticas y similitudes entre palabras.

Estos modelos optimizan la representación de palabras en un espacio vectorial donde las palabras con significados similares tienden a estar más próximas entre sí. Los vectores generados poseen una dimensionalidad fija, que determina tanto la precisión como la capacidad de capturar relaciones contextuales.

En esta investigación, se emplearon los vectores incrustados de 300 dimensiones proporcionados por el modelo preentrenado *GoogleNews-vectors-negative300.bin*[1], uno de los modelos más utilizados en procesamiento del lenguaje natural debido a su rica representación semántica.

Estos vectores permiten que las palabras se integren de manera efectiva en algoritmos de inteligencia artificial, posibilitando operaciones y cálculos que reflejan las relaciones semánticas en un espacio vectorial.

[1] Disponible en: code.google.com/archive/p/word2vec/

Tabla 3. Resultados de similitud coseno

Arch	Opción	Mejor	Bolsa 2	Ning	Total
1	1	46.97	28.03	25.0	75.0
	2	31.06	26.52	42.42	57.58
	3	32.58	26.52	40.91	59.09
2	1	51.61	30.65	17.74	82.26
	2	37.90	39.52	22.58	**77.42**
	3	31.45	35.48	33.06	66.94
3	1	53.73	23.13	23.13	76.87
	2	37.31	22.39	40.30	55.97
	3	26.12	29.85	44.03	55.97
4	1	48.85	29.01	22.14	77.86
	2	38.93	32.06	29.01	70.99
	3	27.48	40.46	32.06	67.94
5	1	62.70	23.02	14.29	85.71
	2	42.86	34.13	23.02	76.98
	3	29.37	30.95	39.68	60.32
6	1	54.10	29.51	16.39	83.61
	2	40.98	24.59	34.43	65.57
	3	29.51	23.77	46.72	53.28
7	1	44.44	38.89	16.67	83.33
	2	36.81	34.72	28.47	71.53
	3	25.69	35.42	38.89	61.11
8	1	56.20	25.62	18.18	81.82
	2	38.02	33.06	28.93	71.07
	3	23.97	35.54	40.50	59.50
9	**1**	62.31	29.23	8.46	**91.54**
	2	32.31	41.54	26.15	73.85
	3	40.00	36.15	23.85	**76.15**
10	1	60.16	29.69	10.16	89.84
	2	34.38	36.72	28.91	71.09
	3	31.25	39.84	28.91	71.09

3.3. Métricas de similitud

Estas métricas permiten cuantificar la semejanza entre dos objetos, tales como palabras, frases, documentos, o vectores.

Tabla 4. Promedio de los resultados de similitud coseno

Sentido	Mejor	Bolsa	Ning	Total
1	54.11	28.68	17.22	82.78
2	37.06	32.53	30.42	69.21
3	29.74	33.40	36.86	63.14

En el contexto de PLN y aprendizaje automático, estas métricas son esenciales para comparar representaciones textuales, clasificar documentos, o realizar tareas de desambiguación de palabras.

Estas métricas pueden dividirse de acuerdo a la información que analizarán, por ejemplo; si se trata de calcular la similitud de dos objetos basados en la distancia que los separa, podemos encontrar la Distancia Euclidiana la cual se calcula con la raíz cuadrada de la suma de las diferencias al cuadrado entre cada una de las componentes [8]; la Distancia de Jaccard, la cual compara conjuntos calculando la proporción de la intersección sobre la unión de los elementos [12]; la Distancia de Hamming que mide diferencias en posiciones específicas entre cadenas de igual longitud [10], etc.

Si el enfoque es sobre cadenas de caracteres encontramos la Distancia de Levenshtein, la cual calcula el número mínimo de operaciones de edición necesarias para transformar una cadena en otra [15] o la Distancia de Damerau-Levenshtein, que es una extensión de la Distancia Levenshtein pero que incluye las transposiciones [5], etc.

3.3.1. Similitud coseno

La similitud coseno es una métrica utilizada para medir la similitud entre dos vectores en un espacio vectorial, llevando a cabo la normalización de cada vector y evaluando el coseno del ángulo entre ellos [29]. Este enfoque es especialmente útil en el PLN cuando los datos textuales son representados como vectores.

La idea fundamental detrás de la similitud coseno es calcular qué tan parecidos son dos vectores en relación con el ángulo formado entre ellos.

El resultado de esta medida siempre se encontrará en el rango de $[-1,1]$. Cuando el cálculo de la similitud tiende hacia -1 nos indica que los vectores son opuestos entre sí, cuando el resultado se acerca a 0, indica que los vectores son casi ortogonales y, por lo tanto, muy diferentes.

Por otro lado, cuando el resultado se acerca a 1, sugiere que los vectores son muy similares, e incluso podrían superponerse en gran medida. Desde una perspectiva geométrica, la similitud coseno puede entenderse visualmente como la proyección de un vector sobre otro.

Cuando los vectores están alineados, la proyección es máxima y, por lo tanto, la similitud coseno se acerca a 1. En el caso de vectores ortogonales, la proyección es mínima, y la similitud coseno se aproxima a 0 y cuando se trata de vectores opuestos, el cálculo tenderá a -1.

En el contexto del PLN, este algoritmo se utiliza para comparar la similitud semántica entre documentos o términos. Al calcular la similitud coseno entre vectores de términos o documentos, es posible evaluar su proximidad semántica.

La similitud coseno pertenece al conjunto de algoritmos de fuerza bruta, ya que implica calcular la similitud entre un vector y todos los demás vectores en el conjunto de datos. Este enfoque exhaustivo puede ser computacionalmente intensivo, pero garantiza encontrar la similitud más cercana. La definición matemática del cálculo de la similitud coseno entre dos vectores U y V se calcula mediante la siguiente fórmula:

$$\text{Similitud Coseno}(U,V) = \frac{U \cdot V}{\|U\|\|V\|}. \qquad (1)$$

Esta fórmula cuantifica la relación coseno del ángulo entre los vectores U y V, proporcionando una medida numérica de su similitud, donde $U \cdot V$ es el producto punto y $\|U\|\|V\|$ corresponde al producto de las normas entre los vectores.

Es posible aplicar esta métrica a diversas aplicaciones dentro del campo del PLN tales como la recuperación de la información [32], clasificación de texto, recomendación de contenido [16], desambiguación del sentido de las palabras, entre algunas.

Tabla 5. Resultados de Lesk simplificado modificado

Arch	Sent	Mejor	Bolsa	Ning	Total
1	1	51.52	17.42	31.06	68.94
	2	28.03	28.03	43.94	56.06
	3	21.97	27.27	50.76	49.24
2	1	49.19	28.23	22.58	77.42
	2	31.45	29.03	39.52	60.48
	3	28.23	29.03	42.74	57.26
3	1	43.28	18.66	38.06	61.94
	2	35.82	24.63	39.55	60.45
	3	25.37	26.12	48.51	51.49
4	1	47.33	20.61	32.06	67.94
	2	32.06	32.82	35.11	64.89
	3	29.01	32.82	38.17	61.83
5	1	60.32	22.22	17.46	**82.54**
	2	36.51	23.02	40.48	59.52
	3	32.54	29.37	38.1	61.9
6	1	55.74	16.39	27.87	72.13
	2	33.61	27.87	38.52	61.48
	3	25.41	23.77	50.82	49.18
7	1	45.83	27.78	26.39	73.61
	2	28.47	33.33	38.19	61.81
	3	28.47	33.33	38.19	61.81
8	1	45.45	22.31	32.23	67.77
	2	23.14	35.54	41.32	58.68
	3	21.49	38.84	39.67	60.33
9	1	56.15	23.08	20.77	79.23
	2	43.08	33.08	23.85	76.15
	3	28.46	36.92	34.62	**65.38**
10	1	50	29.69	20.31	79.69
	2	37.5	34.38	28.12	**71.88**
	3	26.56	36.72	36.72	63.28

3.3.2. Lesk completo

El Algoritmo de Lesk Completo, fue propuesto por Michael Lesk en 1986 y es ampliamente empleado para la desambiguación del sentido de palabras polisémicas en PLN.

Este enfoque implica una comparación exhaustiva de las palabras del contexto con las definiciones de todos los sentidos posibles de la palabra polisémica.

Se utiliza un conjunto más amplio de información léxica, lo que puede incluir sinónimos, hiperónimos, hipónimos y otras relaciones semánticas presentes en recursos léxicos como *WordNet* o cualquier otro diccionario explicativo.

La ponderación y combinación de esta información se realiza para determinar el sentido más adecuado. La idea general es comparar la definición de la palabra objetivo con las definiciones de sus posibles sentidos, midiendo la cantidad de coincidencias de palabras entre ellas.

El sentido con la mayor cantidad de coincidencias se selecciona como el sentido más probable considerando la relación de que cuantas más coincidencias haya es más probable que ese sentido sea el correcto [14].

3.3.3. Lesk simplificado

El Lesk Simplificado se desarrolla como una variante del Lesk Completo, buscando reducir el proceso de desambiguación. En lugar de utilizar la definición completa de cada elemento, este algoritmo se centra en las palabras del contexto que aparecen en las definiciones de los posibles sentidos.

La diferencia clave radica en que, en el Lesk Simplificado, solo se consideran las palabras que aparecen en el contexto de la palabra objetivo y se comparan con las palabras de las definiciones [1]. Esta acción reduce la complejidad computacional, permitiendo una ejecución más rápida y eficiente donde la simplicidad y la velocidad son prioritarias.

En resumen, mientras que Lesk Completo se esfuerza por una desambiguación exhaustiva y precisa utilizando una gama más amplia de información léxica, Lesk Simplificado reduce este enfoque, favoreciendo la eficiencia en situaciones donde la complejidad completa no es necesaria.

ISSN 2007-9737

4. Conjunto de datos

El conjunto de datos con el cual se realizó la experimentación de esta investigación se formó a través de la base de datos léxica *WordNet* la cual contiene conjuntos de elementos denominados *synsets*, asociados con una definición, en algunos casos proporciona ejemplos de uso y contienen una etiqueta gramatical; la cual puede ser sustantivo, verbo, adjetivo o adverbio.

4.1. Extracción de datos

Cada *synset* en *WordNet* incluye el nombre del *synset*, la etiqueta gramatical, una definición y en algunos casos, ejemplos de uso. Para la extracción de información en esta investigación, se consideraron únicamente los *synsets* con la etiqueta *"verbo"*. Se descargaron todos los elementos que cumplían con esta condición junto con sus ejemplos de uso (si estaban disponibles), concatenando la definición y los ejemplos en una única oración. Un ejemplo de los elementos extraídos se muestra a continuación:

- Nombre del *synset*: *inhale.v.02*.
- Etiqueta del *synset*: *v*.
- Definición: *draw in (air)*.
- Ejemplo(s): *'Inhale deeply', 'inhale the fresh mountain air', 'The patient has trouble inspiring', 'The lung cancer patient cannot inspire air very well'*.

A cada una de estas oraciones concatenadas se les aplicó tokenización en unigramas y se identificó la etiqueta gramatical asociada a cada palabra, lo cual permitió localizar todos los verbos en cada oración. Luego, se determinó el número total de verbos en cada entrada y se identificaron las definiciones asociadas a cada verbo proporcionadas por *WordNet*. Para incluir un verbo polisémico en el conjunto de datos a desambiguar, se consideraron los siguientes criterios:

1. Contar con al menos 5 acepciones y
2. Contar con un máximo de 15 acepciones.

Cada verbo que cumplió con estas especificaciones se agregó al conjunto de datos a desambiguar en un archivo en formato *JSON*. A continuación se muestra un ejemplo de los datos agregados al conjunto de datos:

```
{
``definition:``undergo the biomedical
and metabolic processes of respiration
by taking up oxygen and producing carbon
monoxide'',
``verbs_to_evaluate'': 1,
``word_to_evaluate'': ``produce'',
``tag'': ``verb'',
``possible_senses'': 7,
``senses'': {
       ``sense 1'': ``bring forth or yield
       ['he tree would not produce
       fruit']'',
       ``sense 2'': ``create or manufacture
       a man-made product ['We produce
       more cars than we can sell',
       'The company has been making
       toys for two centuries']'',
       ``sense 3'': ``cause to happen,
       occur or exist ['This procedure
       produces a curious effect', 'The
       new law gave rise to many
       complaints',
       'These chemicals produce noxious
       vapor', 'the new President must
       bring about a change in the
       health care system']'',
       ...}
}
```

El campo ‘‘definition’’ presenta el contexto en el que aparece el verbo (incluyendo ejemplos de uso en algunos casos), en ‘‘verbs_to_evaluate’’ se proporciona el número de verbos a desambiguar en la oración; en ‘‘word_to_evaluate’’ se indica el verbo polisémico; y en ‘‘senses’’ se enlistan todas las definiciones asociadas con el verbo.

Cuando se requiere desambiguar múltiples verbos en una misma oración, se habilita un nuevo nivel en el archivo, en donde se proporciona la información individual de cada verbo junto con sus acepciones correspondientes.

Se generaron 10 archivos en formato *JSON*, cada uno conteniendo 100 *synsets*. Cada *synset*

cuenta con al menos un verbo a desambiguar. Dado que no es requisito limitarse a un solo verbo por definición, se incluyeron todos los verbos dentro de cada oración que cumplieran con las especificaciones de contar con entre 5 y 15 acepciones.

En la Tabla 2 se muestra la distribución de verbos polisémicos a analizar en cada uno de los archivos *JSON* generados. En total, el banco de datos generado contiene $1,000$ *synsets* y $1,292$ verbos a desambiguar, distribuidos conforme a las especificaciones de polisemia requeridas. Se decidió mantener la división en 10 archivos y no unirlos en uno solo para poder llevar un mejor control en el etiquetado manual de los datos.

4.2. Etiquetado manual de los datos

Se solicitó el apoyo de 29 estudiantes y un profesor de inglés del Centro de Estudios Nacional de Lenguas Extranjeras (CENLEX Zacatenco) y de la Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME Zacatenco) del Instituto Politécnico Nacional (IPN) de México, para realizar la tarea de asignación manual de los sentidos de los verbos del banco de datos generado.

A cada una de las 30 personas involucradas se le asignó un archivo, de modo que cada archivo contó con 3 revisiones independientes. La tarea asignada consistió en:

- Identificar el verbo a desambiguar,
- Leer la oración en la que aparece el verbo,
- Revisar todas las definiciones del verbo (incluidos los ejemplos de uso),
- Seleccionar las 3 opciones que mejor se ajustan al contexto de la oración:
  - La opción número 1 es la definición que mejor se ajusta al contexto de la oración.
  - Las opciones 2 y 3 representan definiciones aceptables en caso de que la opción número 1 no estuviera disponible.

Esta clasificación se definió así para casos en los que las definiciones eran muy similares. Para la asignación final de los sentidos para cada verbo evaluado, se generaron dos “bolsas” de asignaciones: la primera solo incluye el sentido elegido como mejor opción por cada evaluador, mientras que la segunda incluye los sentidos que los evaluadores seleccionaron como segunda y tercera opción. En ambas bolsas, cada sentido solo se agrega una vez (si un sentido se seleccionó varias veces, se incluye solo una vez).

5. Metodología

En esta sección se detallará la metodología empleada en esta investigación.

5.1. Preprocesamiento

Para la asignación automática de los sentidos en los verbos del conjunto de datos, se utilizó la oración que contiene el verbo a desambiguar junto con todas sus acepciones posibles, aplicando el siguiente preprocesamiento:

- Conversión a minúsculas.
- Tokenización en unigramas.
- Eliminación de caracteres numéricos.
- Etiquetado de las partes del discurso.
- Lematización.
- Eliminación de palabras vacías.
- Eliminación del verbo a desambiguar en el contexto.

La conversión a minúsculas se realizó debido a que, computacionalmente, dos cadenas de caracteres son diferentes incluso si solo varían en el uso de una letra mayúscula. Este paso permitió homogeneizar el conjunto de datos. La tokenización en unigramas dividió las cadenas de caracteres en palabras individuales, tratándolas como tokens separados, incluidas las marcas de puntuación. Se eliminaron los caracteres numéricos, dado que no aportan significado semántico dentro de las oraciones.

Posteriormente, se aplicó el etiquetado de las partes del discurso, el cual clasifica gramaticalmente cada token. Luego, cada token se lematizó, obteniendo su forma base o sin flexión. Tanto el etiquetado gramatical como la lematización se realizaron con la biblioteca *Stanza* [26]. Adicionalmente, se eliminaron palabras vacías, ya que aportan mínimo contenido semántico; esta eliminación se llevó a cabo con la lista de palabras vacías de *NLTK* (Natural Language Toolkit) [2].

5.2. Vectorización

Concluido el preprocesamiento, se avanzó hacia la fase de vectorización de cada oración y de las acepciones posibles para el verbo a desambiguar. Este paso es fundamental, ya que convierte cada palabra de la oración en una representación numérica que facilita el análisis computacional de su contenido semántico.

Para llevar a cabo la vectorización, se consideró cada palabra dentro de la oración y se buscó su representación vectorial en el conjunto de vectores preentrenados de *Word2Vec*, específicamente en el modelo *GoogleNews-vectors-negative300.bin*, el cual proporciona representaciones de alta dimensionalidad con 300 dimensiones.

El conjunto de vectores de *Word2Vec* incluye vectores incrustados para una vasta cantidad de palabras en el idioma inglés, lo cual resulta ventajoso al permitir la obtención de vectores preentrenados sin necesidad de entrenar un modelo desde cero. A medida que se encontraba el vector de cada palabra en la oración, estos se sumaban para obtener una representación compuesta de toda la oración en un solo vector.

Para lograr esto, se calculó el promedio de los vectores obtenidos por cada palabra, generando así un vector promedio que representa la oración en su totalidad. Algo a considerar es que aunque *Word2Vec* fue entrenado con conjuntos grandes de información, aún así no cuenta con un vector asociado para todas las palabras. Cuando se presentó este caso en particular se consideró un vector de ceros para representar dichas palabras. Esta metodología busca capturar el sentido general de la oración mediante un vector que recoge las características semánticas de todas las palabras que la componen. El mismo procedimiento se aplicó a cada una de las posibles definiciones o acepciones del verbo a desambiguar, permitiendo así una comparación uniforme entre el vector de la oración y los vectores de las acepciones. Este método, además de ser eficiente, facilita la evaluación de similitudes entre los vectores generados, lo cual es fundamental para la desambiguación automática del verbo.

5.3. Métricas de similitud

Se empleó el algoritmo de *Similitud Coseno* proporcionado por la biblioteca *SciKit Learn*, como se detalla en [24, 3], con el fin de comparar la relación angular entre el vector incrustado de la oración que contiene el verbo a desambiguar y los vectores incrustados de todas las acepciones posibles de dicho verbo.

Este proceso de comparación permite identificar qué tan cercanas son las acepciones al contexto específico del verbo en la oración original, ofreciendo una medida cuantitativa de similitud entre cada vector. La Similitud Coseno es ampliamente reconocida por su capacidad para medir el grado de coincidencia entre dos vectores, ya que el valor resultante oscila entre -1 y 1.

En este caso, un valor más cercano a 1 indica una mayor similitud contextual. Tras calcular la similitud entre el vector de la oración y los vectores de las acepciones, se seleccionaron las tres coincidencias de similitud más altas por cada oración para profundizar en la evaluación y análisis de los resultados de desambiguación.

Por otro lado, el segundo método implementado en esta investigación fue el algoritmo de *Lesk Simplificado Modificado*. A diferencia del algoritmo de Similitud Coseno, que trabaja en el espacio vectorial, el algoritmo de Lesk se basa en un enfoque de intersección de palabras, comparando las palabras de la oración original con las palabras en cada definición. Lesk, en su forma básica, determina la acepción correcta al calcular la intersección máxima entre la oración que contiene el verbo objetivo y las palabras que componen cada definición. Sin embargo, este enfoque tradicional fue ampliado en nuestra investigación.

Tabla 6. Promedio de los resultados de Lesk simplificado modificado

Sentido	Mejor	Bolsa	Ning	Total
1	50.48	22.64	26.88	73.12
2	32.97	30.17	36.86	63.14
3	26.75	31.42	41.83	58.17

Además de las palabras directas de cada oración y definición, se incorporaron las definiciones de cada palabra contenida en las oraciones y en las acepciones, es decir; cada palabra se expandió con su definición correspondiente para enriquecer su bolsa de palabras. Este proceso adicional amplió sustancialmente las bolsas de palabras tanto de la oración como de cada una de las definiciones.

Al hacerlo, el algoritmo de Lesk Simplificado Modificado pudo capturar más matices semánticos en el proceso de comparación, incrementando las posibilidades de una desambiguación más precisa en contextos complejos. Del mismo modo, se eligieron los 3 mejores resultados obtenidos con este algoritmo por cada entrada.

6. Resultados

Se aplicaron ambas métricas de similitud sobre los 10 archivos del banco de datos, evaluando los 3 mejores resultados obtenidos para cada verbo polisémico comparándolos con los sentidos marcados por los etiquetadores. En la Tabla 3, se presentan los resultados del algoritmo de Similitud Coseno aplicado a los 10 archivos.

De la Tabla 3, la columna `"Arch"` indica el número de archivo evaluado, se debe recordar que cada archivo contiene 100 *synsets* y que es posible evaluar más de un sentido por *synset*. La columna `"Opción"` enumera los sentidos elegidos por los algoritmos de mayor a menor. La columna `"Mejor"` indica la proporción en que se encontró el sentido a evaluar por parte del algoritmo como primera opción por parte de los evaluadores.

La columna `"Bolsa 2"` indica la proporción en que el sentido a evaluar apareció como segunda o tercera opción para los evaluadores. La columna `"Ning"` indica el número de veces en que el resultado del algoritmo no coincidió con ninguna de las opciones elegidas por parte de los evaluadores. La columna de `"Total"` representa la suma de la columna `"Mejor"` más la columna `"Bolsa 2"`. Las cantidades de las columnas `"Mejor"`, `"Bolsa 2"`, `"Ning"` y `"Total"` están expresadas en porcentajes.

En el archivo 9 se encontraron los mejores resultados para los sentidos 1 y 3 obteniendo el 91.54 % y el 76.15 % respectivamente. Por otro lado, el sentido 2 obtuvo los mejores resultados en el archivo 2 con un 77.42 %. En la Tabla 4 se muestra el promedio de los 10 archivos evaluados para cada uno de los sentidos aplicando el algoritmo de Similitud Coseno.

En la Tabla 5, se muestran los resultados derivados de la aplicación del algoritmo de Lesk Simplificado Modificado. Para el algoritmo de Lesk Simplificado Modificado, en el archivo 5 se encontró el mejor resultado para el sentido 1 alcanzando un 82.54 %, en el archivo 10 para el sentido 2 con 71.88 % y finalmente para el sentido 3 en el archivo 9 con 65.38 %. En la Tabla 6 se muestra el promedio de los 10 archivos evaluados para cada uno de los sentidos aplicando el algoritmo de Lesk Simplificado Modificado.

7. Conclusiones

En esta investigación se abordó la desambiguación automática de verbos en las definiciones de un diccionario digital, específicamente utilizando *WordNet*, a través de dos métricas de similitud: Similitud Coseno y el Algoritmo de Lesk Simplificado Modificado.

La meta principal era identificar el sentido correcto de las palabras en función del contexto, una tarea crítica en el campo del Procesamiento del Lenguaje Natural (PLN), que tiene aplicaciones clave en la comprensión de textos, traducción automática, y en el desarrollo de sistemas más precisos para la interacción humano-computadora.

Durante el proceso de evaluación de estos métodos, se utilizaron datos verbales de *WordNet* que presentaban múltiples definiciones para cada verbo objetivo. La comparación de los resultados obtenidos contra las etiquetas de sentido correcto, proporcionadas mediante la evaluación manual,

reveló una precisión del 82.78 % para el método de Similitud Coseno y del 73.12 % para el Algoritmo de Lesk Simplificado Modificado. Estos resultados destacan la eficacia de ambos enfoques en la tarea de desambiguación, aunque cada uno presenta fortalezas particulares en función de las características del texto evaluado.

El rendimiento superior del método de Similitud Coseno, que alcanzó una precisión del 82.78 %, indica su capacidad para capturar matices contextuales donde los significados de los verbos dependen fuertemente de las palabras circundantes. La representación vectorial, basada en *Word2Vec*, permitió que este método comparara de manera más robusta el contexto semántico completo de cada definición.

Por otro lado, el Algoritmo de Lesk Simplificado Modificado, que mostró un rendimiento del 73.12 %, aunque con una precisión menor, destacó por su eficiencia y rapidez en comparación con métodos de procesamiento vectorial intensivo. Este algoritmo demostró ser adecuado para tareas de desambiguación menos complejas, donde la cantidad de información contextual es limitada y donde su enfoque basado en coincidencias léxicas puede captar significados de forma eficaz.

Además, la investigación revela desafíos inherentes en la tarea de desambiguación, como la limitación de datos, especialmente en definiciones con acepciones muy similares entre sí. Esta situación afectó particularmente al Algoritmo de Lesk Simplificado Modificado, el cual depende de la superposición léxica y puede tener dificultades para diferenciar definiciones con leves variaciones semánticas.

Esta limitación sugiere la importancia de incorporar técnicas híbridas que combinen la capacidad de los enfoques basados en contextos amplios con la eficiencia de métodos simplificados, para así ofrecer una desambiguación precisa y escalable en distintos niveles de complejidad textual.

Los resultados de esta investigación subrayan la necesidad continua de mejorar y refinar las técnicas de desambiguación de sentidos para el procesamiento de lenguaje natural, con el fin de abordar contextos más complejos y mejorar la aplicabilidad práctica de estas metodologías.

Como trabajo a futuro se podría combinar los métodos para lograr una mayor precisión y robustez ante definiciones ambiguas, así como en la adaptación de estos métodos a otros lenguajes y dominios específicos donde la precisión semántica es esencial.

Agradecimientos

Este trabajo fue realizado con el apoyo parcial del Gobierno Mexicano a través de la beca A1-S-47854 de CONACYT, México, y las becas 20232138, 20232080, 20231567 de la Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional, México. Los autores agradecen a CONACYT por los recursos informáticos proporcionados a través de la Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje del Laboratorio de Supercómputo del INAOE, México, y reconocen el apoyo de Microsoft a través del Microsoft Latin America PhD Award.

Referencias

1. **Banerjee, S., Pedersen, T. (2005).** An adapted lesk algorithm for word sense disambiguation using wordnet. Proceedings of the 5th Conference on Natural Language Processing (NLP), pp. 241–246.
2. **Bird, S., Klein, E., Loper, E. (2009).** Natural language processing with Python: Analyzing text with the natural language toolkit. O'Reilly Media.
3. **Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G. (2013).** API design for machine learning software: experiences from the scikit-learn project. ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122. DOI: 10.48550/arXiv.1309.0238.
4. **Chomsky, N. (1965).** Aspects of the theory of syntax. The MIT Press, Cambridge, MA.

5. **Damerau, F. J. (1964).** A technique for computer detection and correction of spelling errors. Communications of the ACM, Vol. 7, No. 3, pp. 171–176. DOI: 10.1145/363958.363994.
6. **Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Minneapolis, MN, pp. 4171–4186.
7. **Encyclopaedia Britannica (2023).** Natural language. www.britannica.com/technology/natural-language.
8. **Euclid (1883).** The elements. Cambridge University Press.
9. **Fellbaum, C. (1998).** WordNet: An Electronic Lexical Database. MA: MIT Press.
10. **Hamming, R. W. (1950).** Error detecting and error correcting codes. The Bell System Technical Journal, Vol. 29, No. 2, pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.
11. **Hinton, G. E., Osindero, S., Teh, Y. W. (2006).** A fast learning algorithm for deep belief nets. Neural Computation, Vol. 18, No. 7, pp. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527.
12. **Jaccard, P. (1901).** Étude comparative de la distribution florale dans une portion des alpes et des jura. Bulletin de la Société Vaudoise des Sciences Naturelles, Vol. 37, pp. 547–579.
13. **Jurafsky, D., Martin, J. H. (2000).** Speech and language processing. Prentice Hall, Upper Saddle River, NJ.
14. **Lesk, M. (1986).** Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24–26. DOI: 10.1145/318723.318728.
15. **Levenshtein, V. I. (1966).** Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, Vol. 10, pp. 707–710.
16. **Linden, G., Smith, B., York, D. (2003).** Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, Vol. 7, No. 1, pp. 76–80. DOI: 10.1109/MIC.2003.1167344.
17. **Litkowski, K. C. (2004).** Senseval-3 task word-sense disambiguation of wordnet glosses. Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pp. 13–16.
18. **Litkowski, K. C. (2007).** Semeval-2007 task 06: Word-sense disambiguation of prepositions. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007).
19. **Manning, C. D. (2020).** Emerging trends and future directions in natural language processing. Computational Linguistics, Vol. 46, No. 4, pp. 701–707.
20. **Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013).** Efficient estimation of word representations in vector space. Proceedings of the International Conference on Learning Representations (ICLR).
21. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013).** Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS), Curran Associates, Inc, pp. 3111–3119.
22. **Miller, G. A. (1995).** Wordnet: A lexical database for English. Communications of the ACM, Vol. 38, No. 11, pp. 39–41. DOI: 10.1145/219717.219748.
23. **Navigli, R. (2009).** Word sense disambiguation: A survey. ACM Computing Surveys, Vol. 41, No. 2, pp. 1–69. DOI: 10.1145/1459352.1459355.

24. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011).** Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, Vol. 12, pp. 2825–2830.

25. **Pinker, S. (1994).** The language instinct. William Morrow and Company, New York, NY.

26. **Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C. D. (2020).** Stanza: A Python natural language processing toolkit for many human languages. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. DOI: 10.48550/arXiv.2003.07082.

27. **Raganato, R., Delli Bovi, R., Navigli, R. (2017).** Neural sequence learning models for word sense disambiguation. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1156–1167. DOI: 10.18653/v1/D17-1120.

28. **Shannon, C. E. (1948).** A mathematical theory of communication. Bell System Technical Journal, Vol. 27, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

29. **Sidorov, G. (2013).** Construcción no lineal de n-gramas en la lingüística computacional. Sociedad Mexicana de Inteligencia Artificial.

30. **Weaver, W. (1955).** Translation. Technical report, Institute for Advanced Study, Reprinted in Machine Translation of Languages MIT Press.

31. **Wilks, Y. (1972).** Grammar, meaning and the machine analysis of language. Routledge and Kegan Paul, London.

32. **Zhang, X., Zhao, Y., Zhang, C. (2016).** A survey of content-based recommendation systems. Journal of Computer Science and Technology, Vol. 31, No. 4, pp. 719–746.

Article received on 04/09/2023; accepted on 14/12/2023.
**Corresponding author is Grigori Sidorov.*