

# Práctica de laboratorio 6: Introducción desambiguación de las palabras.

Castro Elvira D. <sup>2022710039</sup>, Pineda Rugerio N. <sup>2022710240</sup>, Castañón Hernández V. J. <sup>2022710020</sup>, Galicia Cocoltzi N. <sup>2022710234</sup>, Sánchez Zanjuampa M. A. <sup>2022710029</sup>, y Nava Mendez E. U. <sup>2021710144</sup>.

Profesor: Lauro Reyes Cocoltzi

**Resumen**—En esta práctica se analiza cómo la desambiguación de palabras y el modelado de embeddings pueden combinarse para capturar el significado contextual de las palabras, para una recuperación más precisa de información como palabras similares dentro de un corpus de texto. Al tener en cuenta tanto el significado desambiguado de acuerdo al contexto como la representación vectorial en un espacio semántico, se logró una comprensión más robusta, permitiendo mejorar la búsqueda de esta relación semántica entre palabras. Al desambiguar el sentido correcto de cada palabra según su contexto en una oración y luego mapear estos sentidos a vectores (embeddings), las palabras similares semánticamente tienden a agruparse cerca en este espacio de dimensionalidad. Esto facilita la recuperación de términos relacionados de mejor manera, superando las limitaciones de los enfoques si se utilizan por separado.

**Palabras clave**—Desambiguación, Lesk, word2vec, etiquetado POS.

## I. INTRODUCCIÓN

### I-A. Marco teórico

La ambigüedad léxica, donde una palabra puede tener múltiples significados según el contexto, es un reto significativo actualmente en el procesamiento de lenguaje natural. Por ello, el uso de algoritmos para desambiguar palabras es importante para determinar el sentido correcto de las palabras en un contexto dado. Pero...¿qué es la desambigüedad?

#### I-A1. Desambiguación de las palabras.:

Se trata de identificar el significado preciso de una palabra con múltiples significados en un contexto determinado. Se trata de discernir entre las diversas acepciones que la palabra puede tener, tomando en cuenta el entorno lingüístico en el que se emplea.

Por ejemplo, la palabra "banco" puede referirse a una institución financiera o a un asiento alargado. La desambiguación de palabras utiliza técnicas y algoritmos para determinar cuál de los posibles significados de una palabra es el correcto según el contexto pasado.

Algunos algoritmos populares para la desambiguación de palabras incluyen el algoritmo de Lesk, el cual compara definiciones de diccionario con el contexto de la palabra, y los métodos basados en corpus, que analizan cómo se utiliza una palabra en grandes conjuntos de datos de texto. En Procesamiento de Lenguaje Natural a este proceso se le conoce como Word Sense Disambiguation (WSD)

#### I-A2. Algoritmo Lesk.:

Su objetivo es determinar cuál es el significado correcto de una palabra polisémica, es decir, una palabra que tiene múltiples significados, según el contexto en el que aparece.

Primero, analiza el contexto de la palabra objetivo, esto incluye palabras cercanas en la oración o párrafo en el que se encuentra. Después consulta la base de datos WordNet para obtener su posible significado. En seguida, compara el contexto con las definiciones y ejemplos de uso obtenidos en los synset de WordNet, que es la definición que sea más adecuada según el contexto proporcionado. Finalmente, selecciona ese synset y lo asocia como el sentido más probable de la palabra objetivo.

#### I-A3. WordNet.:

WordNet es una base de datos léxica en inglés que contiene semánticas entre palabras, como sinónimos, antónimos, hiperónimos e hipónimos. Agrupa palabras en synsets, cada uno de los cuales expresa un concepto distinto, y proporciona ejemplos de uso de cada uno.

#### I-A4. SpaCy.:

SpaCy ofrece funcionalidades como tokenización, etiquetado POS, así como otras herramientas útiles para el procesamiento de texto.

#### I-A5. Etiquetado POS.:

Este etiquetado es el proceso de asignar una categoría gramatical o etiqueta de parte del discurso (part of speech) a cada palabra en un texto.

Por ejemplo:

- Sustantivo (noun)
- Verbo (verb)
- Adjetivo (adjective)
- Adverbio (adverb)
- Pronombre (pronoun)
- Preposición (preposition)
- Conjunción (conjunction)

Digamos, la oración: **"El gato negro duerme en el sofá"**, tendría el siguiente etiquetado POS:

- El/DET
- gato/NOUN
- negro/ADJ
- duerme/VERB
- en/ADP
- el/DET
- sofá/NOUN

#### I-A6. Modelo word2vec.:

Es un algoritmo para aprender representaciones vectoriales densas y de baja dimensionalidad de palabras basadas en su contexto en un corpus de texto. Tiene dos variantes: skip-gram y CBOW. Ambos predicen palabras vecinas a partir de una palabra de entrada o viceversa, respectivamente.

#### I-B. Aporte

- Uso del algoritmo Lesk y la combinación con el etiquetado de partes del discurso (POS), lo que permite una mejor comprensión del significado de las palabras en un contexto específico, y mejora la precisión de la relación de las palabras.
- Utilizar WordNet, Lesk, y word2vec, para llevar a cabo una mejor comprensión de la semántica del texto y obtener mejores resultados en la recuperación de palabras similares.
- Uso de bibliotecas como SpaCy y Gensim para implementar técnicas de modelado de embeddings, importante para trabajar con grandes conjuntos de datos.

## II. DESARROLLO

### II-A. Código 1. Word Sense Disambiguation con WordNet y Lesk.

1. **Carga de bibliotecas y recursos** El código plantea el uso de wordnet el cual es una base de datos del léxico del idioma inglés. Esta base de datos cuenta con sustantivos, verbos, adjetivos y adverbios que se agrupan en conjuntos de sinónimos llamados 'synsets' los cuales expresan un concepto diferente. Al igual que en otros trabajos se hace uso de las librerías NLTK para el procesamiento de Lenguaje Natural.
2. **Carga del modelo de Spacy:** El modelo que se utiliza se llama en\_core\_web\_sm y es uno de los modelos más ligeros de Spacy para el idioma inglés, ofrece funcionalidades como el etiquetado POS (part of speech, asigna una etiqueta a cada palabra en un texto según su categoría gramatical) y el análisis morfológico.
3. **Obtención de definiciones** Se pasa una palabra a wordnet para obtener las frases que existen para dicha palabra. De estas frases se obtienen múltiples definiciones las cuales se pueden clasificar en verbos, sustantivos, adjetivos, etc.

4. **Desambiguación con el algoritmo LESK** Lo que hace el algoritmo de Lesk es comparar el contexto de la palabra (en este caso "die") con los contextos de los diferentes sentidos de esa palabra, para esto se utiliza la base de datos y luego seleccionar el sentido que tenga el mayor solapamiento de palabras en común con el contexto dado.

---

#### Algorithm 1 Simplified Lesk Algorithm

---

```
1: function SIMPLIFIEDLESK(word, sentence)
2:   bestSense ← most frequent sense for word
3:   maxOverlap ← 0
4:   context ← set of words in sentence
5:   for each sense in senses of word do
6:     signature ← set of words in the gloss and examples
7:     overlap ← ComputeOverlap(signature, context)
8:     if overlap > maxOverlap then
9:       maxOverlap ← overlap
10:      bestSense ← sense
11:    end if
12:  end for
13:  return bestSense
14: end function
```

---

5. **Etiquetado POS** En el POS tagging (Part of Speech) cada palabra es etiquetada según su categoría gramatical. En esta se busca una frase similar que coincida con las definiciones encontradas anteriormente, ya sea sustantivos, verbos, adjetivos, etc., incluso es posible etiquetar los signos de puntuación.
6. **Analizar los resultados** Finalmente, se analizan los resultados obtenidos.

### II-B. Código 2. Desambiguación y representación vectorial de palabras.

Para el siguiente código se utiliza un corpus de texto contenido en webchat de la biblioteca NLTK, en primera instancia se trabaja con el archivo "firefox.txt".

El proceso realizado es el siguiente:

- Una vez cargado el corpus de texto, que en este caso es el archivo de texto, se realiza una limpieza de los datos, eliminando stopwords, caracteres, y aplicando un algoritmo de Stemming, esto forma parte del preprocesamiento, para después trasladarlo a un etiquetado POS.
- En el etiquetado se utiliza la biblioteca spaCy, que identifica el part of speech de cada palabra en las oraciones del corpus de texto proporcionado. Estos es trasladado a un corpus de sentidos, en el que se asocia el posible significado de la palabra según el contexto, teniendo como referencia el etiquetado realizado.
- Se utiliza el corpus de sentidos para entrenar un modelo word2vec, y se prueban dos configuraciones, una con el enfoque de skip-gram, y un modelo con parámetros específicos que indican el tamaño del vector generado para cada palabra, el tamaño de ventana, etc.

- Utilizando este modelo se puede determinar las palabras que sean más similares a otras según el contexto por oración, esto puede realizarse utilizando una sola palabra e indicando cuantas palabras similares se requieren, o bien solicitando las palabras similares a dos palabras a la vez. Esto se realiza con el método *most\_similar* de word2vec.
- Finalmente, se visualizan los espacios semánticos creados por el modelo word2vec, aplicando PCA para reducir dimensionalidad, y poder observarlo mediante una gráfica 2D.

### II-C. Unión de ambos códigos.

Para unir ambos códigos se planteó lo siguiente:

1. Utilizando el primer código se realizó la desambiguación de las palabras, primero utilizando el etiqueda POS, y se hace la mejora utilizando el algoritmo Lesk, que básicamente genera un conjunto de significados para una palabra.
2. Ya que se tienen las oraciones propuestas, se tokenizan, y son pasadas al algoritmo Lesk, de tal forma que, la oración es el contexto, y se le pasa cada una de las palabras.
3. Se utiliza spaCy para calcular las etiquetas POS de cada palabra en las oraciones.
4. Se aplica Lesk para obtener el sentido de cada palabra y se contruye un corpus de sentidos.
5. Después, se entrena un modelo de word2vec utilizando el corpus de sentidos generado, con un mínimo de veces de aparición de cada palabra igual a 1, una dimension de 50 para los vectores de palabras, un tamaño de 3 subprocesos, un tamaño de 5 para la ventana de contexto, que se considera para predecir la siguiente palabra, y se utiliza el modelo de skip-gram.
6. A continuación, se encuentran las palabras similares utilizando el modelo word2vec descrito en el paso anterior, y se visualizan los vectores de palabras en un espacio de dimensionalidad, para observarlo gráficamente se realiza una reducción de dimensionalidad con PCA y 2 componentes principales, es decir se obtienen gráficas 2D.

## III. RESULTADOS

### III-A. Código 1. Word Sense Disambiguation con WordNet y Lesk.

El código proporcionado venia con algunos ejemplos de frases y de una palabra propuestos para la demostracion de su funcionamiento, sin embargo se propusieron otros para un mejor entendimiento.

**EJEMPLO DEFAULT:** La palabra utilizada fue **die** y se utilizó en las siguientes frases:

X = 'The die is cast.'

Y = 'Roll the die to get a 6.'

Z = 'What is dead may never die.'

En este caso al pasar la palabra por Wordnet se encontraron 3 definiciones como sustantivos:

definicion 0 : a small cube with 1 to 6 spots on the six faces; used in

gambling to generate random numbers

definicion 1 : a device used for shaping metal

definicion 2 : a cutting tool that is fitted into a diestock and used for cutting male (external) screw threads on screws or bolts or pipes or rods

Mientras que se encontraron 11 definiciones que indicaban un verbo:

definicion 0 : pass from physical life and lose all bodily attributes and functions necessary to sustain life

definicion 1 : suffer or face the pain of death

definicion 2 : be brought to or as if to the point of death by an intense emotion such as embarrassment, amusement, or shame

definicion 3 : stop operating or functioning

definicion 4 : feel indifferent towards

definicion 5 : languish as with love or desire

definicion 6 : cut or shape with a die

definicion 7 : to be on base at the end of an inning, of a player

definicion 8 : lose sparkle or bouquet

definicion 9 : disappear or come to an end

definicion 10 : suffer spiritual death; be damned (in the religious sense)

Una vez obtenidas las definiciones se utiliza el algoritmo LESK para encontrar la definición de la palabra con un mejor contexto, pero la definición seleccionada aun no es del todo correcta y por eso se utiliza POS para considerar también la etiqueta de la palabra a definir. Con esto se obtiene una definición mas correcta.

**EJEMPLO 1:** La palabra utilizada fue **bat** y se utilizó en las siguientes frases:

X = 'The baseball player swung the bat and hit a home run.'

Y = 'The fruit bat sleeps hanging upside down in the cave during the day.'

Z = 'She used a tennis racket to bat away the flies buzzing around her. '

La base de datos de WordNet arrojó las siguientes definiciones basándose en los diferentes contextos y formas de la palabra.

Como sustantivo:

	X	Y	Z
Frase	The die is cast.	Roll the die to get a 6.	What is dead may never die.
LESK	cut or shape with a die	to be on base at the end of an inning, of a player	a cutting tool that is fitted into a diestock and used for cutting male (external) screw threads on screws or bolts or pipes or rods
POS	a cutting tool that is fitted into a diestock and used for cutting male (external) screw threads on screws or bolts or pipes or rods	a small cube with 1 to 6 spots on the six faces; used in gambling to generate random numbers	stop operating or functioning

Tabla I: Resultados con LESK y POS

definicion 0 : nocturnal mouselike mammal with forelimbs modified to form membranous wings and anatomical adaptations for echolocation by which they navigate  
definicion 1 : (baseball) a turn trying to get a hit  
definicion 2 : a small racket with a long handle used for playing squash  
definicion 3 : the club used in playing cricket  
definicion 4 : a club used for hitting a ball in various games

Como verbo:

defination 0 : strike with, or as if with a baseball bat  
defination 1 : wink briefly  
defination 2 : have a turn at bat  
defination 3 : use a bat  
defination 4 : beat thoroughly and conclusively in a competition or fight

A partir de aquí se realiza la desambiguación utilizando LESK y rectificando con POS:

	X	Y	Z
Frase	The baseball player swung the bat and hit a home run	The fruit bat sleeps hanging upside down in the cave during the day	She used a tennis racket to bat away the flies buzzing around her
LESK	strike with, or as if with a baseball bat	the club used in playing cricket	a small racket with a long handle used for playing squash
POS	(baseball) a turn trying to get a hit	the club used in playing cricket	use a bat

Tabla II: Resultados con LESK y POS

Como se puede ver, para la frase X tanto las definiciones obtenidas por LESK y POS fueron acertadas, el contexto que entendió el algoritmo trata sobre un bate de béisbol; para la frase Y, sin embargo, fue todo lo contrario, el algoritmo

asimiló que se trataba de un bate para jugar cricket, mientras que en realidad se trata de un murciélago (bat en inglés); la frase Z obtuvo un resultado similar, se habla sobre espantar moscas pero el algoritmo lo tomó como una raqueta para jugar squash.

**EJEMPLO 2:** La palabra utilizada fue **light** y se utilizó en las siguientes frases:

X = 'The traffic light turned red, so we had to stop the car.'  
Y = 'She found solace in the warm light of the setting sun.'  
Z = 'Please turn off the light before leaving the room. '

La base de datos de WordNet arrojó las siguientes definiciones basándose en los diferentes contextos y formas de la palabra.

Como sustantivo (solo se colocaron las primeras 5 definiciones, en total fueron 14):

definicion 0 : (physics) electromagnetic radiation that can produce a visual sensation  
definicion 1 : any device serving as a source of illumination  
definicion 2 : a particular perspective or aspect of a situation  
definicion 3 : the quality of being luminous; emitting or reflecting light  
definicion 4 : an illuminated area  
definicion 5 : a condition of spiritual awareness; divine illumination

Como verbo:

defination 0 : make lighter or brighter  
defination 1 : begin to smoke  
defination 2 : to come to rest, settle  
defination 3 : cause to start burning; subject to fire or great heat  
defination 4 : fall to somebody by assignment or lot  
defination 5 : alight from (a horse)

A partir de aquí se realiza la desambiguación utilizando LESK y rectificando con POS:

	X	Y	Z
Frase	The traffic light turned red, so we had to stop the car	She found solace in the warm light of the setting sun	Please turn off the light before leaving the room
LESK	the quality of being luminous; emitting or reflecting light	the quality of being luminous; emitting or reflecting light	the quality of being luminous; emitting or reflecting light
POS	(baseball) a turn trying to get a hit	the quality of being luminous; emitting or reflecting light	alight from (a horse)

Tabla III: Resultados con LESK y POS

Esta vez los resultados no fueron muy buenos, como se puede observar en la tabla en su mayoría el algoritmo relacionó la palabra light con la luz; a pesar de que en sí significa lo mismo en los 3 y la definición es correcta, el contexto es diferente.

### III-B. Código 2. Desambiguación y representación vectorial de palabras.

A continuación, se muestra el corpus de sentidos después de aplicar un etiquetado POS, el cual contiene 9999 palabras:

```
[['cookie_NOUN',
'manager_NOUN',
':_PUNCT',
'allow_VERB',
'sites_NOUN',
'that_PRON',
'set_VERB',
'removed_VERB',
'cookies_NOUN',
'to_PART',
'set_VERB',
'future_ADJ',
'cookies_NOUN',
'"_PUNCT',
'should_AUX',
'stay_VERB',
'checked_VERB'],
['when_SCONJ',
'in_ADP',
'full_ADJ',
'screen_NOUN',
'mode_NOUN'],
...]
```

Ahora se muestran los resultados para los dos diferentes modelos de word2vec, es decir utilizando Skip-gram o CBOW.

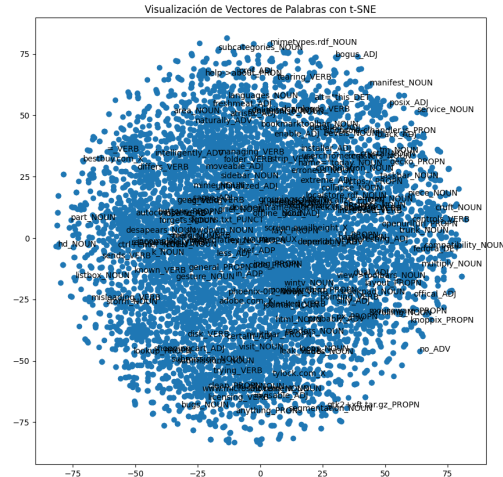
#### III-B1. Modelo CBOW.: Palabras similares a web\_NOUN e internet\_NOUN

```
general\_ADJ',
0.34646329283714294),
('yellow\_ADJ',
0.34283873438835144),
('forms\_NOUN',
0.3241673409938812),
('targeting\_VERB',
0.31832149624824524)
,...
```

Tres palabras similares a web\_NOUN

```
('secure\_ADJ',
0.47040387988090515),
('.....\_PUNCT',
0.3857492208480835),
('tools->options->web\_PROPN',
0.3680274188518524)
```

#### Representación vectorial con t-SNE:



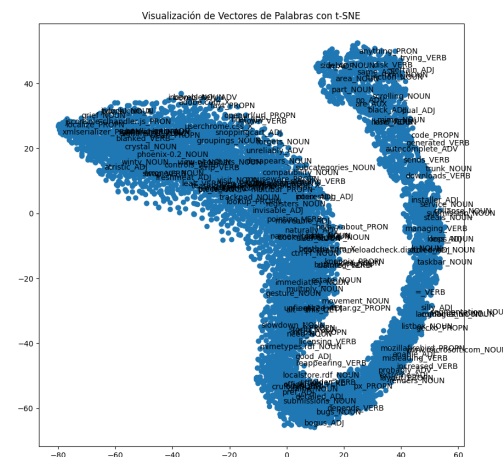
#### III-B2. Modelo Skip-gram.: Palabras similares a web\_NOUN e internet\_NOUN

```
('images_NOUN', 0.44073545932769775),
('page_NOUN', 0.4390312135219574),
('whole_ADJ', 0.4226228594779968),
('css_PROPN', 0.40971434116363525),
('text_NOUN', 0.3950037360191345),
...
```

Tres palabras similares a web\_NOUN

```
('mantis_PROPN', 0.7291897535324097),
('bugtracker_PROPN', 0.7142958641052246),
('washington_PROPN', 0.693264365196228)
```

#### Representación vectorial con t-SNE:



### III-C. Unión de ambos códigos.

A continuación, se muestra el corpus de texto generado utilizando el archivo **firefox.txt**, y la desambiguación de las palabras asignándoles el sentido más probable.

```
[['cookie.n.03',
'director.n.01',
```



```

':',
'give_up.v.11',
'web_site.n.01',
'that',
'set.v.08',
'take_out.v.01',
'cookie.n.03',
'to',
'set.v.08',
'future.n.02',
'cookie.n.03',
'",',
'should',
'stay.v.02',
'see.v.10',
['when',
'indium.n.01',
'wax.v.03',
'screen.n.05',
'mood.n.03'],
...

```

Ahora se comparan dos palabras: *internet.n.01* y *web.n.06*, y se muestran aquellas que según el modelo word2vec son más similares a ambas:

```

(['exercise.v.03', 0.46153298020362854),
('internet_explorer.n.01', 0.44807639718055725),
('port.v.01', 0.40679794549942017),
('give.v.43', 0.40302610397338867),
('up.r.05', 0.39083972573280334),
('conflicting', 0.390753835439682),
('machine.n.01', 0.3898971974849701),
('integration.n.03', 0.3888862729072571),
('dell.n.01', 0.38726648688316345),
('extention', 0.3853917717933655)]

```

Ahora se muestra el resultados con las palabras *toolbar*, *item.n.03*:

```

(['specify.v.03', 0.3582044839859009),
('taskbar', 0.3134940564632416),
('executables', 0.30302685499191284),
('vary.v.03', 0.29566851258277893),
('align.v.03', 0.2899031341075897),
('transfer.v.06', 0.281352698802948),
('cruise.n.01', 0.28080374002456665),
('drag.v.07', 0.280578076839447),
('modem.n.01', 0.27898168563842773),
('lose.v.06', 0.27585598826408386)]

```

También se pueden buscar palabras similares a una sola palabra, e indicar el número de palabras que deseamos obtener, por ejemplo 3 palabras similares a la palabra *mozilla*:

```

(['misspell.v.01', 0.36404678225517273),
('search.v.04', 0.3287902772426605),
('scat.v.01', 0.32856523990631104)]

```

Posteriormente, se hacen obtienen los vectores generados por el modelo word2vec de cada palabra, y se realiza una

reducción de dimensionalidad a 2 componentes de ese vector, para visualizarlo utilizando t-SNE (Figura 1).

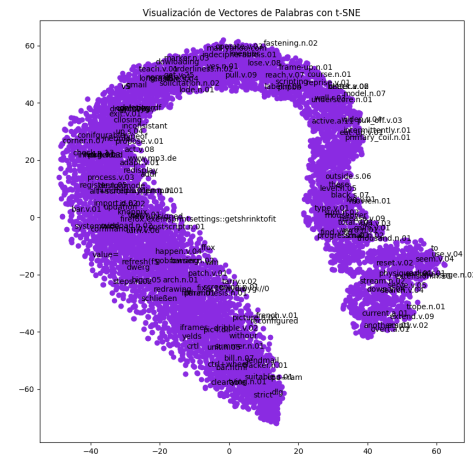


Figura 1. Vectorización de palabras en el archivo *firefox.txt*

Se realiza lo mismo utilizando ahora el archivo **grail.txt**. Se muestra el corpus de sentidos de las palabras:

Se muestran las palabras relacionadas con las palabras *spanking.n.01* y *angstrom.n.01*.

```

(['awful.s.02', 0.5297589898109436),
('transfer.v.06', 0.5247588157653809),
('aah', 0.506284236907959),
('wayy', 0.5005611181259155),
('hiyah', 0.4962727725505829),
('weather.n.01', 0.4951319992542267),
('aaah', 0.48896506428718567),
('armor.n.01', 0.4818258583545685),
('hah', 0.4798458516597748),
('hallo', 0.4762493073940277)]
...

```

Y se muestra el espacio vectorial generado (Figura 2).

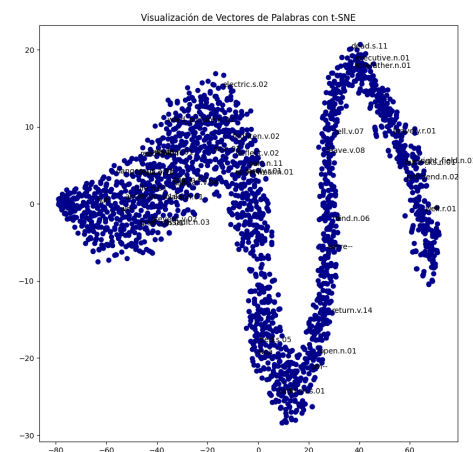


Figura 2. Vectorización de palabras en el archivo *grail.txt*

Ahora se utiliza el archivo **overhead.txt**. Se analizan el corpus de texto:

```

['guy.n.03',
':',
'ohio.n.01',

```

```
'please.v.03',
',',
'deoxyadenosine_monophosphate.n.01',
'human.a.03',
'buttocks.n.01',
'can.v.01',
'take.v.24',
'deoxyadenosine_monophosphate.n.01',
'knight.n.02',
'!'],
...
```

Se encuentran las 3 palabras más similares a *deoxyadenosine\_monophosphate.n.01*.

```
[('aruond', 0.43771716952323914),
('angstrom.n.01', 0.43519920110702515),
('dead.s.13', 0.4214862287044525)]
```

Y se muestra su espacio vectorial (Figura 3).

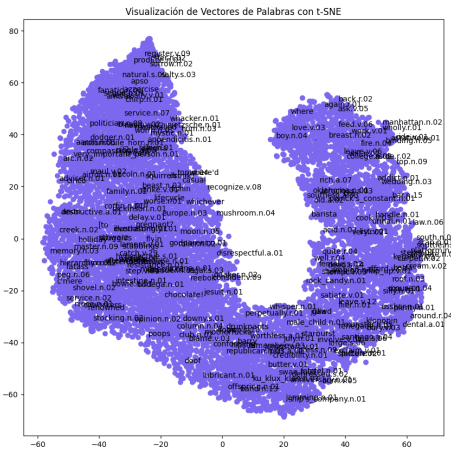


Figura 3. Vectorización de palabras en el archivo overhead.txt

Finalmente, para el archivo **pirates.txt**. Se analizan el corpus de texto:

```
['set_up.v.04',
'out.s.04',
'entirely.r.02',
'on.r.03',
'deoxyadenosine_monophosphate.n.01',
'longboat.n.01',
']'],
...
```

Se encuentran las palabras similares a *set\_up.v.04*.

```
[('ocean.n.02', 0.7877715229988098),
('fiery.s.02', 0.7789612412452698),
('entirely.r.02', 0.7541050314903259)]
```

Y se muestra su espacio vectorial (Figura 4).

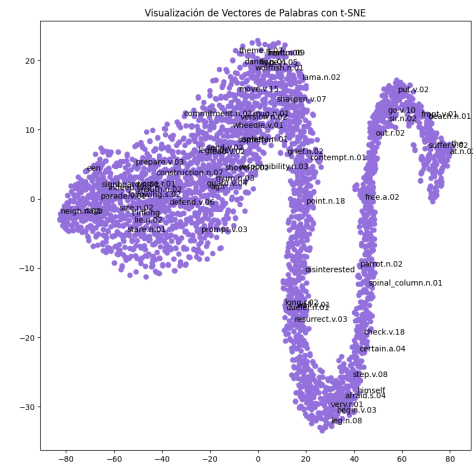


Figura 4. Vectorización de palabras en el archivo pirates.txt

#### IV. CONCLUSIONES

Esta práctica aborda la ambigüedad léxica y semántica a través de la desambiguación de palabras y el modelado de embeddings. La desambiguación de palabras, con el algoritmo Lesk, permite determinar el significado preciso de una palabra en un contexto específico, teniendo en cuenta las palabras circundantes y sus definiciones en un conjunto de datos como WordNet. Este enfoque proporciona una comprensión más profunda del lenguaje y permite una interpretación más precisa de los textos, lo que a su vez mejora la calidad de las representaciones semánticas.

Por otro lado, el modelado de embeddings, en este caso utilizando Word2Vec, permite capturar la estructura semántica de un corpus de texto al asignar representaciones vectoriales a las palabras.

Al combinar la desambiguación de palabras con el modelado de embeddings, se logra una doble ventaja: se captura tanto el significado contextual de las palabras como su representación vectorial en un espacio semántico.

Esta combinación permite una recuperación de información más precisa y relevante al buscar palabras similares o relacionadas en el corpus de texto, esto podría ser relevante llevándolo ya a una aplicación, tales como la búsqueda de información, la clasificación de texto o incluso la traducción automática.

#### REFERENCIAS

- [1] REvista Signos <https://revistasignos.cl/index.php/signos/article/view/116..> Recuperado el 02 de mayo de 2024  
Rago A. y Marcos C. (2009) Técnicas de NLP y WSD Asistiendo Al Desarrollo de Software Orientado a Aspectos. <http://alejandrorgo.com.ar/files/conferences/2009-asai-jaiio.pdf?i=1>  
Torres-Ramos S. (2012) Estudio sobre métodos tipo Lesk usados para la desambiguación de sentidos de palabras. [https://rsc.cic.ipn.mx/2012\\_47/Estudio%20sobre%20metodos%20tipo%20Lesk%20usados%20para%20la%20desambiguacion%20de%20sentidos%20de%20palabras.pdf](https://rsc.cic.ipn.mx/2012_47/Estudio%20sobre%20metodos%20tipo%20Lesk%20usados%20para%20la%20desambiguacion%20de%20sentidos%20de%20palabras.pdf)