

Procesamiento del Lenguaje Natural

Tema 6. Semántica léxica

Índice

Esquema

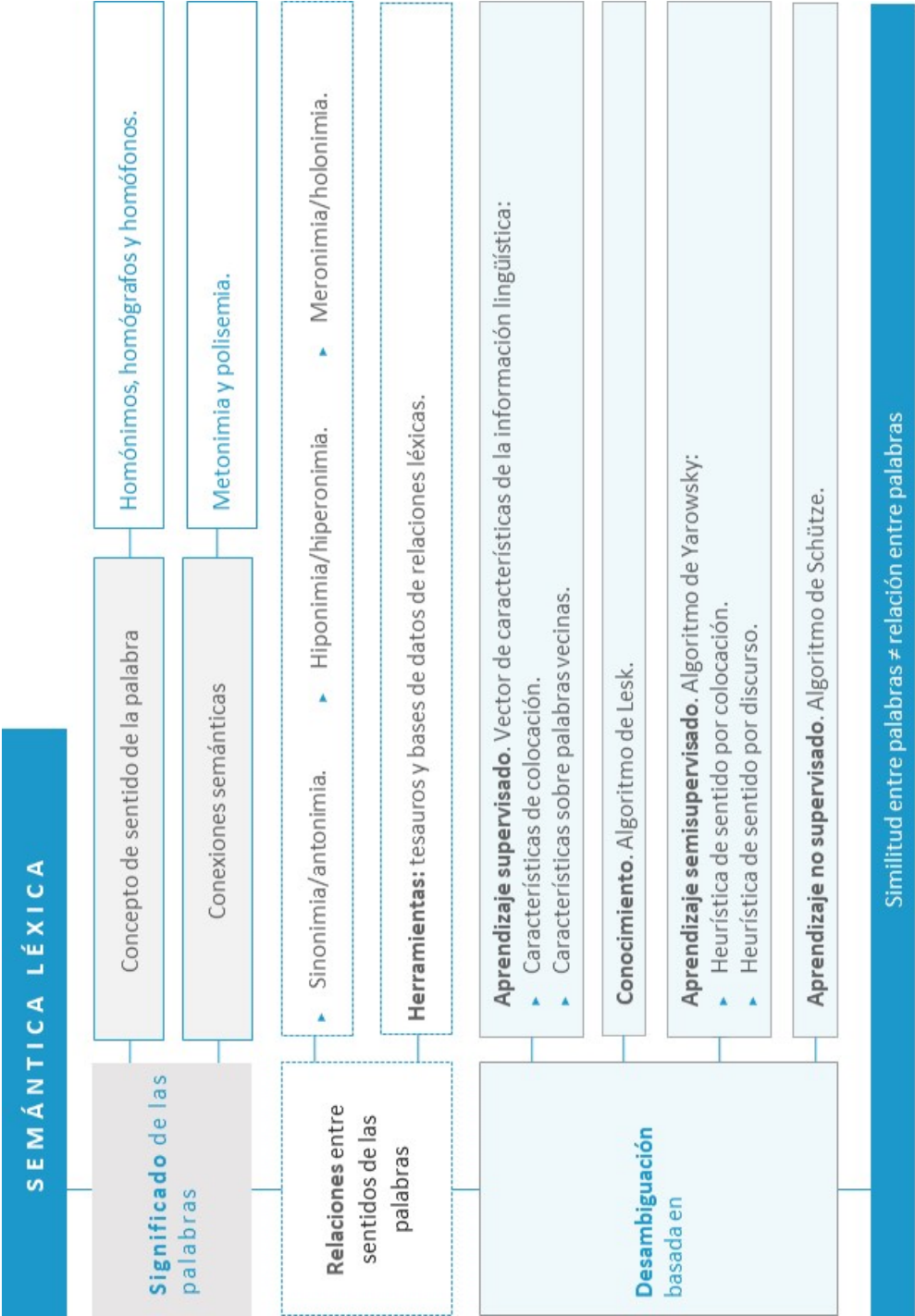
Ideas clave

- 6.1. Introducción y objetivos
- 6.2. Significado de las palabras
- 6.3. Relaciones entre sentidos de las palabras
- 6.4. Desambiguación del sentido de las palabras
- 6.5. Similitud entre palabras
- 6.6. Referencias bibliográficas

A fondo

- Una orientación semántica al análisis de sentimientos
- Word Sense Disambiguation

Test



6.1. Introducción y objetivos

Los objetivos al finalizar el estudio de este tema son:

- ▶ Definir los conceptos relevantes en el ámbito de la semántica léxica, campo de la lingüística que estudia el significado de las palabras.
- ▶ Identificar los diferentes tipos de relaciones entre los sentidos y significados de las palabras: homonimia, metonimia, sinonimia, antonimia, hiponimia, etc.
- ▶ Desambiguar el sentido de las palabras aplicando técnicas de aprendizaje automático, ya sean de aprendizaje supervisado, no supervisado o semisupervisado.
- ▶ Calcular la similitud entre palabras utilizando tesauros.

6.2. Significado de las palabras

La Real Academia Española (RAE, s. f. a) en su diccionario de la lengua española define **semántica léxica** como: «1. f. Rama de la semántica que estudia el significado de las palabras, así como las diversas relaciones de sentido que se establecen entre ellas».

La forma más sencilla de representar el significado de una palabra sería utilizando el **lema**. Tal como se vio anteriormente, **se llama lema a la raíz de la palabra** y, por lo tanto, es la forma gramatical de la palabra que se utiliza en los diccionarios. Por ejemplo, para la palabra *bancos* el lema es la forma «banco».

Un lema puede tener varios significados: en el caso del lema «banco» puede referirse al tipo de asiento donde pueden sentarse varias personas o a la empresa que realiza operaciones financieras, entre otras definiciones que aparecen en el diccionario de la RAE. A cada uno de los significados del lema «banco» se le llama un sentido de la palabra o simplemente sentido.

Un sentido es una representación de uno de los aspectos del significado de una palabra.

Cada uno de los sentidos de una palabra se puede representar añadiendo un superíndice a la forma ortográfica del lema. Entonces, los dos sentidos de la palabra «banco» se representarían como «banco1» y «banco2», respectivamente. Los diferentes sentidos de una palabra (en el ejemplo, el banco como un tipo de asiento («banco1») y como una entidad financiera («banco2») normalmente no tienen ninguna relación a nivel de significado entre sí, por eso se dice que «banco1» y «banco2» son homónimos, término que se aplica a cosas que tienen el mismo nombre.

Si una palabra tiene varios sentidos que no están relacionados entre sí, se dice que los sentidos son homónimos y a la relación entre dos de estos sentidos se le llama relación de homonimia.

Además, en el caso del *banco* como un tipo de asiento («banco1») y como una entidad financiera («banco2») se escriben de la misma forma y, por lo tanto, se dice que estos dos usos son **homógrafos**: según la RAE (s. f. b), una palabra «que tiene la misma grafía que otra».

Existe también otra forma en que dos palabras pueden ser homónimas, sería el caso de palabras que se escriben de forma diferente, pero que se pronuncian igual. Por ejemplo, la palabra *vello* (en referencia al pelo del cuerpo) y la palabra *bello* (hermoso) son **homónimas**: según la RAE (s. f. c), una palabra «que se pronuncia como otra, pero tiene diferente origen o significado muy distante».

Además, *vello* (pelo del cuerpo) y *bello* (hermoso) son dos lemas que suenan igual por lo que se dice que son **homófonos**: según la RAE (s. f. d), una palabra «que suena igual que otra, pero que tiene distinto significado y puede tener distinta grafía».

- ▶ La homofonía es una de las causas de que se cometan errores ortográficos en la vida real y que también afecta el procesamiento del lenguaje natural cuando se realiza el reconocimiento automático del habla.
- ▶ De forma similar, la homografía afecta a la tarea de conversión de texto al habla.
- ▶ La homonimia es la principal causa de diferentes errores que se dan en el procesamiento del lenguaje natural. Para tratarla se aplican técnicas de desambiguación del sentido de la palabra.

La desambiguación del sentido de la palabra es la tarea de determinar qué sentido de una palabra se usa en un contexto particular.

A diferencia de lo que se acaba de exponer, a veces existe una **conexión semántica** entre los sentidos de las palabras. Volviendo al ejemplo de la palabra «banco», puede existir un tercer significado que sería la oficina donde opera la entidad financiera («banco3»). Este último significado sí está relacionado con «banco2» (la entidad financiera); estos dos sentidos, «banco2» y «banco3», tienen una conexión semántica. Sin embargo, «banco3» (la oficina de la entidad financiera) sigue sin tener ninguna relación semántica con «banco1» (el tipo de asiento).

Por otro lado, la oficina bancaria («banco3») coge el nombre de la propia organización, es decir, la entidad financiera («banco2»). Lo mismo pasa con otros ejemplos como la palabra «universidad», que hace referencia a la institución de enseñanza superior, «universidad1», pero también a los edificios de las cátedras y oficinas de la institución universitaria, «universidad2». Este tipo de relación semántica entre los sentidos de las palabras se llama **metonimia**.

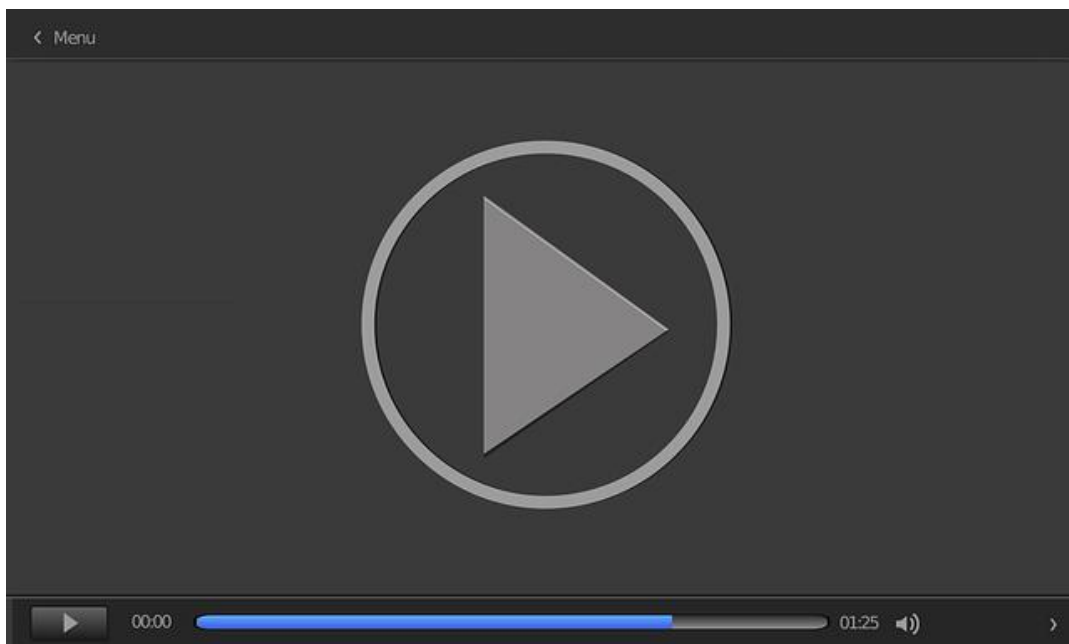
La metonimia es una forma de emplear una palabra en un sentido distinto al que propiamente le corresponde, pero con el que tiene algún tipo de conexión.

Concretamente, la metonimia «consiste en designar algo con el nombre de otra cosa tomando el efecto por la causa o viceversa, el autor por sus obras, el signo por la cosa significada, etc. Por ej., las canas por la vejez; leer a Virgilio, por leer las obras de Virgilio» (RAE, s. f. e). Entonces, el sentido «banco3» (la oficina de la entidad financiera) tiene una **relación de metonimia** con «banco2» (la entidad financiera), así como «universidad1» (institución de enseñanza superior) con la «universidad2» (los edificios de las cátedras y oficinas de la institución universitaria).

La relación semántica de la metonimia es un tipo particular de **polisemia**, el nombre genérico que se utiliza para definir cualquier relación semántica entre dos significados, ya que la palabra polisemia se refiere a la pluralidad de significados.

Si una palabra tiene varios sentidos que están relacionados entre sí, se dice que los sentidos son polisémicos y a la relación entre dos de estos sentidos se le llama relación de polisemia.

En el vídeo *Significado de las palabras* nos centraremos en la semántica léxica, la parte de la lingüística que estudia el significado de las palabras y las relaciones de sentido que se forman en torno a ellas.



06.01. Significado de las palabras

Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=fc89d862-3366-446c-832e-ac6b009c5aae>

Los diccionarios son los repertorios donde se recogen, según un orden determinado, las palabras de una lengua acompañadas de su definición o explicación. Incluyen una descripción detallada y comprensible para un humano de los diferentes sentidos de las palabras. En el procesamiento del lenguaje natural, el formato que siguen los diccionarios tradicionales no es el más útil para obtener y entender los significados de las palabras.

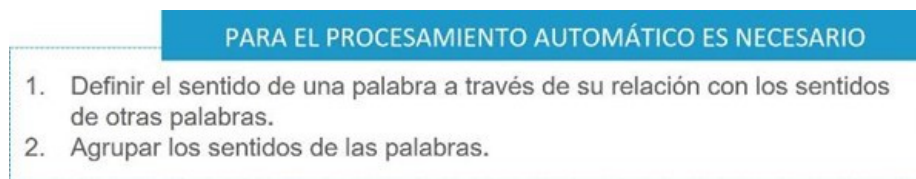
- 
- El diagrama muestra un recuadro con un encabezado azul que dice 'PARA EL PROCESAMIENTO AUTOMÁTICO ES NECESARIO'. Debajo de este encabezado, hay una lista numerada de dos puntos:
1. Definir el sentido de una palabra a través de su relación con los sentidos de otras palabras.
 2. Agrupar los sentidos de las palabras.

Figura 1. Requisitos a la hora de elaborar un repertorio de los sentidos de las palabras en PLN. Fuente: elaboración propia.

Por ejemplo, para el procesamiento del lenguaje natural puede ser útil definir que «rojo» y «azul» son lemas del mismo tipo, concretamente del tipo color. Además de establecer que pertenecen al mismo grupo y son complementarios, lo que indica que algo que es rojo no puede ser azul. También puede ser interesante definir que la sangre es roja y el mar es azul. Si se tiene una base de datos suficientemente exhaustiva con estas relaciones entre los sentidos de las palabras será posible realizar tareas semánticas muy sofisticadas.

6.3. Relaciones entre sentidos de las palabras

Las relaciones más comunes que se dan entre los sentidos de las palabras son la **sinonimia y la antonimia**. También destacan la **hiponimia y la hiperonimia**, y, aunque sea menos común, **la meronimia**.

Cuando los sentidos de dos palabras diferentes y, por extensión, los significados de sus dos lemas son idénticos o casi idénticos, se puede decir que hay una relación de **sinonimia** entre ambos. Por ejemplo, las palabras *empezar* y *comenzar* son sinónimos ya que, si se intercambian en una frase, esta mantiene prácticamente el mismo significado.

Una palabra es un sinónimo de otra si tiene el mismo significado o un significado muy parecido.

La **antonimia** es la **relación contraria**. Los antónimos son palabras que tienen significados opuestos como, por ejemplo, *claro* y *oscuro* o *antes* y *después*. Dos sentidos pueden ser antónimos si definen una oposición binaria o están en extremos opuestos de alguna escala. Este es el caso de *largo* y *corto*, *rápido* y *lento* o *grande* y *pequeño*, porque se encuentran en extremos opuestos de la escala de longitud, de tiempo y de tamaño, respectivamente. Además, se puede definir un grupo de antónimos llamados **reversos**, que describen cambios o movimientos en direcciones opuestas como *abierto* y *cerrado* o *subir* y *bajar*.

Una palabra es un antónimo de otra si expresa una idea opuesta o contraria.

Un sentido es un hipónimo de otro si el primer sentido es más específico que el segundo, es decir, si el primero es una subclase del segundo. Por ejemplo, *coche* es hipónimo de *vehículo*; *perro* lo es a *animal* y *mango* lo es a *fruta* porque el coche es un tipo de vehículo, el perro un tipo de animal y el mango un tipo de fruta.

Una palabra es un hipónimo de otra si su significado incluye el significado de la otra palabra.

La relación contraria a la hiponimia es la **hiperonimia**. Entonces, se dice que *vehículo* es hiperónimo de *coche*, *animal* es hiperónimo de *perro* y *fruta* es hiperónimo de *mango* porque la clase que representa los vehículos incluye como miembros a todos los coches, la que representa a los animales incluye a los perros y la que representa a las frutas incluya a los mangos.

Una palabra es un hiperónimo de otra si su significado está incluido en el significado de la otra palabra.

Formalmente, se define que un sentido

A es hipónimo de un sentido

B si todo lo que es

A también es

B , entonces ser un

A implica ser un

B , lo que en lógica de primer orden se escribiría como:

$$\forall x A(x) \Rightarrow B(x)$$

La hiponimia suele ser una **relación transitiva**, si

A es hipónimo de

B y

B es un hipónimo de

C, entonces

A es un hipónimo de

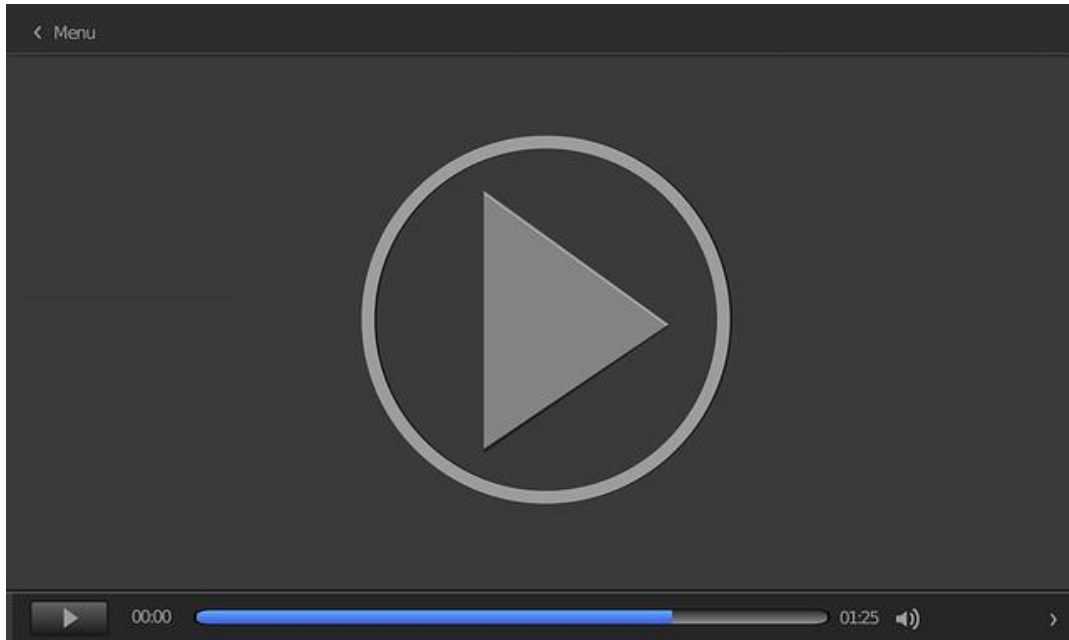
C.

En las ontologías, representaciones semánticas basadas en lógica descriptiva, se modelan las relaciones de hiponimia e hiperonimia a través de la **jerarquía IS-A**. La expresión *A IS-A B* indica que un sentido *A* es hipónimo de un sentido *B*, o que un sentido *B* es hiperónimo de un sentido *A*.

Una última relación destacada es la **meronimia** que define una relación del tipo parte-todo entre los sentidos de las palabras. Por ejemplo, una rueda es una parte de un coche, por lo que se dice que *rueda* es merónimo de *coche* o, al contrario, que *coche* es **holónimo** de *rueda*.

Una palabra es un merónimo de otra si su significado mantiene con el significado de la otra palabra una relación de la parte respecto al todo.

En el vídeo *Relaciones entre sentidos de las palabras* se estudiarán las diferentes relaciones entre los sentidos que pueden tener las palabras.



06.02. Relaciones entre sentidos de las palabras

Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=cfc679a6-e49b-4912-b48d-ac6b009db75d>

Para poder realizar tareas de procesamiento del lenguaje natural es necesario recoger las relaciones entre los sentidos de las palabras de un diccionario. De hecho, en literatura se utiliza el término **tesauro**, *thesaurus*, *thesauri* o tesoro para referirse a los diccionarios que contienen una lista de palabras con sus sinónimos y sus antónimos. En lingüística, los términos que conforman un tesauro se relacionan entre sí para mostrar las relaciones entre significados.

Las llamadas **bases de datos de relaciones léxicas** contienen un conjunto de lemas, cada uno de los cuales está anotado con el posible conjunto de sentidos de la palabra. Cada uno de los sentidos contiene, además de su definición en un formato tipo glosario y una lista de sinónimos. No es obligatorio que las bases de datos de relaciones léxicas contengan la pronunciación los lemas, sin embargo, es imprescindible que estas contengan detalles de las relaciones entre lemas.

La Figura 2 muestra un ejemplo de algunas relaciones entre sentidos en WordNet (Fellbaum, 1998), la que es la principal base de datos léxica para procesamiento del lenguaje natural en inglés.

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	<i>breakfast</i> ¹ → <i>meal</i> ¹
Hyponym	Subordinate	From concepts to subtypes	<i>meal</i> ¹ → <i>lunch</i> ¹
Instance Hypernym	Instance	From instances to their concepts	<i>Austen</i> ¹ → <i>author</i> ¹
Instance Hyponym	Has-Instance	From concepts to concept instances	<i>composer</i> ¹ → <i>Bach</i> ¹
Member Meronym	Has-Member	From groups to their members	<i>faculty</i> ² → <i>professor</i> ¹
Member Holonym	Member-Of	From members to their groups	<i>copilot</i> ¹ → <i>crew</i> ¹
Part Meronym	Has-Part	From wholes to parts	<i>table</i> ² → <i>leg</i> ³
Part Holonym	Part-Of	From parts to wholes	<i>course</i> ⁷ → <i>meal</i> ¹
Substance Meronym		From substances to their subparts	<i>water</i> ¹ → <i>oxygen</i> ¹
Substance Holonym		From parts of substances to wholes	<i>gin</i> ¹ → <i>martini</i> ¹
Antonym		Semantic opposition between lemmas	<i>leader</i> ¹ ↔ <i>follower</i> ¹
Derivationally Related Form		Lemmas w/same morphological root	<i>destruction</i> ¹ ↔ <i>destroy</i> ¹

Figura 2. Relaciones entre sentidos en WordNet. Fuente: Jurafsky y Martin, 2009.

6.4. Desambiguación del sentido de las palabras

Los métodos para el análisis semántico composicional que se han estudiado en el tema anterior no tienen en cuenta que una palabra puede tener más de un significado. De hecho, esos métodos obvian la ambigüedad léxica que es una realidad existente y es un problema importante en el procesamiento del lenguaje natural.

La **desambiguación del sentido de las palabras** es la tarea de seleccionar el sentido correcto para una palabra. Los algoritmos de desambiguación del sentido toman como entrada una palabra en su contexto y una lista de posibles significados de esa palabra y devuelven como salida el sentido correcto para ese uso concreto de la palabra.

Existen diferentes opciones para implementar un algoritmo de desambiguación del sentido de las palabras. La **primera opción** se basa en aplicar técnicas de **aprendizaje automático supervisado** para aprender un modelo clasificador que permita desambiguar las palabras. Estos algoritmos de desambiguación basados en aprendizaje supervisado requieren tener un corpus de palabras etiquetadas con sus sentidos correctos para poder entrenar el clasificador. Tener acceso a este tipo de datos etiquetados puede ser complejo y es por eso por lo que, aunque los algoritmos de desambiguación basados en aprendizaje supervisado sean los que proporcionan mejores resultados, a veces no se utilizan por el elevado coste asociado a la obtención de los datos etiquetados.

Si no se dispone de un corpus etiquetado, una alternativa es utilizar diccionarios, tesauros u otras bases de conocimiento para realizar un entrenamiento indirecto, aplicando algoritmos de aprendizaje supervisado débil.

Este tipo de métodos de desambiguación se llaman algoritmos de desambiguación basados en conocimiento y no utilizan textos que hayan sido etiquetados manualmente, sino que utilizan grandes bases de conocimiento.

Otra opción a los algoritmos de desambiguación basados en aprendizaje supervisado y a los basados en conocimiento es aplicar técnicas de **aprendizaje semisupervisado** o *bootstrapping*. Estos algoritmos de desambiguación basados en aprendizaje semisupervisado no requieren de grandes recursos lingüísticos generados a mano, es decir, ni de un gran conjunto de datos de entrenamiento ni de un gran diccionario, sino que, para funcionar es suficiente con tener un pequeño conjunto de datos de entrenamiento etiquetados a mano.

Una última alternativa, que pretende evitar la tarea compleja y costosa de construir un corpus de palabras etiquetadas con los sentidos para la desambiguación de las palabras, se basa en técnicas de **aprendizaje no supervisado**.

Los algoritmos basados en este tipo de aprendizaje realizan la desambiguación sin utilizar definiciones de los sentidos de las palabras por humanos.

El conjunto de sentidos de cada palabra se crea automáticamente a partir de las instancias de la palabra disponibles en los datos de entrenamiento. Esta tarea se llama también **inducción del sentido de las palabras**.

Desambiguación basada en aprendizaje supervisado

Los algoritmos de desambiguación basados en el aprendizaje automático supervisado **utilizan un conjunto de instancias etiquetadas** para entrenar un clasificador. Una vez entrenado, este sirve para predecir el mejor sentido de las palabras ambiguas. Por lo tanto, el resultado del entrenamiento es un modelo clasificador capaz de asignar etiquetas de sentido a las palabras no etiquetadas que aparecen en un contexto determinado.

En una **primera opción de implementación** de los algoritmos de aprendizaje supervisado para aprender desambiguaciones, se parte de un conjunto de palabras ambiguas y se empieza preseleccionando un subconjunto de sentidos posibles para estas. Así, para cada palabra, se selecciona un número de instancias de la aparición de esa palabra en un corpus. Estas instancias se etiquetan a mano con el sentido correcto, según el contexto en el que aparecen en el corpus. Una vez se tienen todos los ejemplos etiquetados, estos se utilizan para entrenar el clasificador.

Como en este caso se trabaja con un conjunto reducido de palabras y el conjunto reducido de sentidos, se dice que el algoritmo de aprendizaje supervisado se entrena con una **muestra léxica**, con lo que aprende un modelo clasificador que permite desambiguar una única palabra o, como máximo, un conjunto reducido de palabras concretas, algo que puede ser poco práctico.

En una **segunda opción de implementación** de los algoritmos de aprendizaje supervisado, se utiliza un corpus de palabras ya etiquetadas con su sentido para aprender la desambiguación de un texto entero. Es decir, que se aprende a desambiguar todas las palabras del texto y no solo algunas palabras concretas, como era el caso de utilizar una muestra léxica.

El algoritmo de desambiguación que utiliza un **lexicón** para entrenar requiere de un conjunto muy grande de etiquetas porque cada lema tiene su propio set de etiquetas. Gestionar este conjunto enorme puede representar un problema y reducir la aplicabilidad de estas técnicas de aprendizaje supervisado basadas en un corpus etiquetado.

Los algoritmos de desambiguación basados en el aprendizaje supervisado ya sean los que para entrenar utilizan una muestra léxica etiquetada a mano o los que utilizan un corpus ya etiquetado, tienen que extraer características de texto y, a partir de ellas, asignar etiquetas con el sentido correcto.

Entonces, el primer paso en los algoritmos de aprendizaje supervisado es extraer características que sean predictivas de los sentidos de las palabras.

Los algoritmos modernos de desambiguación del sentido extraen información del contexto que rodea la palabra a desambiguar (Weaver, 1955). Para ello se utiliza una ventana que incluye la palabra ambigua y las palabras anteriores y posteriores, también llamadas palabras de contexto, y se extrae un **vector de características de la información lingüística** de las palabras de contexto contenidas en la ventana. Dicho vector consta de valores numéricos o nominales y contiene dos tipos de características:

- ▶ Características de colocación.
- ▶ Características sobre las palabras vecinas.

Características de colocación

Codifican información sobre la posición de las palabras de contexto y su relación con la palabra ambigua. Algunas características típicas que se pueden extraer de las palabras de contexto incluyen la palabra en sí, la raíz de la palabra y su categoría gramatical o morfosintáctica.

Así, estas características codifican información léxica y gramatical que a menudo puede ayudar a identificar con precisión el sentido de la palabra.

Por ejemplo, un vector de características de colocación extraído de una ventana con dos palabras a la derecha y a la izquierda de la palabra objetivo, donde las características son las palabras, sus categorías gramaticales y los pares de palabras, tendría la siguiente forma:

$$[w_{i-2}, POS_{i-2}, w_{i-1}, POS_{i-1}, w_{i+1}, POS_{i+1}, w_{i+2}, POS_{i+2}, w_{i-2}^{i-1}, w_i^{i+1}]$$

Donde:

- ▶ w_{i-2} es la palabra dos posiciones a la izquierda de la palabra ambigua.
- ▶ POS_{i-2} es la categoría gramatical de la palabra w_{i-2} .
- ▶ w_{i-2}^{i-1} es el par de palabras w_{i-2} y w_{i-1} .

Ejemplo ilustrativo 1

Vector de características de colocación para desambiguar la palabra «*bass*» (bajo) en la oración:

- An electric guitar and bass player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps (del corpus WSJ).

Se define una ventana con dos palabras a la derecha y a la izquierda de la palabra «*bass*»:

<i>guitar</i>	<i>and</i>	<i>bass</i>	<i>player</i>	<i>stand</i>
w_{i-2}	w_{i-1}		w_{i+1}	w_{i+2}

Tabla 1. Ventana con la palabra «*bass*» como palabra objetivo. Fuente: elaboración propia.

Para calcular algunas de las características se realiza el análisis morfosintáctico y se identifican las siguientes categorías gramaticales de las palabras de contexto:

<i>guitar</i>	<i>and</i>	<i>bass</i>	<i>player</i>	<i>stand</i>
NN	CC		NN	VB

Tabla 2. Categorías gramaticales de las palabras contexto. Fuente: elaboración propia.

Como características se utilizan las propias palabras de contexto, sus categorías gramaticales y los pares de palabras posicionadas antes y después de la palabra «*bass*». El vector de características de colocación sería:

[guitar, NN, and, CC, player, NN, stand, VB, and guitar, player stand]

Características sobre las palabras vecinas

El segundo tipo de características codifican información sobre la llamada *bag-of-words* (bolsa de palabras), es decir, sobre un conjunto de palabras sin considerar su orden y que rodean a la palabra ambigua. Las características que se extraen de una bolsa de palabras son eficaces para capturar el **tema general del discurso** en el que ocurre la palabra ambigua. Además, el tema del discurso tiende a identificar a su vez los sentidos de las palabras que son específicas en un cierto contexto.

En el enfoque más simple de bolsa de palabras, el contexto de una palabra ambigua se representa como un vector de características donde cada característica binaria indica si una palabra de un vocabulario aparece o no en el contexto. El vocabulario lo conforman un subconjunto de palabras útiles o de uso frecuente y preseleccionadas del corpus de entrenamiento. La región de contexto se define como una ventana de tamaño fijo donde la palabra ambigua se encuentra en el centro y que contiene un número generalmente pequeño de palabras situadas a la izquierda y a la derecha de la palabra ambigua.

Ejemplo ilustrativo 2

Vector de características sobre las palabras vecinas para desambiguar la palabra *bass* (bajo) en la oración:

An electric guitar and bass player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps (del corpus WSJ).

Se define la siguiente *bag-of-words* (bolsa de palabras) con las doce palabras que aparecen más frecuentemente en frases donde se encuentra la palabra *bass* en el corpus WSJ:

[fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band]

Se utiliza una ventana de tamaño nueve: ventana que incluye cuatro palabras a la izquierda y cuatro a la derecha de la palabra ambigua. Por lo tanto, las palabras que quedarían dentro de la ventana son:

An electric guitar and bass player stand off to

En la ventana aparecen las palabras «*guitar*» y «*player*» y no aparecen ninguna de las otras palabras que forman parte de la bolsa (*bag-of-words*).

Entonces el vector de características sobre las palabras vecinas sería:

[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0]

Cualquier conjunto de datos de entrenamiento puede servir para extraer tanto características de colocación como características sobre las palabras vecinas y entrenar un clasificador de sentidos de las palabras con un algoritmo de aprendizaje automático supervisado. Sin embargo, el buen funcionamiento del clasificador va a depender de la cantidad de datos de entrenamiento con los que se cuente, es decir, de la disponibilidad de datos etiquetados con los sentidos. Debido a este problema, algunos investigadores han empezado a usar la Wikipedia como fuente de datos de entrenamiento (Mihalcea, 2007) (Ponzetto y Navigli, 2010).

Desambiguación basada en conocimiento

Los algoritmos de desambiguación que utilizan bases de conocimiento ya sean diccionarios o tesauros, para realizar un entrenamiento indirecto aplican algoritmos de aprendizaje supervisado débil.

- ▶ En el caso de utilizar diccionarios, se aplica el algoritmo de Lesk, que se presenta a continuación.
- ▶ En el caso de utilizar un tesoro, se aplican métodos basados en gráficos (Agirre, López de Lacalle y Soroa, 2014) (Navigli y Lapata, 2010).

El llamado **algoritmo de Lesk** se refiere en realidad a una familia de algoritmos que seleccionan el sentido de la palabra para el cual su definición en el diccionario comparte la mayor cantidad de palabras con los vecinos de la palabra ambigua.

La versión más simple del algoritmo de Lesk llamada **algoritmo de Lesk simplificado** (Kilgarriff y Rosenzweig, 2000) compara la firma de la palabra ambigua (*signature*), que se corresponde con su definición en el diccionario, con las palabras de contexto (*context*), es decir, con las palabras vecinas a la palabra ambigua. El pseudocódigo se presenta en la Figura 3, donde la función COMPUTEOVERLAP devuelve el número de palabras que tienen en común los conjuntos de palabras *signature* y *context*.

```
function SIMPLIFIED LESK(word, sentence) returns best sense of word

  best-sense ← most frequent sense for word
  max-overlap ← 0
  context ← set of words in sentence
  for each sense in senses of word do
    signature ← set of words in the gloss and examples of sense
    overlap ← COMPUTEOVERLAP(signature, context)
    if overlap > max-overlap then
      max-overlap ← overlap
      best-sense ← sense
  end
  return(best-sense)
```

Figura 3. Pseudocódigo del algoritmo de Lesk Simplificado. Fuente: Jurafsky y Martin, 2009.

Ejemplo ilustrativo 3

Funcionamiento del algoritmo de Lesk simplificado para desambiguar la palabra «*bank*» (banco) en la oración

- ▶ The bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

Del diccionario WordNet se obtienen las siguientes definiciones para la palabra «*bank*»:

bank ¹	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank ²	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

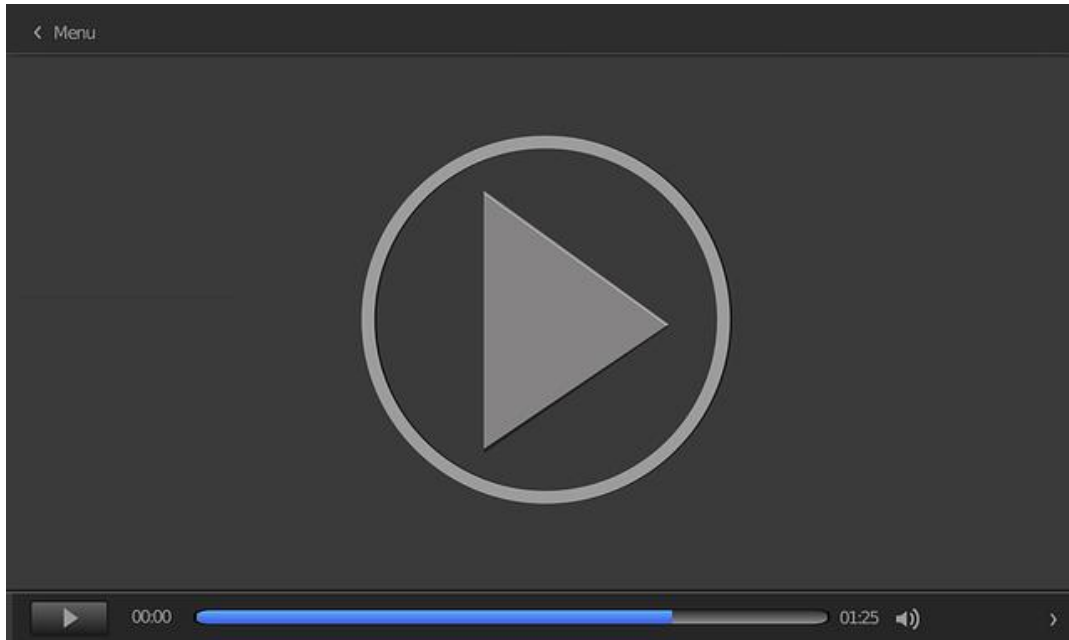
Figura 4. Definiciones de la palabra «*bank*» en WordNet. Fuente: WordNet, s. f.

Se observa que en la definición y los ejemplos del sentido «bank1» aparecen dos palabras, «*deposits*» y «*mortgage*», que también aparecen en el contexto de la palabra a desambiguar, es decir, en la frase bajo análisis.

Por el contrario, ni en la definición ni en los ejemplos del sentido «bank2» aparecen palabras relevantes que también aparezcan en el contexto de la palabra a desambiguar. Es importante notar que los determinantes (por ejemplo, «*the*»), preposiciones, conjunciones y pronombres no se tienen en cuenta por no aportar información relevante.

Por lo tanto, el algoritmo de Lesk simplificado escogería el sentido «bank1», que hace referencia a la entidad financiera, como sentido correcto para la palabra «*bank*» en la frase que se ha analizado.

En el vídeo *Algoritmo de Lesk simplificado* se verá cómo busca desambiguar el sentido de una palabra en una oración utilizando el contexto de esta.



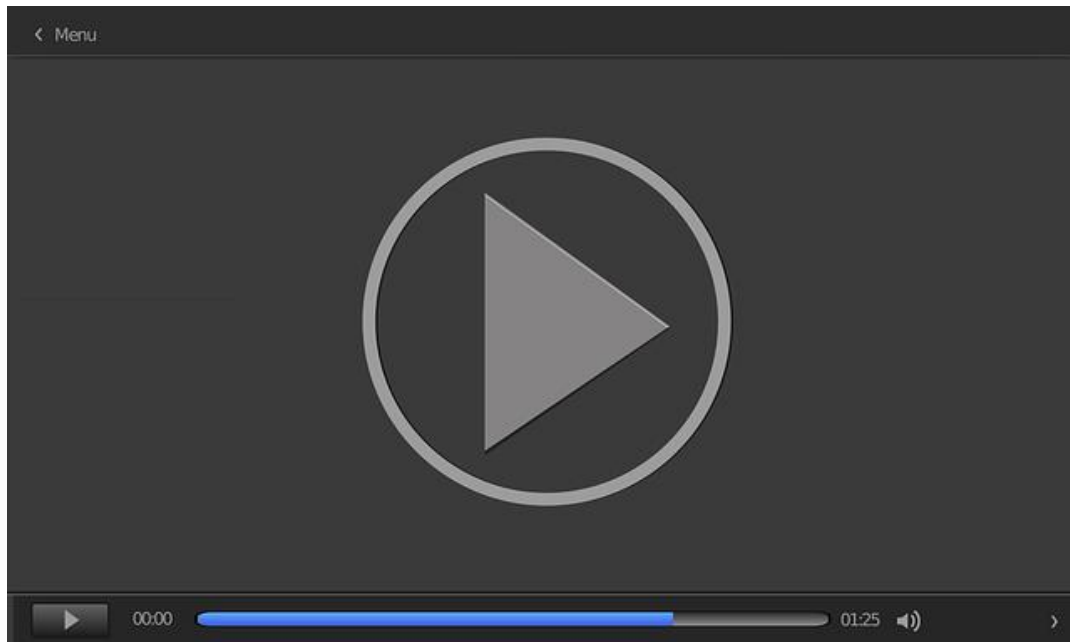
06.03. Algoritmo de Lesk simplificado

Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=5c4e6073-a936-4215-a5eb-ac6b00a1e80b>

La **versión original del algoritmo de Lesk** (Lesk, 1986) utiliza para el entrenamiento una experiencia incluso más indirecta que la versión simple del algoritmo. De hecho, la versión original lo que hace es comparar la firma de la palabra ambigua con las firmas de cada una de las palabras del contexto.

En el vídeo *Versión original del algoritmo de Lesk* se estudiará un ejemplo del funcionamiento de este algoritmo.



06.04. Versión original del algoritmo de Lesk

Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=2ea553f6-3f27-4057-893e-ac6b00a2cb8f>

El principal problema de la versión original y la versión simplificada del algoritmo de Lesk es que las entradas del diccionario para las palabras ambiguas son cortas y puede que no se dé la oportunidad de que se solapen con las palabras del contexto (Lesk, 1986).

La solución es añadir todas las palabras en las oraciones del corpus etiquetado para un sentido de la palabra al cómputo de la firma de ese sentido. Esta versión del algoritmo se llama **Corpus Lesk** y es la que proporciona mejores resultados (Kilgarriff y Rosenzweig, 2000) (Vasilescu, Langlais y Lapalme, 2004). Además, el algoritmo de Corpus Lesk lo que hace es **aplicar un peso a cada palabra coincidente en lugar de contar el número de palabras**. Aplicar un peso sirve para dar menos relevancia a las palabras poco importantes, pero que se repiten mucho como, por ejemplo, **los determinantes, preposiciones, conjunciones y pronombres**.

Para calcular los pesos en el algoritmo de Corpus Lesk se utiliza una medida llamada ***inverse document frequency (IDF)*** y que se define para la palabra i según la siguiente fórmula:

$$idf_i = \log \frac{Ndoc}{nd_i}$$

Donde:

- ▶ $Ndoc$ es el número total de «documentos», es decir, definiciones y ejemplos.
- ▶ nd_i es el número de estos documentos que contienen la palabra i .

Se puede combinar el algoritmo de Lesk y los enfoques de aprendizaje automático supervisado añadiendo nuevas características similares a las computadas con el algoritmo de Lesk a la bolsa de palabras. Por ejemplo, las definiciones y las frases de ejemplo para el sentido de una palabra del diccionario se podrían usar en la bolsa de palabras utilizada en el aprendizaje supervisado (Yuret, 2004).

Desambiguación basada en aprendizaje semisupervisado

Los algoritmos de desambiguación basados en aprendizaje semisupervisado o *bootstrapping* no requieren de grandes recursos lingüísticos generados a mano, sino que para funcionar solo necesitan un pequeño conjunto de datos de entrenamiento etiquetados a mano que se irá ampliando durante el proceso de aprendizaje.

Uno de estos algoritmos más conocidos es el **algoritmo de Yarowsky**, que permite **aprender un clasificador para una palabra ambigua** (Yarowsky, 1995). El algoritmo de Yarowsky parte de un conjunto pequeño de datos de entrenamiento etiquetados (Λ_0) que contiene instancias para cada sentido de la palabra y un corpus no etiquetado mucho más grande (V_0).

En la primera iteración, el algoritmo de Yarowsky entrena un clasificador utilizando las instancias etiquetadas de Λ_0 y, a continuación, utiliza este clasificador para etiquetar el corpus no etiquetado V_0 . El algoritmo selecciona entonces los ejemplos de V_0 para los que tiene mayor confianza en la clasificación, los elimina de V_0 (pasando a llamarse ahora el conjunto de datos no etiquetados V_1) y los añade al conjunto de datos de entrenamiento (que ahora pasa a llamarse Λ_1).

En la siguiente iteración, el algoritmo entrena un nuevo clasificador con Λ_1 , utiliza el clasificador para etiquetar V_1 y extrae un nuevo conjunto de datos de entrenamiento Λ_2 , y así sucesivamente. Con cada iteración, el corpus de los datos de entrenamiento Λ crece y el corpus de datos sin etiquetar V disminuye. El proceso se repite hasta que se alcanza una tasa de error suficientemente baja o hasta que no quedan más ejemplos no etiquetados.

El conjunto inicial de datos de entrenamiento Λ_0 se puede etiquetar a mano, escogiendo un conjunto de instancias aleatorias del corpus no etiquetado V_0 (Hearst, 1991) o se puede hacer uso de la ayuda de una heurística para seleccionar las mejores instancias (Yarowsky, 1995).

El algoritmo original de Yarowsky propone dos posibles heurísticas: la de un sentido por colocación y la de un sentido por discurso.

Heurística de un sentido por colocación

Se basa en la idea de que ciertas palabras o frases muy relacionadas con el sentido analizado no tienden a ocurrir con los otros sentidos. Entonces se define el conjunto de datos de entrenamiento escogiendo una única colocación para cada uno de los sentidos.

Heurística de un sentido por discurso

Se basa en la idea de que una palabra ambigua que aparece varias veces en un texto o discurso aparece a menudo con el mismo sentido (Gale, Church y Yarowsky, 1992). Esta heurística proporciona mejores resultados para los casos de homonimia que para los de polisemia (Krovetz, 1998) y es muy importante en el procesamiento del lenguaje porque muchas veces las tareas de desambiguación mejoran si se da un sesgo y se resuelve la ambigüedad de la misma manera en un mismo discurso.

Desambiguación basada en aprendizaje no supervisado

Los algoritmos de desambiguación basados en aprendizaje no supervisado aprenden los sentidos de las palabras a partir de un conjunto de datos de entrenamiento sin necesidad de disponer de definiciones para los sentidos que sean entendibles por parte de las personas. Estos algoritmos de desambiguación utilizan algún método estándar de agrupamiento, también llamado *clustering*, y una medida de distancia para determinar la similitud entre clústeres. De hecho, el método más usado en tareas lingüísticas es el **agrupamiento aglomerativo** que va fusionando sucesivamente los clústeres más similares.

El **algoritmo de Schütze** (1992) (1998) representa cada palabra como un vector de contexto para una bolsa de palabras y, entonces, entrena el algoritmo siguiendo tres pasos:

- ▶ Para cada aparición w_i de la palabra w en un corpus, se calcula un vector contexto c .
- ▶ Se utiliza un algoritmo de agrupamiento para agrupar los vectores de contexto c en un número predefinido de clústeres. De hecho, cada uno de los clústeres define un sentido de la palabra w .
- ▶ Se calcula el centroide de cada clúster. Cada centroide s_j es un vector que representa un sentido de la palabra w .

Dado que este es un algoritmo de clasificación no supervisado, no se conoce el nombre de cada uno de los sentidos de la palabra w ; por lo que simplemente se les llama «el sentido j de la palabra w ».

Para eliminar la ambigüedad de una instancia particular t de la palabra w utilizando el algoritmo de Schütze se aplican otros tres pasos:

- ▶ Se calcula el vector contexto c para la instancia t .
- ▶ Se recuperan todos los vectores de sentido s_j de la palabra w .
- ▶ Se asigna t al sentido representado por el vector de sentido s_j que está más cerca t .

6.5. Similitud entre palabras

Una de las relaciones entre palabras que más se usa en el procesamiento del lenguaje natural es la **sinonimia**. La sinonimia es una **relación binaria entre dos palabras**: las palabras son sinónimas o no lo son. Sin embargo, se puede utilizar una métrica más relajada que permita calcular la similitud entre palabras, llamada **distancia semántica**.

Dos palabras son más similares si comparten más características de su significado, es decir, si son casi sinónimos. Por el contrario, dos palabras son menos similares o tienen mayor distancia semántica si comparten menos elementos de su significado.

Aunque se hable de similitud entre palabras, realmente la similitud debe medirse entre los sentidos de las palabras. Además, se debe distinguir entre **similitud entre palabras** y **relación entre palabras**. Dos palabras son similares si son casi sinónimos y una puede sustituir a la otra en un contexto dado, sin embargo, algunas palabras pueden estar relacionadas sin ser similares.

Por ejemplo, las palabras *coche* y *gasolina* están estrechamente relacionadas, pero no son similares; mientras que las palabras *coche* y *bicicleta*, además de estar relacionadas, son más similares. De hecho, los antónimos son palabras que están relacionadas, pero que no son similares en absoluto.

Es importante destacar que muchos de los algoritmos que calculan la similitud entre palabras, lo que en realidad hacen es utilizar una medida de relación entre palabras y no únicamente de similitud, aunque típicamente siguen llamándose medidas de similitud.

Para medir la similitud entre palabras o, mejor dicho, la relación entre sentidos de las palabras, existen dos tipos de algoritmos.

- ▶ Los primeros calculan la similitud entre palabras utilizando la estructura de un tesoro y se estudian en este tema.
- ▶ Los segundos calculan la similitud entre palabras aplicando métodos de distribución y encontrando directamente palabras que tienen distribuciones similares en un corpus.

Similitud entre palabras basada en tesauros

Los algoritmos que calculan la similitud entre palabras utilizando tesauros miden la distancia entre dos sentidos en un tesoro en línea como, por ejemplo, en WordNet. Estos algoritmos utilizan la estructura jerárquica del tesoro para definir la similitud entre palabras.

En principio, se podría medir la similitud utilizando cualquier **conocimiento** existente en el tesoro, como podría ser la meronimia o el glosario. Sin embargo, en la práctica, los algoritmos de similitud entre palabras basados en tesauros utilizan en general solo la jerarquía de relaciones de hiperonimia/hiponimia.

Los algoritmos más simples que utilizan tesauros se basan en la hipótesis de que los sentidos de las palabras son más similares si existe un camino más corto entre ellos en la representación gráfica del tesoro (Quillian, 1969). Entonces, un sentido es lo más parecido a sí mismo, luego a sus padres o hermanos y luego, es menos similar a los sentidos de las palabras que están lejos en el gráfico.

Por lo que se define la **longitud del camino entre dos sentidos**, representados por los nodos

c_1 y

c_2 , del gráfico del tesoro como

$pathlen(c_1, c_2)$ y se calcula añadiendo una unidad al número de aristas existentes

en el camino más corto entre los nodos

c_1 y

c_2 .

Entonces, se define la medida de **similitud basada en la longitud del camino** como:

$$path(c_1, c_2) = \frac{1}{pathlen(c_1, c_2)}$$

Sin embargo, en la mayoría de las aplicaciones no se dispone de datos de entrada con los sentidos etiquetados, por lo que es necesario que el algoritmo de similitud pueda proporcionar la similitud entre las palabras en lugar de entre sentidos. Se puede aproximar dicha similitud entre palabras (requeriría de un proceso de desambiguación de sentidos) utilizando el par de sentidos de las propias palabras, que son las que dan el máximo valor de similitud de los sentidos (Resnik, 1995). Por lo que formalmente se define la **similitud entre palabras** a partir de la similitud entre sentidos de la siguiente forma:

$$wordsim(w_1, w_2) = \max_{\substack{c_1 \in senses(w_1) \\ c_2 \in senses(w_2)}} (c_1, c_2)$$

El algoritmo que mide la similitud basada en la longitud del camino hace la suposición implícita de que cada eslabón, en la representación gráfica del tesoro, presenta una distancia uniforme. En la práctica, este supuesto no es apropiado porque las relaciones en un nivel más profundo de la jerarquía representan a menudo una distancia más cercana, mientras que otras relaciones más arriba en la jerarquía representan una distancia más amplia. La solución pasa por normalizar las distancias en función a la profundidad de la jerarquía (Wu y Palmer, 1994) o asociar una distancia diferente a cada una de las aristas del tesoro.

Otros algoritmos que utilizan tesauros para calcular la similitud entre palabras se basan en la estructura del tesoro, pero también incluyen información probabilística

derivada de un corpus. Son los llamados algoritmos de **similitud entre palabras basados en la cantidad de información** y utilizan medidas de teoría de la información para extraer información del corpus.

Según Resnik (1995), se define $P(c)$ como la probabilidad de que una palabra seleccionada al azar en un corpus sea una instancia del concepto c , es decir, una variable aleatoria de las palabras asociadas con cada concepto. Cualquier palabra del tesoro es un descendiente del concepto de raíz, llamado *root*, lo que implica que $P(\text{root}) = 1$. Intuitivamente, cuanto más bajo se encuentre un concepto en la jerarquía, menor será su probabilidad.

Entonces, para calcular las probabilidades, se hace un recuento del corpus y cada palabra en el corpus cuenta como una ocurrencia de cada concepto que sea su ancestro.

Esto formalmente se define como:

$$P(c) = \frac{\sum_{w \in \text{words}(c)} \text{count}(w)}{N}$$

Donde:

- ▶ $\text{words}(c)$ es el conjunto de palabras descendientes del concepto c .
- ▶ N es el número total de palabras en el corpus que también están presentes en el tesoro.

A partir de la teoría de la información, se define la **cantidad de información** (IC) de un concepto c como:

$$IC(c) = -\log P(c)$$

Además, a partir de la teoría de grafos se utiliza la definición de **ancestro común más bajo** de dos conceptos (LCS, lowest common subsumer en inglés). El

$LCS(c_1, c_2)$ es el nodo más bajo en la jerarquía que tiene como descendientes a los conceptos

c_1 y

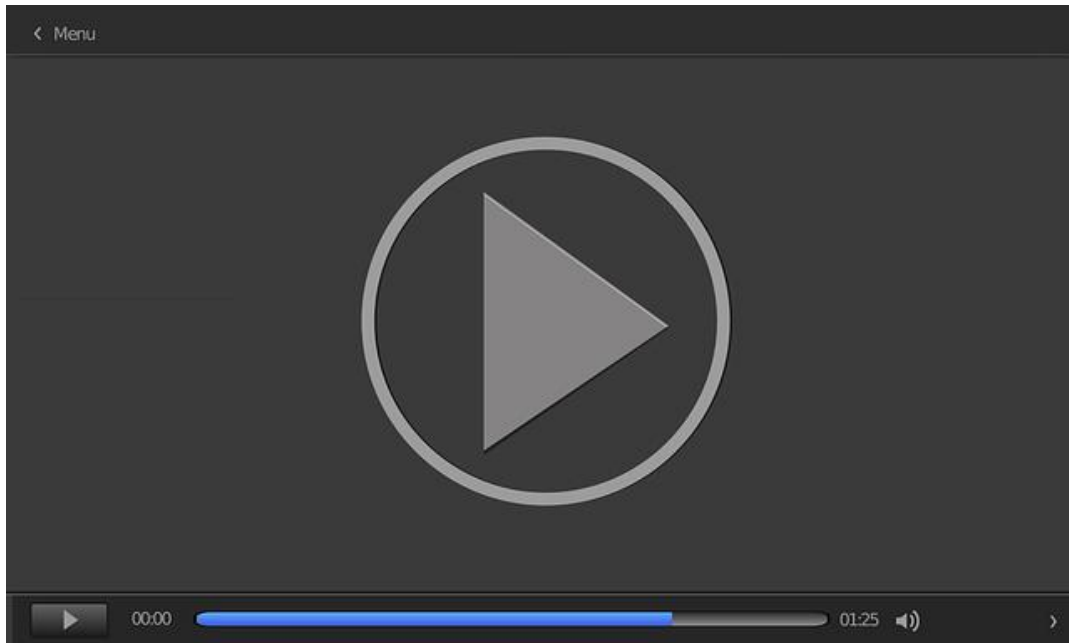
c_2 , es decir que es el ancestro (padre, abuelo, etc.) más cercano de ambos conceptos.

Existen diferentes medidas de similitud entre palabras calculadas a partir de la cantidad de información de un nodo. De hecho, la similitud entre dos palabras está relacionada con la información mutua entre las palabras, cuanto más información compartan las dos palabras, más similares van a ser. Resnik (1995) propone estimar la información mutua entre dos palabras como la cantidad de información del ancestro común más bajo de los dos nodos que representan las palabras. Por lo tanto, la **medida de similitud de Resnik** se calcula con la siguiente fórmula:

$$resnik(c_1, c_2) = -\log P(LCS(c_1, c_2))$$

Otras medidas de similitud que amplían las ideas de Resnik son la medida de similitud de Lin (Lin, 1998) y la distancia de Jiang-Conrath (Jiang y Conrath, 1997).

En el vídeo *Cálculo de la similitud entre palabras utilizando la estructura jerárquica de un tesoro* se explicará cómo se calcula la similitud entre palabras utilizando la estructura jerárquica de un tesoro.



06.05. Cálculo de la similitud entre palabras utilizando la estructura jerárquica de un tesoro

Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=a68ea15c-2a25-4fb0-8a96-ac6b00a3e848>

6.6. Referencias bibliográficas

Agirre, E., López de Lacalle, O. y Soroa, A. (2014). *Random walks for knowledge-based word sense disambiguation*. *Computational Linguistics*, 40(1), 57-84.

Fellbaum, C. (Ed.). (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Gale, W. A., Church, K. W. y Yarowsky, D. (1992). *One sense per discourse*. En *Proceedings DARPA Speech and Natural Language Workshop* (pp. 233-237). New Jersey, Estados Unidos.

Hearst, M. A. (1991). Noun homograph disambiguation. *Proceedings of the 7th Conference of the University of Waterloo Centre for the New OED and Text Research*, 1-19.

Jiang, J. J. y Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. En *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*. Taiwan: ACL.

Jurafsky, D. y Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition and Computational Linguistics*. Prentice-Hall.

Kilgariff, A. y Rosenzweig, J. (2000). Framework and results for English SENSEVAL. *Computers and the Humanities*, 34, 15-48.

Krovetz, R. (1998). *More than one sense per discourse*. Princeton, Estados Unidos: NEC Labs America.

Lesk, M. E. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. En *Proceedings of the 5th International Conference on Systems Documentation* (pp. 24-26). Association for Computing Machinery.

Lin, D. (1998). An information-theoretic definition of similarity. En *Proceeding ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 296-304). Morgan Kaufmann Publishers.

Mihalcea, R. (2007). Using wikipedia for automatic word sense disambiguation. *NAACL-HLT 07*, 196-203.

Navigli, R. y Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4), 678–692.

Ponzetto, S. P. y Navigli, R. (2010). Knowledge-rich word sense disambiguation rivaling supervised systems. En *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 1522-1531). Association for Computational Linguistics.

Quillian, M. R. (1969). The teachable language comprehender: A simulation program and theory of language. *Communications of the ACM*, 12(8), 459-476.

RAE. (S. f. d). Homófono. En *Diccionario de la lengua española* (actualización de la 23ª ed.). <http://dle.rae.es/?id=KbZUpzR>

RAE. (S. f. b). Homógrafo. En *Diccionario de la lengua española* (actualización de la 23ª ed.). <http://dle.rae.es/?id=Kbl2l5o>

RAE. (S. f. c). Homónimo. En *Diccionario de la lengua española* (actualización de la 23ª ed.). <http://dle.rae.es/?id=Kbrilov>

RAE. (S. f. e). Metonimia. En *Diccionario de la lengua española* (actualización de la 23ª ed.). <http://dle.rae.es/?id=P7kP7xl>

RAE. (S. f. a). Semántica. En *Diccionario de la lengua española* (actualización de la 23ª ed.). <http://dle.rae.es/?id=XVRDns5>

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *International Joint Conference for Artificial Intelligence (IJCAI-95)* (pp. 448-453).

Schütze, H. (1992). Dimensions of meaning. En *Proceedings of Supercomputing '92* (pp. 787-796). IEEE Press.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1), 97-124.

Vasilescu, F., Langlais, P. y Lapalme, G. (2004). Evaluating variants of the lesk approach for disambiguating words. En *4th International Conference on Language Resources and Evaluation* (pp. 633-636). ELRA.

Weaver, W. (1955). Translation. En W. N. Locke y A. D. Boothe (Eds.), *Machine Translation of Languages* (pp. 15-23). MIT Press.

Wu, Z. y Palmer, M. (1994). Verb semantics and lexical selection. En *Proceedings of the 32th Annual Meetings of the Associations for Computational Linguistics (ACL-94)* (pp. 133-138). ACL.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. ACL-95, 189-196.

Yuret, D. (2004). Some experiments with a Naive Bayes WSD system. *Senseval-3: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. <http://www.aclweb.org/anthology/W04-0864>

Zhong, Z. and Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. En *Proceedings of the ACL 2010 System Demonstrations* (pp. 78-83). Association for Computational Linguistics.

Una orientación semántica al análisis de sentimientos

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 417-424.
<http://www.aclweb.org/anthology/P02-1053.pdf>

El artículo presenta un método basado en el aprendizaje no supervisado para clasificar revisiones y utiliza una orientación semántica para el análisis de sentimientos. A la hora de calcular la orientación semántica de una oración, es decir, determinar si el sentimiento recogido en la frase es positivo o negativo, se aplica un algoritmo de similitud entre palabras basado en la cantidad de información y la información mutua. De hecho, lo que se hace es calcular la similitud de cada adjetivo y adverbio de la frase con una palabra de referencia positiva («*excellent*», excelente en inglés) y una de referencia negativa («*poor*», malo en inglés) y, a partir de esos valores, se asigna una valoración global a la frase que indica la orientación semántica de la oración.

Word Sense Disambiguation

Manning, C. y Schütze, H. (1999). Word sense disambiguation. En *Foundations of Statistical Natural Language Processing* (pp. 229-264). MIT Press.
<https://github.com/shivamms/books/blob/master/nlp/Foundations%20of%20Statistical%20Natural%20Language%20Processing%20-%20Christopher%20D.%20Manning.pdf>

El capítulo siete del libro describe varios algoritmos probabilísticos para la desambiguación de los sentidos de las palabras. Los algoritmos presentados se basan tanto en el aprendizaje supervisado como en el no supervisado.

1. Indica las afirmaciones correctas sobre el significado de las palabras:
 - A. El sentido es la representación de uno de los aspectos del significado de una palabra.
 - B. El significado de una palabra se representa a través de su lema.
 - C. La semántica léxica estudia el significado de las palabras y las relaciones de sentido entre ellas.
 - D. Cada uno de los significados de una palabra se llama sentido.

2. Indica las afirmaciones correctas sobre las palabras y sus sentidos:
 - A. Dos palabras son homófonas si suenan igual, tienen distinto significado y se escriben igual.
 - B. Dos sentidos de una palabra son homógrafos porque se escriben de la misma forma.
 - C. Dos palabras son homónimas si se pronuncian igual, pero se escriben de forma diferente.
 - D. Dos sentidos de una palabra son homónimos si están relacionados entre sí.

3. Indica las afirmaciones correctas sobre las relaciones semánticas entre sentidos:
 - A. A veces puede existir una conexión semántica entre los sentidos de las palabras.
 - B. La metonimia es una forma de emplear una palabra en un sentido distinto al que propiamente le corresponde, pero con el que tiene alguna conexión.
 - C. La polisemia es una relación semántica entre los sentidos de una misma palabra.
 - D. La polisemia es un tipo particular de metonimia.

4. Indica las afirmaciones correctas sobre las relaciones entre sentidos de las palabras:

- A. La sinonimia es la relación contraria a la antonimia.
- B. Existe una relación de antonimia entre los sentidos de dos palabras si sus significados son idénticos o casi idénticos.
- C. Un tesoro es un diccionario que contiene una lista de palabras con sus sinónimos y sus antónimos.
- D. Las bases de datos de relaciones léxicas contienen un conjunto de lemas, el conjunto de sentidos de cada lema, su definición y una lista de sinónimos. Sin embargo, no es obligatorio que contengan la pronunciación de los lemas ni detalles de las relaciones entre lemas.

5. Indica las afirmaciones correctas sobre las relaciones entre sentidos de las palabras:

- A. Una palabra es un hiperónimo de otra si su significado incluye el significado de la otra palabra.
- B. Una palabra es un hipónimo de otra si su significado está incluido en el significado de la otra palabra.
- C. Las relaciones de hiponimia e hiperonimia se modelan a través de la jerarquía IS-A en las ontologías.
- D. Una palabra es un merónimo de otra si su significado mantiene con el significado de la otra palabra una relación de la parte respecto al todo.

6. Indica las afirmaciones correctas sobre los algoritmos de desambiguación del sentido de las palabras basados en aprendizaje supervisado:

- A. El clasificador se puede entrenar para desambiguar algunas palabras concretas de una muestra léxica o un texto entero.
- B. Utilizan un tesauro que contiene las relaciones entre sentidos de las palabras para poder entrenar el clasificador.
- C. Requieren tener un corpus de palabras etiquetadas con sus sentidos correctos para poder entrenar el clasificador.
- D. El vector de características lingüísticas de las palabras de contexto con las que se entrena el clasificador puede constar de características de colocación y características sobre las palabras vecinas.

7. Indica las afirmaciones correctas sobre los algoritmos de desambiguación del sentido de las palabras basados en conocimiento:

- A. Aplican el algoritmo de Lesk para seleccionar el sentido cuya definición en el diccionario comparte la mayor cantidad de palabras con el contexto.
- B. Utilizan un tesauro que contiene las relaciones entre sentidos de las palabras para poder entrenar el clasificador.
- C. Requieren tener un corpus de palabras etiquetadas con sus sentidos correctos para poder entrenar el clasificador.
- D. Realizan un entrenamiento indirecto aplicando algoritmos de aprendizaje supervisado débil.

8. Indica las afirmaciones correctas sobre los algoritmos de desambiguación del sentido de las palabras basados en aprendizaje semisupervisado:

- A. Aplican el algoritmo de Yarowsky para en cada iteración entrenar un clasificador a partir del corpus etiquetado, utilizar el clasificador para clasificar las instancias no etiquetadas y añadir a los datos de entrenamiento los ejemplos para los que se tenga mayor confianza en la clasificación.
- B. Utilizan un tesauo que contiene las relaciones entre sentidos de las palabras para poder entrenar el clasificador.
- C. Requieren tener un corpus de palabras etiquetadas con sus sentidos correctos para poder entrenar el clasificador.
- D. Para seleccionar las instancias a etiquetar en el conjunto inicial se debe utilizar una heurística, por ejemplo, la heurística de un sentido por colocación o la heurística de un sentido por discurso.

9. Indica las afirmaciones correctas sobre los algoritmos de desambiguación del sentido de las palabras basados en aprendizaje no supervisado:

- A. Utilizan un método estándar de agrupamiento, normalmente el algoritmo de agrupamiento aglomerativo, y una medida de distancia para determinar la similitud entre clústeres.
- B. Utilizan un tesauo que contiene las relaciones entre sentidos de las palabras para poder entrenar el clasificador.
- C. Requieren tener un corpus de palabras etiquetadas con sus sentidos correctos para poder entrenar el clasificador.
- D. También se llama inducción del sentido de las palabras porque el conjunto de sentidos de cada palabra se aprende automáticamente.

10. Indica las afirmaciones correctas sobre la similitud entre palabras:
- A. Los algoritmos que calculan la similitud entre palabras miden una relación binaria, es decir si las palabras son casi-sinónimos o no lo son.
 - B. Los algoritmos que calculan la similitud entre palabras utilizando la estructura de un tesoro se basan en la hipótesis de que los sentidos de las palabras son más similares si existe un camino más corto entre ellos.
 - C. Muchos de los algoritmos que calculan la similitud entre palabras realmente lo que hacen es utilizar una medida de relación entre palabras y no de similitud.
 - D. Existen algoritmos que calculan la similitud entre palabras utilizando la estructura de un tesoro e información probabilística derivada de un corpus como, por ejemplo, la cantidad de información.