

1. Introducción a la minería de textos

Definición 1 (Minería de textos). *Área que busca encontrar y extraer automáticamente información relevante a partir de texto expresado en lenguaje natural.*

Así como la minería de datos se basa en encontrar patrones en datos, la minería de textos, como una rama de ésta, busca encontrar patrones dentro de texto.

Justificación. El lenguaje natural es la forma en que nos comunicamos y a través de éste producimos una gran cantidad de información. Mucha de esta información está plasmada en texto. Sin embargo, dada la gran cantidad de textos que se producen, es casi imposible para un humano procesar esta información. Por tanto, se buscan métodos que hagan esta tarea más fácil. Tales como:

Extracción de palabras clave. Encontrar las palabras claves que describan un texto. Asimismo, se busca encontrar unidades fraseológicas (términos) que describan al texto.

Identificación de lenguaje. Determinar a qué lenguaje pertenece cada uno de los textos para clasificarlos por idioma.

Recuperación de Información (IR). Es una área amplia que busca extraer patrones relevantes según la necesidad del usuario. Aquí entra la recuperación de documentos a través de motores de búsqueda, la extracción de definiciones, de relaciones semánticas, identificación de tópicos, etc.

Categorización documental. Busca clasificar y agrupar documentos a partir de su temática o de otra característica relevante.

Minería de sentimientos. Esta área se preocupa por la detección de opiniones; por ejemplo, si una opinión en un texto es positiva, negativa o neutra. Asimismo, puede buscar más sentimientos relevantes, como depresión, agresividad, etc.

Atribución de autoría. Busca identificar quién es el autor de cierto texto a partir de semejanzas en los patrones de escritura de dicho texto.

Resumen automático. Quiere sintetizar la información que se encuentra dentro de un texto para hacerla más accesible al usuario.

Procesamiento del Lenguaje Natural. Al hablar de la minería de textos, es imposible dejar de lado el Procesamiento de Lenguaje Natural (PLN) pues los textos normalmente están codificados en lenguaje natural y, al tratarse de datos no estructurados, necesitamos lidiar con la estructura interna del lenguaje.

Definición 2 (Procesamiento de Lenguaje Natural). *El Procesamiento de Lenguaje Natural o PLN es un área de la inteligencia artificial que se enfoca en entender y emular los procesos humanos que implica el lenguaje con el objetivo de procesar computacionalmente el lenguaje natural.*

2. Datos estructurados y no estructurados

Definición 3 (Datos estructurados). *Los datos estructurados se caracterizan por ser datos que se encuentran en un campo fijo dentro del texto (o cualquier tipo de archivo).*

Definición 4 (Datos no estructurados). *Son tipos de datos que se oponen a los datos estructurados. En estos datos no se encuentran dentro de campos ni en bases de datos.*

Extracción de información estructurada. Cabe señalar que la mayoría de los textos cuentan con información estructurada a pesar de no contar con un lenguaje de marcado. Por ejemplo:

1. Números telefónicos
2. Fechas
3. Direcciones postales
4. Correos electrónicos
5. Direcciones web
6. Tablas, figuras, referencias, etc.

Un tipo especial de información estructurada son las **Entidades Nombradas** (*Named Entities*).

Definición 5 (Entidad nombrada). *Las entidades nombradas son frases que contienen nombres de personas, organizaciones, lugares, tiempos y cantidades.*

A la extracción de entidades nombradas las llamaremos **reconocimiento de entidades nombradas** o **NER** (Named-Entity Recognition).

2.1. Extracción de patrones regulares

Para la extracción de datos estructurados dentro del texto es común utilizar expresiones regulares. Para esto, presentamos algunas definiciones útiles.

Definición 6 (Alfabeto). *Un alfabeto es un conjunto $\Sigma = \{a_1, a_2, \dots, a_n\}$, donde cada $a_i \in \Sigma$ es un símbolo al que llamamos letra.*

Definición 7 (Cadena). *Sea Σ un alfabeto. Una cadena sobre Σ es una función $\bar{x} : n \rightarrow \Sigma^n, n \in \mathbb{N}$.*

Notación 1. Al conjunto de todas las cadenas sobre Σ se le denota como Σ^* .

Definición 8 (Lenguaje). *Un lenguaje sobre un alfabeto Σ es un subconjunto de Σ^* .*

Ejemplo: Sea $\Sigma = \{a, b, c, \dots, x, y, z\}$, son lenguajes sobre este alfabeto: Σ^* , las cadenas que constan sólo de vocales, las que constan sólo de consonantes, las palabras de un texto escrito, etcétera.

Nota 1. n denota la longitud de \bar{x} tal que $|\overline{overlinex}| = n$. Para denotar el i -ésimo segmento de \bar{x} se puede utilizar la notación $\bar{x}[i]$. Si $|\bar{x}| = 0$, entonces se denota a \bar{x} como $\epsilon = \emptyset$ (o bien λ).

Ejemplo: Sea $\Sigma = \{a, b\}$ un alfabeto y sea $\bar{x} = bababa$ una cadena de longitud 6, $|\bar{x}| = 6$. Se tiene que $\bar{x}[1] = b, \bar{x}[2] = a, \bar{x}[6] = a$.

Definición 9 (Concatenación). Sean $\bar{x} = (a_1, \dots, a_n)$ y $\bar{y} = (b_1, \dots, b_m)$ dos cadenas sobre Σ , entonces la concatenación de \bar{x} con \bar{y} , denotada como $\bar{x} \cdot \bar{y}$, es la cadena $(a_1, \dots, a_n, b_1, \dots, b_m)$.

Nota 2. Queda claro que $|\bar{x} \cdot \bar{y}| = |\bar{x}| + |\bar{y}|$.

Ejemplo: Siendo $w_1 = saca$ y $w_2 = puntas$ dos cadenas sobre el alfabeto $\Sigma = \{a, b, c, \dots, w, y, z\}$, entonces $w_1 \cdot w_2 = sacapuntas$.

Definición 10 (Operador de exponente). Sea $w \in \Sigma^*$ una cadena, entonces:

1. $w^0 := \epsilon$
2. $w^{i+1} := w^i \cdot w$

De donde se puede obtener, que:

$$\prod_{i<0} w_i := \epsilon$$

y

$$\prod_{i<n+1} w_i := \prod_{i<0} w_i \cdot w_n$$

Definición 11 (Kleene *). Sea $x \in \Sigma^*$ una cadena. x^* es la repetición de la cadena tanto como se quiera.

Ejemplo: Sea $w = ba$ una cadena. Entonces $w^0 = \epsilon, w^1 = w = ba, w^2 = w \cdot w = baba, w^3 = w^2 \cdot w = bababa$, etcétera.

Definición 12 (Operaciones sobre cadenas). Sean L y M dos lenguajes sobre un alfabeto Σ , entonces, se pueden realizar las siguientes operaciones:

1. $L \cdot M := \{x \cdot y | x \in L, y \in M\}$
2. $L^0 := \epsilon$
3. $L^{n+1} := L^n \cdot L$
4. $L^* := \bigcup_{n=0}^{\infty} L^n$

Cerradura de Kleene

5. $L^+ := \bigcup_{n=1}^{\infty} L^n$
6. $L/M := \{y \in \Sigma^* \mid \exists x \in M \ni y \cdot x \in L\}$
7. $L \setminus M := \{y \in \Sigma^* \mid \exists x \in M \ni x \cdot y \in L\}$

Finalmente, podemos definir lo que es una expresión regular:

Definición 13. Dado un alfabeto Σ , el conjunto de expresiones regulares sobre Σ , E , se define de la siguiente manera:

1. \emptyset es una expresión regular.
2. ϵ es una expresión regular.
3. $a \in \Sigma$ es una expresión regular.
4. $r_1 \in E, r_2 \in E \implies (r_1 \cdot r_2) \in E$
5. $r \in E \implies (r)^* \in E$

Ejemplo. Las siguientes cadenas son expresiones regulares:

- $/[^]+/$ “todo lo que no es un espacio en blanco una o más veces. Puede servir para indocar palabras”.
- $/[A-Z][^]+/$ “Una cadena iniciada con una letra mayúscula que no contenga espacios. Puede usarse para nombres propios”.
- $/[a-z]+@[a-z].com/$ “Un correo electrónico”.

2.2. Minería de datos no estructurados

El curso abordará la extracción de datos no estructurados en texto. Para esto, usaremos en gran medida el PLN. En términos generales, presentaremos la extracción de datos estructurados de la siguiente forma:

Representación de textos. Aprenderemos cómo representar los textos en vectores dentro de un espacio vectorial a partir de diferentes técnicas. Esto, con el fin de poder elaborar diferentes aplicaciones de manera más sencilla.

Clasificación y agrupamiento. A partir de diferentes modelos de aprendizaje de máquina, se verá cómo clasificar y agrupar textos con características similares para extraer datos relevantes a partir de éstos.

Recuperación de información. Se aprenderá cómo extraer información relevante a partir de textos.

Modelos de tópicos. Dentro de la información relevante de un texto, se encuentran los tópicos. En este apartado aprenderemos diferentes técnicas para su extracción y modelado.

3. ¿Qué es la lingüística?

El procesamiento del lenguaje natural es una rama de la Inteligencia Artificial que integra recursos y métodos computacionales a la metodología lingüística. Busca que la máquina sea capaz de interpretar el lenguaje natural. Para esto, es necesario conocer la materia de estudio: el lenguaje. El lenguaje humano es complejo: se manifiesta a través de sonidos producidos por el sistema fonador para formar palabras, oraciones, discursos completos. Desde las palabras hasta un discurso es interpretado, y se le asigna un significado. Es la lingüística la que se encarga de estudiar al lenguaje. Entonces, podemos dar la siguiente definición.

Definición 14 (Lingüística). *La lingüística es el estudio de todas las manifestaciones del lenguaje humano.*

Ferdinand de Saussure asignaba las siguientes tareas a la lingüística:

- Descripción de las lenguas naturales. La lingüística busca entender los procedimientos que las personas usamos para producir y entender el lenguaje. Asimismo, busca describir las características particulares de las lenguas, como su estructura, las relaciones que se establecen entre elementos, etc.
- Deducir leyes generales del lenguaje natural. Además de describir características particulares, la lingüística busca establecer rasgos que sean generales a través de diferentes lenguas, así como establecer elementos que ayuden a distinguir y clasificar las lenguas del mundo.
- Delimitarse y definirse ella misma.

3.1. Diacronía y sincronía

El lenguaje, como fenómeno humano ha acompañado al hombre a través de los años. Por esto mismo, el lenguaje se ha visto sujeto a cambios, que en largos períodos de tiempo son más notorios. Por ejemplo, la palabra «gato» se decía en latín «catum» y proviene de ésta. Un lingüista bien podría estudiar los fenómenos que generaron a «gato» a partir de «catus»¹. Otro ejemplo de un estudio lingüístico de este tipo es la conjugación del futuro en español, como «amaré» que proviene de la forma latina «amare habeo» que pasó por «amar he» y que terminó en el futuro que conocemos hoy. En general, a este tipo de análisis lingüístico que implica el estudio de la lengua a través del tiempo se le conoce como diacronía.

Definición 15 (Diacronía). *Estudio del lenguaje a través de diferentes períodos históricos.*

¹El sonido de 'c' pronunciado como 'k' en latín cambió ciertos rasgos hasta convertirse en 'g'. También la terminación 'um' se simplificó en 'o'.

Por otra parte, existen estudios lingüísticos que se dedican a analizar la manifestación de un fenómeno del lenguaje en un período específico del tiempo. Un ejemplo podría ser el estudio la forma «haber + verbo en infinitivo» en el español mexicano actual. A este tipo de estudios se les conoce como sincrónicos. La lingüística computacional, generalmente, se enfoca a fenómenos sincrónicos; sin embargo, esto no implica que no pueda enfocarse también a la diacronía.

Definición 16 (Sincronía). *Es el estudio del lenguaje sin tomar en cuenta cambios o influencias históricas.*

3.2. Competencia y actuación

Ya decíamos más arriba que uno de los objetivos de la lingüística es deducir leyes generales del lenguaje. Para esto, es necesario abstraer fenómenos observados en hablantes particulares. Es decir, para saber que un verbo se conjuga en pasado, primera persona singular con la terminación -ó, se tiene que ver primero que varios hablantes producen formas como «comi-ó», «am-ó», etc. De allí, el lingüista puede generalizar una ley del tipo *verbo-ar, -er, -ir → verbo-ó*. De esta forma, el lingüista podría llegar a construir una gramática general de la lengua. A este sistema abstracto, lo llamamos *competencia*.

Definición 17 (Competencia). *Es el sistema lingüístico en abstracto, también conocido como lengua.*

Sin embargo, constantemente escuchamos que los hablantes producen formas lingüísticas diferentes a las establecidas por una gramática; además, es más que obvio que cada hablante tiene una forma particular de hablar: usan diferentes palabras, conjugan de manera distinta, pronuncias sonidos diferentes, etc. Por tanto, debemos introducir el concepto de *actuación*.

Definición 18 (Actuación). *Es la producción lingüísticas de hablantes particulares, también conocida como habla.*

Dentro de la actuación, cabe resaltar que el conocimiento de palabras de cada hablante varía considerablemente. En palabras más rebuscadas, el léxico de los hablantes es diferente. A este grupo de palabras, o lexemas, que tiene un hablante, se le ha llamado lexicón. Así, introducimos las siguientes tres definiciones.

Definición 19 (Lexema). *El lexema es una unidad con significado léxico. En otras palabras, es un elemento del lexicón.*

Definición 20 (Léxico). *Es un conjunto de lexemas.*

Definición 21 (Lexicón). *Diccionario mental del hablante. También podemos considerarlo como un diccionario (electrónico) que se utiliza para diferentes tareas de procesamiento del lenguaje natural.*

4. Concepto de corpus

Un corpus lingüístico puede definirse, en su forma más sencilla, como cualquier repertorio de textos; es decir, cualquier colección de más de un texto puede considerarse ya como un corpus, puesto que reúne muestras de lenguaje humano. Para el caso de la filología, el corpus bien puede entenderse como la recopilación bien organizada de los textos orales o escritos, así como de los documentos que los contienen (Torruella y Llisterri, 1999). McEnery y Wilson (2001) aseveran que para la elaboración de un corpus lingüístico deben considerarse los siguientes puntos:

- Que esté bien seleccionado y que cuente con representatividad (variedad y equilibrio).
- Que tenga un tamaño finito de palabras.
- Que cuente con una estructura capaz de ser interpretada por una computadora.
- Que contenga una referencia estandarizada.

Definición 22 (Corpus). *Un corpus es una recopilación bien organizada de muestras del lenguaje a partir de materiales escritos o hablados, agrupados bajo criterios mínimos.*

Un corpus, debido a su constitución y sus características, puede ofrecer en su análisis tanto datos de tipo cuantitativo como de tipo cualitativo, puesto que permite que el estudio de las muestras contenidas en éste se realice, ya sea desde una perspectiva estructural o bien por medios estadísticos. Cabe resaltar que estas dos formas de análisis están estrechamente ligadas y los datos cuantitativos también arrojan información de formas estructurales de la lengua; es decir, obtenemos datos como flexiones verbales, formas de afijos, sintácticos, etcétera.

Cuando hablamos de un análisis cuantitativo nos referimos al uso de los datos arrojados por el corpus como una base para describir aspectos del uso real de la lengua a partir de datos estadísticos, mientras que el análisis cualitativo se enfoca en la estructura de la lengua, es decir, del análisis de las muestras contenidas en el corpus desde una perspectiva lingüística. En otras palabras, el análisis cuantitativo arroja información numérica de un corpus, como puede ser la ocurrencia de determinados lemas, estructuras, formas, etc., mientras que el análisis cualitativo se basa en la observación de los fenómenos lingüísticos derivados del corpus. Sin embargo, los datos numéricos también son interpretados como fenómenos lingüísticos y se ligan estrechamente con los cualitativos y lo mismo pasa en viceversa. Los datos cualitativos pueden presentar información estadística.

4.1. Tipología de corpus

Los corpus lingüísticos pueden ser clasificados de distintas maneras según determinados criterios. Una primera tipología básica de los corpus lingüísticos

distingue entre corpus textuales y corpus orales (Jiménez Pozo, 1999). Como su nombre lo dice, un corpus textual es aquel conformado por documentos que contengan únicamente muestras de lenguaje escrito, mientras que el corpus oral lo constituyen transcripciones de lengua hablada o grabaciones de ésta. Los corpus textuales pueden considerarse como repertorios de escritos, ya sean físicos o electrónicos, y, por su naturaleza, su manejo es más sencillo para una computadora. Por el otro lado, los corpus orales son repertorios de muestras del lenguaje hablado y su procesamiento digital es más complicado que el del lenguaje escrito.

Otra clasificación es la dada por Llisterri y Torruela (1999), que se asemeja a la de Atkins, Clear y Ostler (1992), y está basada en el nivel de distribución dentro del corpus; estos autores distinguen entre corpus, subcorpus y componente, los cuales explicamos a continuación:

Corpus. Un corpus es definido por los autores como «un conjunto homogéneo de muestras de lengua de cualquier tipo (orales, escritos, literarios, coloquiales, etc.), los cuales se toman como modelo de un estado o niveles de lengua predeterminado» (Torruella y Llisterri, 1999, pág. 8). Este conjunto de enunciados contenido en un corpus, para los autores, debe permitir un análisis que dé pie al mejoramiento en el conocimiento de las estructuras al interior del sistema lingüístico que representan.

Subcorpus. En su forma más simple, se puede definir como un subgrupo de un corpus (Atkins, Clear, y Ostler, 1992). Para Llisterri y Torruela un subcorpus puede ser de dos tipos: el primero está representado por una selección estática de textos derivados de un corpus de mayor tamaño y complejidad, que divide, a su vez, en muestras textuales más específicas; el segundo tipo es definido como una selección dinámica de textos pertenecientes a un corpus en crecimiento, en otras palabras se trata de textos cuyo fin es integrarse al apartado de un corpus general, de mayor tamaño.

Componente. Es una colección de muestras de un corpus o subcorpus, las cuales responden a un criterio lingüístico específico muy concreto. Los componentes reflejan un tipo determinado de lengua. Podemos decir que tanto los corpus como los subcorpus son muy heterogéneos, mientras que los componentes son muy homogéneos.

Para desarrollar una división más amplia de los tipos de corpus, tomamos en consideración principalmente el trabajo sobre tipología de corpus de Sierra y Rosas (2009) , quienes clasifican los corpus a partir de los siguientes criterios:

4.1.1. El origen de los elementos.

Según el origen de sus elementos se considera a un corpus como oral y escrito; esto se refiere a que, como sus nombres lo dicen, responden a las muestras que conforman el corpus. Tales muestras pueden ser de tipo oral o material escrito, las primeras responden a grabaciones o transcripciones fonéticas o fonológicas

del lenguaje hablado, mientras que las segundas son propiamente muestras de lenguaje escrito.

4.1.2. La codificación y anotación

Conforme a la anotación, la distinción es la misma sugerida por McEnery y Wilson (2001) y Torruella y Llisterri (1999); es decir, se distingue entre un corpus simple y un corpus anotado o codificado. De igual forma, Sierra y Rosas (2009) proponen un esquema más amplio de tipos de anotación.

4.1.3. La especificidad de los elementos

Conforme a la especificidad de sus elementos, se distingue entre dos tipos de corpus: los corpus generales y los corpus especializados o específicos. Los primeros aportan información de tipo general, esto es que recogen todo tipo de géneros y tipologías textuales. Los corpus especializados, por su parte, recogen información de una o varias áreas en particular; éstos pueden, a su vez, ser informativos y contener textos periodísticos, científicos o similares, mientras que, por otro lado, están los literarios, que se enfocan a textos del área de la literatura.

4.1.4. La temporalidad

Conforme al criterio de temporalidad, una primera distinción se hace entre los corpus diacrónicos y los corpus sincrónicos. Un corpus diacrónico puede definirse como aquel que responde a diferentes períodos de tiempo, mientras uno sincrónico sólo responde a un período temporal. Los corpus diacrónicos, a su vez, pueden subdividirse en cronológicos y periódicos; los cronológicos contienen textos de años en orden consecutivo; los periódicos se encargan de estudiar la lengua en diversos períodos históricos. Los corpus sincrónicos también presentan una subdivisión en contemporáneos e históricos; los contemporáneos se componen de textos actuales, mientras los históricos de textos de una época pasada, sin llegar a abarcar más de un período temporal.

4.1.5. El propósito

Según el propósito, encontramos también dos tipos de corpus. El primero de propósito específico y el segundo multipropósito. Los de propósito específico son corpus construidos para un estudio lingüístico concreto, a diferencia de los multipropósito que tratan de abarcar un análisis lingüístico más amplio y por tanto pueden ser reutilizables para diferentes investigaciones.

4.1.6. El lenguaje

A partir del criterio de la lengua, encontramos los corpus monolingües, es decir que cuentan con textos en un solo idioma; los corpus comparables son una especie de corpus monolingües que cuentan con traducciones de textos a una

misma lengua; por último, los corpus multilingües, que consisten en textos en varios idiomas.

4.1.7. La cantidad de texto

De acuerdo con la cantidad de texto, tenemos los siguientes tipos: corpus grande, que pueden contener una cantidad considerable de texto (desde diez millones de palabras); corpus pequeño que son aquellos que contienen una cantidad menor de textos y corpus monitor, que contiene un volumen fijo que se actualiza constantemente.

4.1.8. La distribución de los textos

Por la distribución de los textos, clasificamos los corpus en desequilibrados y equilibrados. El corpus desequilibrado contiene textos en cantidades no proporcionales entre sí; por otro lado, el equilibrado procura distribuir sus textos de manera proporcional entre sí, dentro de éste se encuentran los corpus piramidales, cuyos textos están distribuidos en diferentes niveles ascendente; el primer nivel de un corpus de este tipo contiene poca variedad temática en una cantidad grande de textos, el segundo contiene más variedad temática en menos textos, el tercer nivel tiene más variedad temática en pocos textos y así sucesivamente, según se determinen los niveles de la pirámide en el corpus.

4.1.9. La documentación

Según su documentación, se tienen los corpus documentados y no documentados; los primeros contienen registros de la documentación de los textos que permiten hacer búsquedas específicas y conocer la proveniencia de los textos, mientras que los segundos carecen de esto.

4.1.10. La autoría

Según la autoría se tienen dos tipos de corpus: los canónicos y los genéricos. Los primeros responden a textos de un único autor, mientras que los segundos responden a documentos de un solo género literario. Por otra parte, cuando la información contenida en el corpus no responde a ninguno de estos dos criterios, se puede decir que tenemos un corpus de autoría variada.

4.2. Vocablos

Generalmente, el tamaño de un corpus se mide por la cantidad de palabras que contiene. Sin embargo, uno puede darse cuenta de que contar los elementos entre los espacios en blanco no equivale a contar cada ocurrencia de una forma de palabra; es decir, no es lo mismo decir que un corpus cuenta con 3 apariciones de 'gato' que contar las palabras distintas, donde 'gato' a pesar de tener 3 apariciones sólo contaría como un único elemento. Por tanto, es importante distinguir entre los conceptos de *tipo* y *token*.

Definición 23 (Token). *Un token es la ocurrencia individual de una palabra dentro de un corpus.*

Definición 24 (Tipo). *Los tipos son los diferentes elementos lingüísticos que existen en un corpus.*

Considérese el siguiente párrafo de un texto:

El cacomixtle es una especie de mamífero carnívoro de la familia de los prociónidos, de tamaño medio a pequeño, de color pardo claro y con cola muy larga, ésta con una coloración característica de anillos oscuros. El cacomixtle es arborícola, nocturno y de naturaleza solitaria.

En este pequeñísimo corpus existen 53 tokens; es decir, hay 53 elementos o palabras sin importar que estas se repitan (en este caso 'cacomixtle' cuenta dos veces). Por su parte, tiene 39 tipos, es decir, hay 39 palabras distintas (aquí 'cacomixtle cuenta sólo una vez').

5. Niveles del lenguaje

El estudio del lenguaje es complicado y abarca muchos factores que van desde el fenómeno físico de los sonidos producidos por el hablante hasta el significado y las connotaciones que los hablantes de una sociedad le asignan a las palabras. Por tanto, es importante dividir la lingüística dentro de ramas que ayuden a facilitar y estructurar el estudio del lenguaje. Podemos hablar de los siguientes niveles lingüísticos:

1. Fonética y fonología
2. Morfología
3. Morfosintaxis
4. Sintaxis
5. Semántica
6. Pragmática

5.1. Fonética y fonología

En términos generales, la fonética y la fonología se encargan del estudio de los sonidos del habla. La fonética, por una parte, se encarga de estudiar estos sonidos a nivel de la competencia; es decir, la fonética estudia los sonidos del habla como un fenómeno físico. Para esto, necesita de grabaciones de hablantes particulares para analizar las producciones de los sonidos en cada individuo particular.

Por otra parte, la fonología estudia los sonidos del lenguaje en forma abstracta, es decir, en la competencia. Puede verse que para llegar a la fonología se requiere antes de la fonética, pues la fonología generaliza las observaciones hechas en la fonética. Supóngase que se tiene la palabra «chango» y dos hablantes: el uno del centro de México y el otro del norte del país. En la fonética, se podrá ver que el hablante del centro pronunciará la secuencia inicial de la palabra como 'ch', mientras la del norte como 'sh'. La fonología reconocerá ambas producciones como un mismo sonido o *fonema* y le asignará una representación, en este caso el símbolo /tʃ/, el cual es una representación abstracta de las producciones de los hablantes.

5.2. Morfología

Si bien la fonética y la fonología se encargan de estudiar los sonidos, la morfología se enfoca en los elementos formados por las unidades fónicas, es decir, las palabras. Una definición muy general de la morfología puede ser la siguiente: «Es el estudio de la estructura interna de las palabras». Por ejemplo, la morfología se encarga de determinar que las formas de palabra 'comemos' y 'comió' pertenecen a una misma clase de palabra. Por ejemplo, tómese en cuenta las siguientes palabras:

- (1) niñ-o
- (2) niñ-a
- (3) niñ-o-s

Todas ellas hacen referencia a un infante humano, pero las variaciones en sus terminaciones afectan el significado. En el primer caso, se habla de un infante masculino, en el segundo de género femenino y en el tercer caso se habla de dos o más infantes. Entonces, la morfología se encarga de ver cómo estas variaciones en la estructura de la palabra afectan al significado, y, al mismo tiempo, determina que todas estas palabras pertenecen a un mismo lexema, es decir, a una misma entrada en el diccionario. Pero, por desgracia, no siempre es tan fácil determinar esto. Considérese:

- (4) soy
- (5) fui

Donde 'es' determina el presente del verbo 'ser' y 'fui' el pasado (ambos en primera persona singular). En este caso, se da un cambio completo de la palabra². Estos fenómenos también son estudiados en la morfología. Por lo tanto, una mejor definición de morfología es la siguiente:

Definición 25 (Morfología). *La morfología es el estudio de las co-variaciones sistemáticas en la forma y el significado de las palabras.*

Retómense los ejemplos de (1), (2) y (3). En este caso, podemos ver que podemos dividir la palabra en diferentes secuencias, como '-o', '-a' o '-s'. También vemos que estos elementos son los que alteran el significado: '-a' introduce el género femenino, así como '-s' el sentido de «muchos». Asimismo, sabemos que podemos combinar estos elementos, por ejemplo, podemos formar la palabra 'niñ-a-s' que ahora, además del género femenino, señala pluralidad. Incluso podemos combinarlo en una palabra como 'gat-a-s', donde ahora lo que cambia es el sentido de infante y se introduce un nuevo sentido: el de felino doméstico. A estos elementos mínimos que podemos intercambiar de esta forma les llamamos *morfemas*.

Definición 26 (Morfema). *Un morfema es la unidad mínima con significado en una producción lingüística.*

Sin embargo, los morfemas pueden variar según cómo éstos se combinan. Por ejemplo, el caso de 'in-':

- (6) in-tocable
- (7) im-personal

²A este fenómeno se le conoce, en la terminología lingüística, como *suplición*.

(8) i-nato

En todos estos casos, se nota que 'in-', 'im-' e 'i-' están introduciendo la misma significación. De hecho, se trata del mismo morfema que se ve alterado por el morfema que lo acompaña. A estos elementos se les conoce como *alomorfos*.

Definición 27 (Alomorfo). *Los alomorfos son las diferentes manifestaciones concretas que tiene un morfema en diferentes contextos lingüísticos.*

Generalmente, los alomorfos tienen un representante de clase, el morfema. En este caso, éste es 'in-', ya que los demás alomorfos pueden generarse de éste a partir de reglas lingüísticas.

Bases y afijos

Ya en los ejemplos anteriores se ha visto que existen morfemas que están más sujetos a recibir en su estructura a otros morfemas con cambios mínimos de significado. En el ejemplo de 'niñ-o' vemos que si modificamos el morfema '-o' por '-a' varía el género, pero el sentido de infante se sigue conservando. No pasa así si modificamos 'niñ-' por 'gat-', verbigracia. En general, se puede decir que los morfemas de este tipo son modificados a partir de un proceso morfológico, es decir, son las *bases* sobre los que se añaden otros morfemas en la realización de un proceso morfológico.

Definición 28 (Base). *La base es el elemento de una palabra que es susceptible un proceso morfológico.*

Más adelante, veremos que las bases pueden encontrarse a parte de la información que estás contienen, que es mayor a los elementos que no son bases. Esto es, las bases son los elementos morfológicos que contienen mayor información. Por su parte, a los morfemas que no son bases sino que se adhieren a éstas para formar palabras, las conocemos como *afijos*.

Definición 29 (Afijo). *Son los elementos morfológicos que se adhieren a las bases para crear nuevas palabras a partir de ellas.*

En el ejemplo de 'niñ-o', 'niñ-' es la base y '-o' es un afijo. En 'in-completo', la base es 'completo' y el afijo es 'in-'. Ya con estos dos ejemplos podemos ver que los afijos pueden adherirse a la base desde la izquierda o bien desde la derecha. De esta forma, podemos dividir los afijos dentro de las siguientes subclases:

Sufijo . Es el afijo que se adhiere a la base desde la derecha, como en el ejemplo de: 'niñ-o'.

Prefijo . Es el afijo que se adhiere a la base desde la izquierda, como en el ejemplo de 'in-completo'.

Interfijo . Es un afijo que se presenta entre dos morfemas, pero nunca en el extremo izquierdo o derecho de una palabra. Por ejemplo, tómese la palabra 'pan' y dérvase con el sufijo '-ero'; en español sería incorrecto decir 'panero', así que se utiliza el interfijo '-ad-' y se obtiene 'pan-ad-ero'.

Infijo . Es un afijo que rompe una base y se inserta en medio de esta. Un ejemplo en español puede ser el diminutivo de 'azucar', pues la misma palabra es una base, pero cuando se deriva con el morfema '-it' se obtiene 'azuqu-it-ar'. El contraste con el interfijo es que el infijo rompe la base, mientras que el interfijo no lo hace, sino que sólo aparece entre dos morfemas, muchas veces por motivaciones fonológicas.

Circunfijo . Es el afijo que rodea a una base. Por ejemplo, en japonés la base 'yum-' tiene el sentido de dormir, pero una forma reverencial de decir dormir se hace mediante un circunfijo: 'o-yum-i-suru', donde 'o-suru' es el circunfijo.

Transfijo . Es el afijo que rompe una base y además rodea una parte de ésta. Es una especie de combinación entre un infijo y un circunfijo. Por ejemplo, en persa, 'kitab' es la palabra para libro y 'k-a-t-i-b' significa «que escribe». En este caso, '-a-i-' está funcionando como transfijo.

En este curso, nos enfocaremos solamente en los sufijos y los prefijos (los cuales se retomarán en la teoría de lenguajes formales). Es más, en PLN es común sólo tratar con los sufijos, pues los otros tipos de afijo representa un reto de mayor complejidad. En general, en PLN se busca obtener las bases de las palabras eliminando sus afijos. Sin embargo, muchas veces también nos será útil obtener la forma de diccionario de una palabra, por ejemplo, de los verbos 'comiendo', 'comí', 'comerás' se puede llegar a 'comer'; a esta forma de diccionario se le llama *lema* y al proceso de obtenerlo *lematización*.

Definición 30 (Lema). *Es la forma de diccionario de las realizaciones concretas de una palabra.*

6. Stemming y Lematización

Definición 31 (Stemming). *El proceso de stemming o truncamiento consiste en reducir un token léxico a una supuesta base por medio de truncar los afijos correspondientes.*

Ejemplo 1. Tómense las palabras *gatos*, *gatito*, *gata*. Un proceso de stemming eliminará los afijos */-o/*, */-ito/* y */-a/*, dejando la forma base *gat-*.

Definición 32 (Lematización). *El proceso de lematización consiste en llevar un token léxico a su forma de diccionario.*

Ejemplo 2. En el ejemplo anterior, la palabra que se obtendría sería *gato*. Otro ejemplo, con los verbos *amé*, *amamos*, *amemos*, el proceso de lematización devolvería el lema *amar*.

6.1. Algoritmo de Porter

El algoritmo de Porter es un algoritmo basado en reglas creado originalmente para el inglés y adaptado para el español. Se basa en buscar elementos similares a sílabas y a partir de ellas realizar los cortes de los afijos necesarios.

Definiciones

1. Vocales del español: a e i o u á é í ó ú
2. R1 es la región después de la primera consonante que sigue a una vocal, o la región nula al final de la palabra si no existe tal vocal.
3. R2 es la región después de la primera consonante seguida por una vocal en R1, o es la región nula al final de la palabra si no existe dicha consonante.
4. RV es la región después de la tercera letra; o bien la región después de la primera vocal que no sea principio de palabra o final de palabra. Si la segunda letra es una consonante, RV es la región después de la vocal siguiente. Si las dos primeras letras son vocales, RV es la región después de la siguiente consonante. En cualquier otro caso, RV es la región después de la tercera letra.

Algoritmo

Siempre realizar los Pasos 0 y 1.

Paso 0 - Clítico agregado:

> Se buscan los siguientes sufijos:

■ me se selo selas selos la le lo las les los nos.

> Se borran si son aparece en RV alguno de los siguiente casos:

- iéndo ándo ár ér ír
- ando iendo ar er ir
- yendo (seguido de) u

Paso 1 - Eliminación de sufijos básicos:

> Se buscan los siguientes sufijos y se lleva a cabo la acción indicada:

- anza anzas ico ica icos icas ismo ismos able ables ible ibles ista istas oso osa osos osas amiento amientos imiento imientos → se borran si aparecen en R2.
- adora ador acción adoras adores acciones ante antes ancia ancias → Se borran si aparecen en R2, o precedidos por «ic» en R2.
- logía logías → Se reemplazan por «log» si aparecen en R2.
- ución ucciones → Se remplazan por «u» si aparecen en R2.
- encia encias → se reemplazan por «ente» si aparecen en R2.
- mente → Se borran si aparecen en R2.
- idad idades → Se borran si aparecen en R2 (si son precedidas por «abil» «ic» o «iv»).
- iva ivo ivas ivos → Se borran si aparecen en R2 (precedidas por «at»).

Realizar el Paso 2a si no se ha removida ningún final de palabra en el Paso 1.

Paso 2a - Sufijos verbales que comienzan por «y»:

> Buscar los siguientes sufijos en RV y borrar si son precedidos por «u»:

- ya ye yan yen yeron yendo yo yó yas yes yais yamos.

Realizar el Paso 2b si en 2a no ha removido sufijos.

Paso 2b - Otros sufijos verbales:

> Buscar los siguientes sufijos en RV y borrarlos si son precedidos por «u»:

- en es éis emos → (si son precedidos por «gu» se borra también «u»).
- arían arías arán arás aráis aría aréis aríamos aremos ará aré erían erías erán erás eráis ería eréis eríamos eremos erá eré irán irías irán irás iráis iría iréis iríamos iremos irá iré aba ada ida ía ara iera ad ed id ase iese aste iste an aban ían aran ieran asen iesen aron ieron ado ido ando iendo ió ar er ir as abas adas idas ías aras ieras ases ieses ís áis abais íais arais ierais aseis ieseis asteis isteis ados idos amos ábamos íamos imos áramos iéramos iésemos ásemos.

Siempre realizar el Paso3.

Paso 3 - Sufijos residuales:

> Buscar por los siguientes sufijos y realizar la acción indicada:

- os a o á í ó → Borrar si se encuentran en RV
- e é → Borrar si están en RV y son precedidos por «gu»; si la «u» está también en RV, ésta también se borra.

Paso final:

> Eliminar acentos agudos.

6.2. Algoritmo basado en teoría de la información

En teoría de la información, podemos definir una base de la siguiente manera:

Definición 33 (Base). *Una base es el elemento que cuenta con mayor información, es decir:*

$$\text{Base} := \arg \max_i \{I(a_i)\}_{i=1}^n \quad (1)$$

Donde $I(x) = -\log P(x)$ y $a_i, i = \{1, \dots, n\}$ son elementos morfológicos. También se puede definir en términos de entropía de la siguiente forma:

$$\text{Base} := \arg \min_i \{H(a_i)\}_{i=1}^n \quad (2)$$

donde $H(x) = -\sum_{j=1}^k P(x_j) \cdot \log P(x_j)$.

6.3. Ultra-stemming

Ultra-stemming es un algoritmo de fuerza bruta que se basa en realizar un corte fijo a las palabras. Se basa en la intuición lingüística de que existe un tamaño mínimo de palabra. Para el español, por ejemplo, este tamaño mínimo es de dos sílabas, como en ga-to, li-bro, can-sar, etc. Estas dos sílabas se manifiestan entre 4 y 6 letras en español. Por tanto, ultra-stemming seleccionará un tamaño de palabra y eliminará todo elemento que exceda esta longitud.

Ejemplo 3. Si se tienen las palabras gato, gatos, comíamos, presidente, y elegimos un tamaño de palabra igual a 4, ultra-stemming devolverá las bases: gato-, comí-, presi-.

7. Morfosintaxis

Antes de introducir el nivel de análisis sintáctico, nos detendremos en la interacción entre la morfología y la sintaxis. En este curso, nos enfocaremos únicamente al estudio de las categorías gramaticales de las palabras o, como se le llama en PLN, *parts of speech*³. En primer lugar, definamos morfosintaxis.

Definición 34 (Morfosintaxis). *La morfosintaxis se encarga del estudio de las reglas que gobiernan la forma en que las palabras se organizan en oraciones, así como la forma y función de éstas.*

Entonces, lo que nos interesa en la morfosintaxis es determinar qué tipo de función desempeña una palabra dentro de una oración. Esta idea es precisamente lo que llamamos partes de la oración.

Definición 35 (Partes de la oración). *Es la clasificación de las palabras según su función.*

Ya hemos escuchado los términos de sustantivo, verbo o adjetivo, que en general son categorías gramaticales comunes a todas las lenguas. Podemos definirlas de la siguiente forma:

Sustantivo. Es una palabra que refiere a un ente concreto o idea. Por ejemplo, las palabras 'carro', 'gato', 'libertad', etc.

Adjetivo. Es una palabra cuya función es calificar a un sustantivo; verbigracia: 'bonito', 'rojo', 'grande', 'chico', etc.

Verbo. Es una palabra que refiere a una acción. Algunos ejemplos son 'correr', 'tomar', 'soñar', entre otras.

7.1. Etiquetado gramatical

Para el etiquetado de partes de la oración es común ver al lenguaje como un proceso estocástico (Harris, 1968). En general, podemos determinar que se trata de una cadena de eventos (palabras) que determinan en gran medida la presencia de palabras subsiguientes.

Definición 36 (Propiedad de Markov de orden r). *Sean X_1, \dots, X_n una cadena de eventos aleatorios. Se dice que cumple la propiedad de Markov de orden r si:*

$$P(X_n | X_1, \dots, X_{n-1}) = P(X_n | X_{r+1}, \dots, X_{n-1}) \quad (3)$$

Sin embargo, en la práctica es conveniente reducir esta propiedad a la, simplemente, llamada *propiedad de Markov*.

³Este término va a ser importante, ya que se utiliza en la literatura y define al etiquetado de categorías gramaticales o POST (*parts of speech tagging*)

Definición 37 (Propiedad de Markov). *Sean X_1, \dots, X_n una cadena de eventos aleatorios. Se dice que cumple la propiedad de Markov si:*

$$P(X_n|X_1, \dots, X_{n-1}) = P(X_n|X_{n-1}) \quad (4)$$

De esta forma, podemos visualizar una cadena del lenguaje como determinada por el elemento inmediatamente anterior. Por ejemplo, si tenemos la cadena “el”, los elementos que aparecerán a continuación están limitados a sólo aquellos que correspondan a un sustantivo, masculino, singular; es decir, palabras como “perros” o “araña” tendrán pocas probabilidades de aparecer.

Con base en esto, surge la idea del etiquetado de Partes de la Oración (Part Of Speech o POS). Pues supóngase que se tiene la cadena “el gato”. Si sabemos que la etiqueta correspondiente a “el” es *ART* (por artículo), es altamente probable que la etiqueta correspondiente a “gato” sea *SUST* (por sustantivo). Para esto, necesitamos entender el concepto de Modelo Oculto de Markov.

Definición 38 (Modelo Oculto de Markov). *Un Modelo Oculto de Markov es un sistema $HMM = (A, B, \Pi, S, O)$, tal que $S = \{s_1, \dots, s_n\}$ son estados de transición; $O = \{o_1, \dots, o_m\}$ son observaciones; $\Pi = \{\pi_1, \dots, \pi_k\}$ son probabilidades iniciales; $A = P(s_i|s_j)$ es la matriz de transición; y $B = P(s_i|o_j)$ es la matriz de probabilidades de emisión.*

7.1.1. Etiquetado POS con HMM

Los pasos para el etiquetado POS se describen a continuación. Para realizar

1. **Creación de un modelo a partir del texto.** A partir de los datos textuales se genera un modelo $\mu = (A, B, \Pi)$, donde A, B y P_i son los elementos arriba definidos. En este caso, A es la matriz conformada por las probabilidades de la secuencia de etiquetas dada la propiedad de Markov; es decir, si t_i es la etiqueta correspondiente a la palabra w_i , entonces $a_{ij} = P(t_i|t_{i-1})$. Por su parte, B tendrá las observaciones o palabras tal que $b_{ij} = P(t_i|w_i)$. Finalmente, Π será un vector contenido las probabilidades de todas las etiquetas iniciales, esto es $\pi_i = P(t_i|i \leq j \forall j \in \{1, \dots, n\})$.
2. **Determinar mejor etiqueta inicial.** Para determinar la etiqueta inicial se necesita encontrar:

$$\arg \max_t \pi_i P(t_i|w_i) \quad (5)$$

y será nuestro punto de inicialización.

3. **Determinar la mejor secuencia de etiquetas.** Dada la etiqueta anterior t_{i-1} , se quiere determinar la etiqueta subsiguiente óptima. Para esto, se debe encontrar:

$$\arg \max_t \prod_{i=1}^n P(t_i|w_i) P(t_i|t_{i-1}) \quad (6)$$

y repetir este proceso hasta que se obtenga la cadena de etiquetas óptima.

7.2. Algoritmo de Viterbi

Dado que la complejidad de un algoritmo como el muestro puede escalar rápidamente, es común utilizar el algoritmo de Viterbi.

Inicialización.

$$\delta_i(1) = \pi_i, i \leq i \leq n$$

Inducción

$$\delta_i(t+1) = \max_i \delta_i(t) P(t_i|w_i) P(t_i|t_{i-1})$$

Guardar:

$$\psi_i(t+1) = \arg \max_i \delta_i(t) P(t_i|w_i) P(t_i|t_{i-1})$$

Terminación

$$\hat{t} = \arg \max_i \delta_i(T+1) P(\hat{t}) = \max_i \delta_i(T+1)$$

A partir de este algoritmo, se reducen los caminos posibles a tomar en la cadena de etiquetas, por lo que la complejidad se reduce.

8. Sintaxis

En la morfosintaxis nos ha interesado el estudio de la función de las palabras en una oración. Esto está ampliamente ligado con la sintaxis, pues ésta se enfoca precisamente en las oraciones.

Definición 39 (Sintaxis). *Es el nivel de la lengua que se dedica al estudio de la estructura de las oraciones.*

Para la sintaxis son importantes los términos de «Frase nominal» y «Frase verbal»⁴, que se pueden entender de la siguiente forma:

Frase nominal. Es una estructura cuyo núcleo es un sustantivo. Por ejemplo: 'el gato feo', 'el perro viejo', 'los niños tontos', etc.

Frase verbal. Es una estructura lingüística cuyo núcleo es un verbo. Ejemplos de esto son: 'comí', 'me desperte', 'se cayó', 'maté un mosquito', 'le di una rosa roja'.

Como se ve, la frase verbal puede estar acompañada por una frase nominal. Para entender mejor cómo funciona esto, deben introducirse otros términos básicos en la sintaxis: sujeto, objeto directo y objeto indirecto.

Sujeto. Es la frase nominal que realiza la acción de un verbo dentro de una oración. Por ejemplo en 'Juan comió la sopa' el sujeto es 'Juan'. También en 'El perro viejo se cayó por la escalera' el sujeto es 'el perro viejo' pues estas frases nominales son las que realizan la acción del verbo.

Objeto directo. Es la frase nominal que recibe la acción de un verbo. En el ejemplo de 'Juan comió la sopa', 'la sopa' está funcionando como objeto directo. Otro ejemplo, está en la oración 'El lobo mató a la oveja' donde '(a) la oveja' es el objeto directo.

Antes de explicar lo que es un objeto indirecto, cabe señalar que, además de frases nominales y verbales, existen otros tipos de frases, las frases preposicionales, que, como su nombre lo dice, su núcleo es una preposición:

Frase preposicional. Es una estructura lingüística cuyo núcleo es una preposición. Esta formado por el núcleo (la preposición) y, generalmente, una frase nominal. Por ejemplo: 'hacia la estación', 'en el parque', 'para mis amigos', etc.

De esta forma, podemos pasar a definir lo que es un objeto indirecto:

Objeto indirecto. Es la frase preposicional⁵ que es beneficiaria de una acción.

Por ejemplo en 'Juan le dio una manzana a la profesora', 'a la profesora' es el objeto indirecto, también en 'Pedro compró flores para su abuela' el objeto indirecto es 'para su abuela'.

⁴También conocidos, en lingüística, como sintagma nominal y sintagma verbal.

⁵Aunque no en todas las lenguas se manifiesta como frase preposicional, como en el inglés: 'I give *her* a book'.

En general, el análisis sintáctico que se realizará en este curso puede llevarse a cabo a partir de los elementos descritos. Este análisis es jerárquico y parte de la oración. Por tanto podemos definir una oración, O , de la siguiente forma:

$$O = (FN) + FV \quad (7)$$

Donde FN refiere a frase nominal y FV a frase verbal. A su vez, se puede decir que:

$$FV = V + (FN) + (FP) \quad (8)$$

Donde V es un verbo y FP frase preposicional. Debe notarse que la diferencia entre la FN de O y la FN de FV es que en O , la FN funje como sujeto, mientras que en FV su función es de objeto directo. Los elementos entre paréntesis pueden o no aparecer en una oración: son opcionales.

También la FN y la FP pueden ser descritos de la siguiente forma:

$$FN = (Art) + NN + (Adj) \quad (9)$$

$$FP = PP + FN \quad (10)$$

Aquí adoptamos las etiquetas de la sección de morfosintaxis, donde Art es un artículo, NN un sustantivo, Adj un adjetivo y PP una preposición.

De esta forma, podemos ejemplificar a partir de la siguiente oración:

El muchacho alto compró flores para María

Primero debemos identificar las categorías gramaticales de cada palabra. Podemos etiquetarlas de la forma *palabra_categoria*, así:

*El_Art muchacho_NN alto_Adj compr_V flores_NN para_PP
Mara_NN*

De esta forma, por como hemos definido las frases, podemos ver que 'El muchacho alto' es una FN , assimismo 'flores' es otra FN y también lo es 'María'. Pero para también se forma una FP con 'para María' y una FV con 'compró flores para María'. Y todo junto forma una oración. Un análisis arboreo de esta oración puede verse en la Figura 1.

8.1. Gramáticas Libres de Contexto Probabilísticas

Dentro del análisis sintáctico, se puede obtener una jerarquización de los elementos a partir de un modelo de aprendizaje y basados, también, en un etiquetado POS a partir de una gramática libre de contexto probabilística.

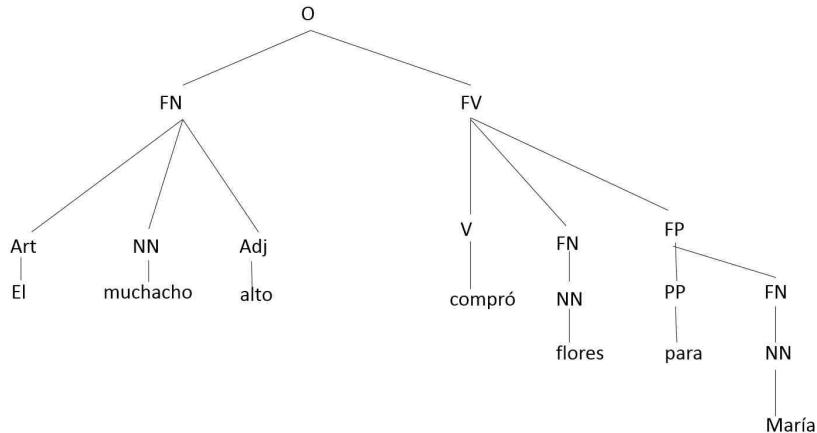


Figura 1: Análisis sintáctico de la oración 'El muchacho alto compró flores para María'.

Definición 40 (Gramática Libre de Contexto Probabilística). *Una Gramática libre de contexto probabilística o PCFG es una cinco-tupla $PCFG = (q, Q, \Sigma, T)$ donde $q \in Q$ es un símbolo de inicio, $Q = \{q_1, \dots, q_n\}$ es el conjunto de símbolos no terminales, Σ es el conjunto de símbolos terminales, $T = (\{q_i \mapsto q_j\}, P)$ es un conjunto de reglas y $P(q_i \mapsto q_j | q_i)$ es una función probabilística tal que $\sum_{j=1}^n P(q_i \mapsto q_j) = 1$.*

Entonces, dada una cadena de palabras $w_{1,n}$ y un árbol jerárquico $G(w_{1,n})$ como el de la figura 1, la probabilidad de la cadena es:

$$P(w_{1,n}) = \sum_G P(w_{1,n}|G(w_{1,n})) \quad (11)$$

9. Semántica

Una de las áreas que presenta mayor complicación dentro del PLN es el procesamiento del significado. El significado se extiende a diferentes unidades lingüísticas. Ya habíamos dicho que el morfema es la unidad mínima con significado; asimismo, las palabras formadas por estos morfemas también tienen significado, y éstas a su vez forman oraciones con significado, y las oraciones forman discursos que también contienen un significado. En general, podemos definir semántica de la siguiente forma.

Definición 41 (Semántica). *La semántica es la rama de la lingüística que se dedica al estudio del significado.*

Una forma de ver el significado es a través de un signo lingüístico. Con este signo podemos representar el **significado** que es el concepto que lleva una construcción lingüística y el **significante** que es la construcción lingüística misma (lo que tiene un significado). Gráficamente lo podemos ver con la Figura 2.

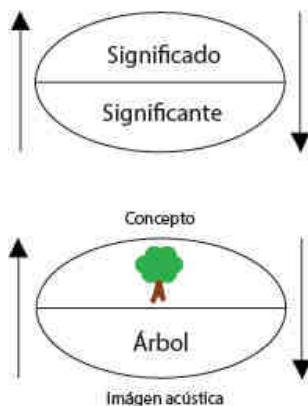


Figura 2: Signo lingüístico: concepto y representación acústica.

Por ejemplo, en la palabra 'árbol' el significado es la imagen abstracta o, dicho de otra forma, el concepto que tenemos en nuestras mentes sobre los árboles, mientras que el significante es la producción concreta, acústica o escrita, de la palabra.

9.1. Ontologías y taxonomías

Existen muchas representaciones de la semántica tanto dentro de la lingüística como dentro del PLN. Primero, adaptaremos la visión de las ontologías, más adelante usaremos también el concepto de espacio vectorial para tratar de capturar la semántica.

Una ontología es un grafo (una especie de red) en donde diferentes elementos lingüísticos (como palabras) se conectan a partir de las relaciones léxicas que

establecen entre ellas. Para entender mejor esto, defínanse antes algunos tipos de relaciones léxicas:

Sinonimia. Se dice que una palabra es sinónimo de otra cuando se usa en los mismos contextos y con el mismo significado. Por ejemplo: 'ordenador' y 'computadora' que bien se pueden usar en el contexto de 'Navegué por internet a través del ordenador/computadora'.

Antonimia. Consideramos un antónimo a aquella palabra que se opone al significado de una palabra dada. Por ejemplo, 'bueno' y 'malo' son antónimos, como 'alto' y 'bajo'.

Hiperonomia. Un hiperónimo es una palabra que abarca a otra palabra dada. Por ejemplo, 'felino' es hiperónimo de 'gato'. Por su parte 'gato' es **hipónimo** de 'felino'.

Se puede, entonces, definir una ontología como sigue.

Definición 42 (Ontología). *Una ontología es un grafo complejo que describe, estructuradamente, el conocimiento de un dominio particular*

En una ontología, generalmente, se relacionan palabras a partir de relaciones léxicas. Existen diferentes niveles de especialización de una ontología, dependiendo de si sus elementos (las palabras, en este caso) son de un ambiente general (que abarque diferentes áreas del conocimiento) o más detalladas (específicas o de especialidad, por ejemplo una ontología del área de computación o de lingüística). A las ontologías más generales se les conoce como *ontologías de alto nivel*, mientras que a las más específicas se les llama *ontologías de aplicación*.

En este curso nos enfocaremos a un tipo particular de ontologías, que son las taxonomías. En una taxonomía, además de las relaciones, se tiene una jerarquía en los elementos que componen a la ontología. Es decir, los elementos tienen un rango mayor o menor.

Definición 43 (Taxonomía). *Es una ontología en la que sus elementos se ordenan jerárquicamente.*

Una taxonomía se puede ver gráficamente, como en la Figura 3; entonces, una taxonomía es una forma jerárquica de representar las relaciones léxicas entre palabras.

Es importante definir cuáles son las partes de una taxonomía:

Conceptos. Son las representaciones de un grupo de individuos distintos que comparten características. En la Figura 3, los conceptos son todos los elementos que tienen una palabra por debajo, como 'COSA', 'felino' 'arachnida', etc. Ya que todos estos pueden bien representar una clase de palabras, las cuales se representarán jerárquicamente por debajo de ellas.

Individuos. Son los objetos concretos o abstractos descritos por la taxonomía.

En este caso, los individuos pueden ser los elementos que no tienen nada por debajo de ellos: 'gato', 'perro', 'alacrán' y 'quelite'.

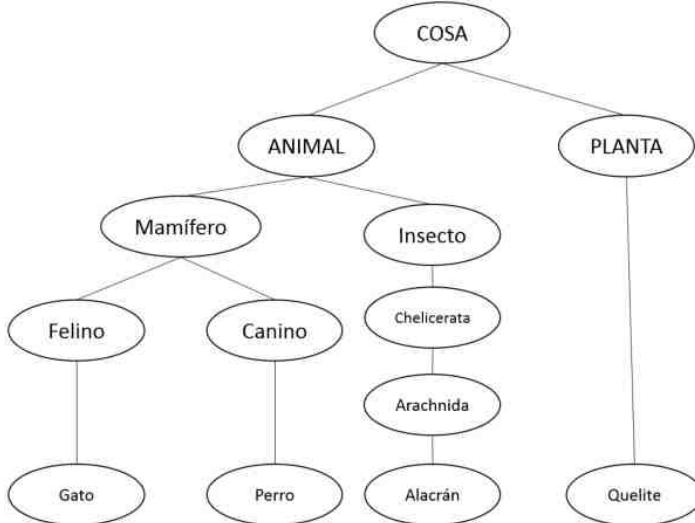


Figura 3: Ejemplo de una taxonomía.

Relaciones. Describen la forma en que los elementos de la taxonomía se relacionan entre sí. Bien pueden ser las relaciones léxicas que establecemos. En este caso se representan por las líneas o aristas. Sabemos, por ejemplo, que 'felino' es hiperónimo de 'gato' y que los conceptos de 'animal' y 'planta' son antónimos.

9.1.1. Similitud taxonómica

En términos cuantitativos, nos interesa medir qué tan relacionadas están dos palabras. Para esto, dada una taxonomía como la de la Figura 3, vemos que 'perro' y 'gato' están más relacionados que 'gato' y 'quelite'. Esto lo podemos ver al ver los conceptos que los conectan y las relaciones que establecen entre sí. Así, 'perro' y 'gato' comparten el concepto de 'animal' y 'mamífero' y ambos están conectados con estos conceptos por medio de aristas (líneas) pues son hiperónimos. Podemos entonces aplicar la siguiente fórmula para cuantificar esta similitud:

$$sim_T(w_1, w_2) = \frac{2 \cdot \#Conceptos(w_1, w_2)}{\#Relaciones(w_1) + \#Relaciones(w_2)} \quad (12)$$

De esta forma, vemos que 'gato' y 'perro' comparten 3 conceptos (los círculos por encima de ellos que tienen en común), que 'gato' tiene 4 relaciones (el número de líneas para llegar desde lo más alto hasta la palabra) y 'perro' tiene también 4. Entonces, sustituyendo en la fórmula, tenemos:

$$sim_T(gato, perro) = \frac{2 \cdot 3}{4 + 4} = \frac{6}{8} = 0,75$$

Es decir, tienen una similitud más grande que la que se da entre 'gato' y 'quelite' que es:

$$sim_T(gato, quelite) = \frac{2 \cdot 1}{4 + 2} = \frac{2}{6} = 0,33$$

10. Leyes empíricas del lenguaje

Definición 44 (Ley de Zipf). *Sea r el rango de una palabra en un corpus y f su frecuencia. Entonces:*

$$f \propto \frac{1}{r^\alpha}$$

Donde α es una constante.

Definición 45 (Fórmula de Mandelbrot). *Sea r el rango de una palabra en un corpus dado y sea f su frecuencia dentro de este corpus. Entonces:*

$$f \propto \frac{P}{(r + \rho)^{-\alpha}}$$

Donde α es una constante, y P, ρ son parámetros del texto.

Definición 46 (Ley de Herdan). *Sea N el tamaño de un corpus dado (número de tokens) y t el número de tipos en este corpus. Entonces:*

$$t \propto N^{\alpha^{-1}}$$

Donde α es una constante.

Definición 47 (Función Beta). *Sea r el rango de una palabra en un corpus dado y sea f su frecuencia dentro de este corpus. Entonces:*

$$f \propto \frac{(t + a - r)^b}{r^\alpha}$$

Definición 48 (Función de Yule). *Sea r el rango de una palabra en un corpus dado y sea f su frecuencia dentro de este corpus. Entonces:*

$$f \propto \frac{b^r}{r^\alpha}$$

Definición 49 (Función de Menzerath-Altman). *Sea r el rango de una palabra en un corpus dado y sea f su frecuencia dentro de este corpus. Entonces:*

$$f \propto r^b \cdot e^{-\alpha/r}$$

11. Representación en espacios vectoriales

Resulta de mayor facilidad realizar un análisis de los datos textuales cuando se tiene una estructura conocida. En este caso, es común utilizar espacios vectoriales, donde un punto en ese espacio representa un documento, una palabra o cualquier elemento lingüístico de interés. Una forma muy sencilla de obtener datos numéricos a partir de textuales surge a partir de la *función de conteo*.

Definición 50 (Función de conteo). *La función de conteo $C(\cdot)$ es la función que asigna un número natural a cada tipo en el corpus.*

Ejemplo 4. Téngase el siguiente *corpus*:

d_1 : *El cacomixtle es arborícola, nocturno y de naturaleza solitaria.*

d_2 : *El cacomixtle tiene naturaleza nocturna.*

d_3 : *El gato es un felino doméstico solitario.*

La función de conteo corresponde a la frecuencia de cada tipo dentro del corpus. Así, $C(\text{cacomixtle}) = 2$ o bien $C(\text{gato}) = 1$.

Ejemplo 5. Una primera intuición sobre como trasladar los documentos a vectores sería a través de los elementos de la función de conteo, de tal forma que tendríamos los siguientes vectores:

$$d_1 = (3 \ 2 \ 2 \ 1 \ 2 \ 1 \ 1 \ 2)$$

$$d_2 = (3 \ 2 \ 1 \ 2 \ 2)$$

$$d_3 = (3 \ 1 \ 2 \ 1 \ 1 \ 1 \ 2)$$

Sin embargo, estos elementos no se encuentran en el mismo espacio y por tanto no se pueden realizar operaciones entre ellos. Es por tanto necesario realizar vectores que correspondan al mismo espacio. Una forma de ello es mediante la siguiente matriz A. Para hacer más corto el ejemplo, eliminemos las palabras funcionales.

$$A = \begin{matrix} & \text{cacomixtle} & \text{aboricola} & \text{nocturn(o|a)} & \text{naturaleza} & \text{solitaria} & \text{gato} & \text{felino} & \text{domestico} \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \end{matrix} & \left(\begin{array}{ccccccc} 2 & 1 & 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 & 1 \end{array} \right) \end{matrix}$$

donde cada columna corresponde al vector que representa a cada documento. La matriz A está dada, entonces por:

$$A = a_{ij} = \begin{cases} 0 & \text{si } w_i \notin d_j \\ C(w_i) & \text{s } w_i \in d_j \end{cases}$$

De esta forma, a simple vista podemos ver que d_1 y d_2 se asemejan más entre sí, que cualquiera de ellos con d_3 . Sin embargo, surgen diversos problemas:

1. El más evidente de ellos es que cuando estamos tratando con documentos reales, la cantidad de tipos, a pesar de eliminar palabras funcionales, incrementa de forma drástica. Esto nos deja con espacios vectoriales de alta dimensión. Se vuelve necesario entonces encontrar métodos que nos permitan trabajar con elementos de tan altas dimensiones.
2. La selección de los elementos que conformarán las entradas del vector no siempre corresponderán a palabras. Se necesita hacer un análisis de las características que mejor representen los documentos según la tarea que deseamos emprender. Es decir, en este caso, los tipos pueden ser un buen indicador de la temática del documento, pero no así del estilo o el autor de éste.
3. Las meras frecuencias absolutas son muy dependientes del tamaño del corpus y otros factores que pueden meter ruido a la formación de los vectores. Por tanto, se necesitan relativizar tales frecuencias y encontrar formas de señalar relaciones textuales específicas.

De esta forma, se pueden y se han desarrollado diferentes formas de representación textual en espacios vectoriales. Aquí analizaremos algunas de ellas que consideramos relevantes.

12. Bolsa de palabras

En general, se puede pensar que dado un documento d , la función de conteo nos puede decir mucho sobre las palabras que lo caracterizan. La forma en que queremos comunicar una idea se da a través de hacer hincapié en elementos léxicos relevantes; es decir, en palabras que remarquen esa idea (Luhn, 1957). Estas palabras son los términos. En principio, podemos pensar que estas palabras tendrán una frecuencia alta en el documento.

Proposición 1. *El peso de un término en un documento d es proporcional a la frecuencia de los términos en ese documento.*

Es decir, la palabra con mayor frecuencia tendrá mayor peso y la de menor frecuencia será menos pesada. De esta forma, podemos definir una frecuencia ponderada de la siguiente forma:

Definición 51 (Term Frequency (TF)). *Sea $d = \{w_1, \dots, w_n\}$ un documento formado por n palabras. Entonces, la frecuencia ponderada de un término está dada por:*

$$tf(w) = c + (1 - c) \frac{C(w : d)}{\max_i \{C(w_i ; d)\}} \quad (13)$$

donde $c \in [0, 1]$.

Sin embargo, esta frecuencia no nos servirá para ponderar un término, pues según la ley de Zipf los términos con mayor frecuencia son palabras funcionales, como “el”, “de”, etc. Sin embargo, notaremos que estos términos aparecen en prácticamente cualquier documento. Es decir, se distribuyen uniformemente a través de un corpus, a diferencia de términos específicos de una temática.

Proposición 2. *La especificidad temática de un término estará dada por una función inversa de su aparición a través de los documentos de un corpus.*

Por tanto, podemos definir la frecuencia inversa de un término a través de documentos de la siguiente manera:

Definición 52 (Inverse Document Frecuency (IDF)). *Siendo $\mathfrak{C} = \{d_1, \dots, d_k\}$ un corpus conformado de k documentos, la frecuencia inversa de un término está dada por:*

$$idf(w) = \log\left(\frac{|\mathfrak{C}|}{|\{d_i : w \in d_i\}|}\right) \quad (14)$$

De esta forma, podemos concluir la siguiente proposición.

Proposición 3. *El comportamiento estadístico de un término está dado en función de su frecuencia ponderada (tf) en un documento y su frecuencia inversa dentro del corpus (idf).*

Es decir, una palabra temáticamente relevante será una palabra que tenga mucha frecuencia en un documento sobre el tema, pero que tendrá bajafrecuencia en documentos de otra temática. De esta forma, para ponderar una palabra según su temática usaremos el producto entre tf e idf . De esta forma definiremos el modelo de Bolsa de Palabras (BoW).

Definición 53 (Bolsa de palabras). *El modelo de bolsa de palabras es la representación vectorial de un corpus a partir de una matriz del tipo:*

$$[\mathfrak{C}] = c_{ij} = \begin{cases} 0 & \text{si } w_i \notin d_j \\ tf(w_i) \cdot idf(w_i) & \text{si } w_i \in d_j \end{cases} \quad (15)$$

Donde cada vector $d = [\mathfrak{C}]_j$ de un documento será el vector renglón correspondiente de $[\mathfrak{C}]$.

13. Modelo distribucional

El modelo distribucional de representación en espacios vectoriales o **DSM** surge a partir de las ideas de la distribución de las palabras y su composicionalidad planteadas por el lingüista Zellig Harris, el filósofo Ludwig Wittgenstein y el lógico, matemático y filósofo Gottlob Frege. En términos muy generales, la hipótesis distribucional plantea lo siguiente:

Proposición 4 (Hipótesis distribucional). *Las palabras con características semánticas similares se distribuyen en contextos similares.*

Realmente esta proposición se basa en los siguientes puntos:

1. Los elementos del lenguaje no ocurren de forma arbitraria, sino relativamente con respecto a otros elementos.
2. La distribución de clases es persistente a través de las ocurrencias de un mismo elemento.
3. Es posible determinar posiciones relativas de elementos respecto a otros.
4. Pueden describirse restricciones de ocurrencias relativas de los elementos.

Ejemplo 6. Consideréense los siguientes contextos:

- *Ayer fuí con un * para que me revisara la vista.*
- *El * me dijo que mi vista estaba dañada.*

*Aquí las probabilidades de que * correspondan a “oculista” o una palabra relacionada son altas, y aún más si se toman los dos contextos juntos. Uno de los elementos que motivan esta selección es la aparición de la palabra “vista”*

El ejemplo anterior nos da una idea de lo que debemos entender por contexto. A continuación presentamos una definición de esto.

Definición 54 (Contexto). *Un contexto de un elemento w es un arreglo de sus co-ocurrencias; esto es, los otros elementos, cada uno en una posición particular, con los que w ocurre.*

La delimitación de un contexto se puede dar a partir de diferentes formas:

- Un contexto natural es la palabra, los elementos que aparecen en ella pueden ser morfemas o caracteres.
- Otro contexto natural es la oración, sus elementos son las palabras. Así mismo, otros contextos naturales son el párrafo y el documento.
- Un contexto común es la ventana de $n \times n$; es decir, se seleccionan $n - 1$ elementos antes de w y $n - 1$ elementos posteriores

Notación 2. Denotaremos al contexto de un elemento w como $N(w)$.

A partir de esto, se propone construir **perfles distribucionales** basados en los siguientes dos puntos:

1. Basados en qué palabras ocurren alrededor de w .
2. Basados en qué región del texto ocurre w .

El perfil distribucional más común está dado por 1, pues la forma en que se puede relativizar las posiciones es más sencilla con respecto a 2.

Ejemplo 7. Téngase el corpus:

d_1 : El perro persiguió al gato.

d_2 : El carro atropeló al perro.

d_3 : El carro atropeló al gato.

d_4 : El carro chocó con el árbol.

Y considérese el contexto como cada $d_i, i = \{1, 2, 3, 4\}$ (en otras palabras, $\forall j \in \{1, \dots, n\} \exists i \in \{1, 2, 3, 4\} : d_i = N(w_j) \iff w_j \in d_i$). Entonces, la palabra w_j comparte contexto con w_k si y sólo si son parte del mismo d_i . De esta forma, podemos construir los perfles distribucionales a partir de sus co-ocurrencias en dichos contextos, de tal forma que obtenemos la matriz:

$$\begin{array}{cccc} & \text{perro} & \text{gato} & \text{carro} & \text{árbol} \\ \text{perro} & \left(\begin{array}{cccc} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right) \\ \text{gato} & & & & \\ \text{carro} & & & & \\ \text{árbol} & & & & \end{array}$$

En el ejemplo anterior, podemos tomar cada vector columna o vector renglón como la representación en \mathbb{R}^n de una palabra. De esta forma, podemos definir un espacio distribucional (DS) de la siguiente forma:

Definición 55 (DS simple). Si n es el número de tipos w en un corpus, un DS es una matriz A de $n \times n$ tal que:

$$A = a_{ij} = \begin{cases} 0 & \text{si } w_i \notin N(w_j) \\ C(w_i, w_j) & \text{si } w_i \in N(w_j) \end{cases} \quad (16)$$

13.1. Hyperspace Analogous to Language (HAL)

En base a lo expuesto anteriormente, Lund y Burgess desarrollaron una metodología para crear un DSM donde no sólo se usarán las co-ocurrencias de las palabras si no que se ponderarán en relación a la distancia de la palabra núcleo. El algoritmo toma una ventana de n palabras a la izquierda y a la derecha como contexto. A este espacio se le denomina HAL.

Definición 56 (HAL). *El modelo HAL de DSM es aquel en el que una palabra núcleo w_0 se encuentra en una ventana de $n \times n$. Esto es:*

$$w_1 \dots w_k \dots w_n - w_0 - w_n \dots w_k \dots w_1$$

tal que:

$$A = a_{ij} = \begin{cases} 0 & \text{si } w_i \notin N(w_j) \\ k \cdot C(w_i, w_j), k \in \{1, \dots, n\} & \text{si } w_i \in N(w_j) \end{cases} \quad (17)$$

Es decir, si se toma una ventana de 5 palabras, si la palabra está inmediatamente al lado de la palabra núcleo, por cada co-ocurrencia se suma 5 al conteo total. Si la palabra aparece al extremo de la ventana sólo se suma 1 por cada co-ocurrencia.

Ejemplo 8. Téngase la siguiente oración:

El perro jugó con el gato

Y si tomamos $n = 3$, se tendrá la siguiente matriz:

$$\begin{array}{ccccc} & \text{el} & \text{perro} & \text{jugo} & \text{gato} & \text{con} \\ \text{el} & \left(\begin{array}{ccccc} 0 & 4 & 4 & 3 & 3 \\ 3 & 0 & 3 & 0 & 2 \\ 3 & 3 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 & 2 \\ 3 & 2 & 3 & 2 & 0 \end{array} \right) \\ \text{perro} & & & & & \\ \text{jugo} & & & & & \\ \text{gato} & & & & & \\ \text{con} & & & & & \end{array}$$

13.2. Perspectiva con Información Mútua

Las dos perspectivas anteriores utilizan conteos de frecuencias absolutas para crear el DMS. Si buscamos una metodología que nos permita relativizar estas frecuencias, es común usar las ideas de la teoría de la información para determinar la relación entre una palabra y otra. Para esto, definiremos la información mutua y conceptos relacionados.

Definición 57 (Entropía relativa). *La entropía relativa o divergencia de Kullback-Liebler es la distancia entre dos distribuciones probabilísticas dada por:*

$$D(p||q) := \sum_{x \in \chi} p(x) \frac{p(x)}{q(x)} \quad (18)$$

De esta forma, lo que queremos establecer es la entropía relativa entre una distribución dependiente y otra independiente de las palabras en cuestión. Por tanto, queremos obtener $D(P(x,y)||P(x)P(y))$ lo que da paso a la definición de información mútua.

Definición 58 (Información mútua). *Sean X e Y dos variables aleatorias. La información mútua entre ambos se define como:*

$$MI(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \quad (19)$$

Como se ve, la información mútua es la entropía relativa a las dsitribuciones de probabilidad conjunta $P(X,Y)$ y la distribución independiente $P(X)P(Y)$. Podemos, además, encontrar una analogía entre la información mútua y la entropía.

Proposición 5. *Siendo $H(X)$ la entropía de la variable aleatoria X y $H(X|Y)$ la entropía relativa de X dado Y , tenemos:*

$$MI(X;Y) = H(X) - H(X|Y) \quad (20)$$

Demostración. Tenemos que:

$$MI(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \quad (21)$$

$$= \sum_{x,y} P(x,y) \log \frac{P(x|y)}{P(x)} \quad (22)$$

$$= - \sum_{x,y} P(x,y) \log P(x) + \sum_{x,y} P(x,y) \log P(x|y) \quad (23)$$

$$= - \sum_x P(x) \log P(x) - \left(- \sum_{x,y} P(x,y) \log P(x|y) \right) \quad (24)$$

$$= H(X) - H(X|Y) \quad (25)$$

□

De esta forma, podemos entender a la información mútua como el cuánto se reduce la incertidumbre de una variable X dada la ocurrencia de otra variable Y .

En el DSM, la información mútua nos sirve para determinar una relación relativa entre las palabras que ocurren en un contexto. Sin embargo, la información mútua toma en consideración diferentes eventos de una variable aleatoria. En el caso de las palabras, tenemos sólo eventos individuales. Si sabemos que la información de un evento está dada por $I(x) = -\log P(x)$, entonces podemos definir una información mútua con esta idea y con la ecuación (24), la cual llamaremos punto de información mútua.

Definición 59 (Punto de información mútua). *Siendo x e y eventos, su punto de información mútua está dado por:*

$$PMI(x; y) = \log \frac{P(x, y)}{P(x)P(y)} \quad (26)$$

Está fórmula se puede entender como $I(x) - I(x|y)$, y nos está dando la información que comparten el elemento x con el evento y . De esta forma, podemos proponer un nuevo DSM.

Definición 60 (PMI-DSM). *Siendo $w_i, i = 1, \dots, n$ palabras en un corpus, un DSM basado en PMI está dado por una matriz A tal que:*

$$A = a_{ij} = PMI(w_i; w_j) \quad (27)$$

En este caso, debe tomarse en consideración lo siguiente:

- $\log 0 = 0$
- $PMI(w_i; w_j) \neq 0 \iff P(x, y) > 0 \iff x \in N(y) \iff y \in N(x)$

Ejemplo 9. Tómese como ejemplo el siguiente corpus:

d_1 : El avión despegó de la pista.

d_2 : El hombre fabrica aviones y coches.

d_3 : El hombre nada con la mujer.

d_4 : Los coches compiten en la pista.

Entonces, tomando como contexto cada oración, la matriz resultante es la siguiente:

$$\begin{array}{c} & \text{avion} & \text{pista} & \text{hombre} & \text{mujer} & \text{coche} \\ \text{avion} & \left(\begin{array}{ccccc} 0 & 0,7 & 0,7 & 0 & 0,7 \\ 0,7 & 0 & 0 & 0 & 1,02 \\ 0,7 & 0 & 0 & 0,7 & 0,7 \\ 0 & 0 & 0,7 & 0 & 0 \\ 0,7 & 1,02 & 0,7 & 0 & 0 \end{array} \right) \\ \text{pista} & & & & & \\ \text{hombre} & & & & & \\ \text{mujer} & & & & & \\ \text{coche} & & & & & \end{array}$$

14. Word embeddings

Un modelo que ha venido tomando impulso en los últimos años es el de word embeddings. Este modelo se basa en los llamados modelos del lenguaje neuronales y el embedido stocástico de vecinos.

14.1. Embedido estocástico de vecinos

El embedido estocástico de vecinos o **SNE** (por sus siglas en inglés: Stochastic Neighbor Embedding) busca determinar la probabilidad de que un elemento sea un vecino de otro elemento en un espacio vectorial. Antes que nada, necesitamos determinar el concepto de vecino.

Definición 61 (Vecino). *Sea $X = \{x_1, \dots, x_t\}$ un conjunto de objetos en un espacio m dimensional. Un vecino de un elemento x_i es otro elemento del conjunto tal que:*

$$\delta(x_i, x_j) \leq \alpha \quad (28)$$

donde $\delta : X \times X \rightarrow \mathbb{R}$ es una función de distancia y $\alpha \in \mathbb{R}$ es la distancia máxima entre x_i y x_j .

Entonces, lo que busca el SNE es determinar una probabilidad de que un elemento x sea vecino de un elemento y ; es decir, se busca $P(y \in N(x))$. Para esto, se define una probabilidad asimétrica.

Definición 62 (Probabilidad asimétrica de vecinos). *Por cada objeto $x_i \in X, i \in \{1, \dots, t\}$ y por cada posible vecino $x_j \in X, j \in \{1, \dots, m\}$, la probabilidad asimétrica de que x_i elija a x_j como su vecino es:*

$$P(x_j \in N(x_i)) := \frac{\exp(-\delta(x_i, x_j)^2)}{\sum_{k \neq i} \exp(-\delta(x_i, x_k)^2)} \quad (29)$$

donde $\delta : X \times X \rightarrow \mathbb{R}$ es una función.

En general, este proceso se usa para reducción de dimensionalidad, donde se usa una segunda función (de distribución) q en un espacio de baja dimensionalidad. Lo que se busca es reducir una función de costo entre la distribución q y la distribución de los datos originales p que están en un espacio de dimensionalidad alta. Sin embargo, para los propósitos de esta sección, es suficiente tener esto presente.

14.2. Modelo de lenguaje neuronal

Cómo hemos visto más arriba, el lenguaje puede modelarse a partir de procesos de Markov de orden r . En general, los modelos de Markov nos permiten ver la probabilidad de una cadena como el productorio de los estados que componen dicha cadena. Lo hemos visto para etiquetas POS. Ahora, si en lugar de tener etiquetas POS como estados usamos las palabras, podemos definir lo que se llama **modelo de n-gramas**.

Definición 63 (Modelo de n-gramas). *Dado una cadena de palabras $w_{1,n} = \{w_1, \dots, w_t\}$ el modelo del lenguaje determina la probabilidad de la cadena a partir de la probabilidad de los estados de la cadena. De tal forma que:*

$$P(w_{1,t}) = \prod_{i=1}^t P(w_i | w_{i-n+1} \dots w_{i-1}) \quad (30)$$

donde $n \in \mathbb{N}$ es el tamaño de la ventana que se toma.

Se trata de un proceso de Markov de orden r donde $r = n$. Asimismo, si tomamos $n = 2$, (30) se convierte en un proceso de Markov, tal que la ecuación (30) se transforma en:

$$P(w_{1,t}) = \prod_{i=1}^t P(w_i | w_{i-1}) \quad (31)$$

Bengio (2003) se basa en estos modelos para proponer lo que él llama un **modelo del lenguaje neuronal**. Esta idea se basa en tres puntos principales.

1. Representar a cada palabra en un vocabulario a partir de un vector distribuido de rasgos (un vector con entradas en \mathbb{R}).
2. Expresar la función de probabilidad conjunta de secuencia de palabras en términos de los vectores de rasgos de las palabras en la secuencia.
3. Aprender simultáneamente los vectores de rasgos y los parámetros de la función de probabilidad.

La función de probabilidad en este caso es del tipo expresado en la ecuación (30). Ahora, sin embargo, se usa una red neuronal para predecir las palabras subsiguientes en la cadena. En este caso, los vectores de representación de la palabra son aprendidos a partir de una máquina de aprendizaje. En general, el modelo se basa en las ideas del DSM; los vectores de palabras similares deben ser similares, y el contexto de las palabras juega un papel importante.

Se tiene, entonces, un vocabulario $\mathcal{C} = \{w_1, \dots, w_t\}$ que se espera sea grande, pero finito. Se trata de aprender un modelo $f(w_{i-n+1} \dots w_i) = P(w_i | w_{i-n+1} \dots w_i)$ tal que $\sum_{j=1}^t f_j(w_{i-n+1} \dots w_{i-1}) = 1$. Podemos definir entonces a f de la siguiente forma:

$$f_j(w_{i-n+1} \dots w_{i-1}) = g_j(v(w_{i-n+1}) \dots v(w_{i-1})) \quad (32)$$

donde

- $v : \mathcal{C} \rightarrow \mathbb{R}^m$ es una función que determina un vector en \mathbb{R}^m a cada palabra. En términos generales, $v(\cdot)$ se representa como una matriz de tamaño $|\mathcal{C}| \times m$.

- Una función g (una red neuronal) que mapea una secuencia de vectores de entrada, tomando en cuenta su contexto, a una distribución de probabilidad condicional para la siguiente palabra w_i . El vector de salida generado por g es un vector de probabilidad tal que en la i -ésima entrada estima la probabilidad $P(w_i|w_{i-n+1}...w_{i-1})$.

Para determinar la probabilidad de cada palabra dada las $n + 1$ palabras anteriores, se utiliza la regresión Softmax.

Definición 64 (Regresión Softmax). *La regresión Softmax es una forma de la probabilidad asimétrica de vecinos donde $\delta = \langle \cdot, \cdot \rangle$ es el producto interno, de tal forma que:*

$$P(w_i|w_{i-n+1}...w_{i-1}) = \frac{\exp(\langle v_i, h(v_{i-n+1}...v_{i-1}) \rangle)}{\sum_{k=1}^t \exp(\langle v_k, h(v_{i-n+1}...v_{i-1}) \rangle)} \quad (33)$$

donde $v_k = v(w_k) \forall k \in \{1, \dots, t\}$ y $h : \mathbb{R}^{m \times (n+1)} \rightarrow \mathbb{R}^m$ es una función que mapea los vectores de las palabras del contexto a un único vector en \mathbb{R}^m .

Por tanto, el modelo propuesto en (30) ahora depende de la ecuación (33). Sin embargo, lo que buscamos es una representación en espacio vectorial de las palabras. Por tanto, debemos buscar la representación vectorial que maximice (33). Presisamente en esto consiste el método de word embeddings.

14.3. Método de word embeddings

El método de word embeddings se basa en el SNE viendo a los elementos en el contextos de una palabra como vecinos. Por tanto se busca maximizar la ecuación (33) al tiempo que se minimiza una función de pérdida dada por la divergencia de Kullback-Liebler.

$$KL(p||q) = \sum_{i=1}^t p_i \log \frac{p_i}{q_i} \quad (34)$$

Se tiene entonces una palabra w y un contexto como en los modelos de DSM que puede estar determinado por una ventana o un contexto natural $N(w_i) = \{w_{i+n}...w_{i-n}\}$. Para generar los vectores de dimensión m , donde cada coordenada representa un rasgo que se aprende con una red neuronal, se siguen los siguientes pasos:

1. Se generan aleatoriamente una matriz $V \in \mathbb{R}^{t \times m}$ de vectores de entrada y otra matriz diferente $V' \in \mathbb{R}^{m \times t}$, donde t es el número de palabras.
2. Dado el contexto, se determina $P(w_i, N(w_i))$ con la regresión softmax, eligiendo la función h como:

$$h := \frac{1}{n} \sum_{k=1}^n v'_k \quad (35)$$

donde $v' = v'(w)$ es el vector correspondiente a la palabra w en la matriz $(V')^t$.

3. Para pasar de una capa oculta a la salida de la red neuronal, se usa una gradiente estocástica con un rango de aprendizaje η , tal que:

$$v'_{i+1}(w_j) = v'_i(w_j) - \eta \nabla \epsilon(w_j) \quad (36)$$

donde ϵ es la función de pérdida que se puede dar de las siguientes formas:

Divergencia KL. En este caso, el error toma la forma de la divergencia KL:

$$\nabla \epsilon(w_j) = \sum_{j=1}^t P(w_j|N(w_j)) \log \frac{P(w_j|N(w_j))}{Q(w_j|N(w_j))} h(N(w_j))$$

donde Q es la distribución en una iteración anterior.

Muestreo negativo. En este caso, se tiene lo siguiente:

$$\nabla \epsilon(w_j) = \begin{cases} 1 - P(w_j|N(w_j))h(N(w_j)) & \text{si } w_j = w_o \\ 0 - P(w_j|N(w_j))h(N(w_j)) & \text{si } w_j \neq w_o \end{cases} \quad (37)$$

donde w_o es la palabra objetivo de la iteración actual.

4. Para pasar de la entrada a las capas ocultas, siendo $w_k \in N(w_j)$ tiene que: $\forall m' \in 1, \dots, m$

$$v_{i+1}(w_k) = v_i(w_k) - \eta \sum_{j=1}^n \epsilon(w_j) \cdot v'_{m',j} \quad (38)$$

donde $v'_{m',j}$ representa la m' -ésima entrada de la matriz V' .

El algoritmo, entonces se correrá iterativamente hasta que la función de costo sea menor a un cierto rango dado. O bien, se puede correr un número determinado de iteraciones.

De esta forma, lo que el algoritmo hace es tratar de aproximar las distribuciones entre las matrices V y $(V')^t$ a partir de la observación de los contextos en que una palabra ocurre. La hipótesis distribucional sigue presente y por lo tanto a este tipo de aproximaciones se le ha llamado modelo de espacio distribucional continuo o CDSM.

En la actualidad los CDSM's han superado a los modelos anteriores en la mayoría de las tareas del procesamiento del lenguaje. Además, presentan la ventaja de que no requieren de una reducción de dimensionalidad, pues la dimensión se elige a priori y, por tanto, se puede trabajar con vectores que no tengan una alta dimensionalidad, a diferencia de los otros métodos del DSM que hemos visto.

15. Reducción de dimensionalidad

Dado que algunos de los modelos de representación en espacios vectoriales tienen una alta dimensionalidad, para trabajar con ellos es necesario reducir la dimensión de los vectores. Aquí abordaremos la descomposición en valores singulares (SVD) y la relajación espectral.

15.1. Descomposición en valores singulares

Para reducir la dimensión de un espacio vectorial, el SVD o descomposición en valores singulares se presentan varios resultados.

15.1.1. Diagonalización de matrices

Teorema 1. Si A es una matriz del operador T , existe una base ortogonal de sus vectores propios, tal que el operador T es diagonalizable en una matriz D , y se tiene que:

$$D = Q^{-1}AQ$$

donde Q es una matriz formada por los vectores propios de A .

Siendo de esta forma, tenemos entonces que:

$$A = QDQ^{-1} \tag{39}$$

Lo que nos dice la descomposición en valores singulares es que si tomamos Q como una matriz cuyas columnas son ortogonales entre sí, entonces:

$$A = QDQ^t \tag{40}$$

donde Q se puede ver como un operador simétrico (hermitiano) de rotación.

Proposición 6. Para Q se tiene que:

$$QQ^t = I = Q^tQ$$

El que pase esto, se debe precisamente a la ortogonalidad de Q . Es común, utilizar los vectores propios de A para formar las columnas de Q , por lo que D será una matriz diagonal compuesta de los valores propios de A .

15.1.2. Descomposición en valores singulares

A partir de lo anterior podemos determinar una descomposición para matrices cuadradas. Sin embargo, no todas las matrices son cuadradas. Para esto, se formula la descomposición en valores singulares.

Definición 65 (Adjunto de un operador). *Sea T un operador, su adjunto es el único operador T^* tal que:*

$$\langle Tu, v \rangle = \langle u, Tv \rangle, \forall u, v \in \mathbb{R}^n$$

En este caso, tomamos sólo el espacio \mathbb{R}^n pues es el que nos interesa. A partir de esta definición podemos definir el concepto de **valores singulares**.

Definición 66 (Valores singulares). *Un valor singular s es un valor propio del operador $(TT^*)^{\frac{1}{2}}$*

Es decir, dada la matriz A (que es la forma matricial del operador T) sus valores singulares son los valores propios de $(AA^t)^{\frac{1}{2}}$. Queda claro que esta nueva matriz es una matriz cuadrada.

El siguiente resultado nos da una idea de como determinar una descomposición en valores singulares.

Teorema 2. *Sea T un operador con valores singulares s_1, \dots, s_n . Entonces, existen bases ortonormales g_1, \dots, g_n y f_1, \dots, f_n tales que:*

$$T(v) = s_1 \langle v, g_1 \rangle f_1 + \dots + s_n \langle v, g_n \rangle f_n, \forall v \in \mathbb{R}^n$$

Este teorema puede comprobarse a partir del Teorema espectral y de la descomposición polar. Queda claro que si A es la matriz que representa al operador T , y si se tiene la base estándar e_1, \dots, e_n en \mathbb{R}^n entonces se tiene:

$$A = (\langle s_1 e_1, g_1 \rangle f_1 \dots \langle s_n e_n, g_n \rangle f_n) \quad (41)$$

$$= (\langle g_1, s_1 e_1 \rangle f_1 \dots \langle g_n, s_n e_n \rangle f_n) \quad (42)$$

$$= GSF \quad (43)$$

Donde G es la matriz formada por los vectores de la base g_1, \dots, g_n y F es la matriz formada por los vectores de la base f_1, \dots, f_n . Por último S es la matriz diagonal que contiene los valores singulares.

El algoritmo para reducir la dimensionalidad a partir de la descomposición en valores singulares consiste en los siguientes pasos:

1. Dada una matriz A de tamaño $m \times n$ obtener los valores singulares a partir de los valores propios de $(AA^t)^{\frac{1}{2}}$. A partir de los valores singulares formar una matriz diagonal donde la diagonal sea cada uno de los valores singulares.
2. Obtener las bases ortonormales G y F , principalmente de G y formar las matrices correspondientes. G se obtiene a partir de los vectores propios de $(AA^t)^{\frac{1}{2}}$ y F a partir de los vectores propios de $(A^t A)^{\frac{1}{2}}$. Si estos vectores no son ortonormales, se aplicará un proceso de Gram-Schmidt.
3. Determinar una dimensión k tal que las matrices G y S se reduzcan tal que $G_{n \times k}$ sea una matriz de $n \times k$ y $S_{k \times k}$ una matriz de $k \times k$.
4. Cada nuevo vector $v' \in \mathbb{R}^k$ será representado de la siguiente forma:

$$v' = S^{-1} \cdot G^t \cdot v \quad (44)$$

donde $v \in \mathbb{R}^n$ es el vector original.

15.2. Relajación espectral

El método de relajación espectral se basa en la teoría de grafos y en la idea de construir un grafo a partir de las matrices generadas por los vectores columnas que representan las palabras.

La relajación espectral contruye un grafo a partir de los vectores generados y toma los eigenvectores con los más pequeños valores propios de la matriz Laplaciana para determinar un nuevo espacio de menor dimensión. Las ventajas de la relajación espectral consisten en los resultados teóricos. Los cuáles presentamos a continuación.

Proposición 7. Una matriz Laplaciana, L , satisface las siguientes propiedades:

1. Para cada vector $x \in \mathbb{R}^n$ se tiene:

$$x^t L x = \frac{1}{2} \sum_{i,j=1}^n a_{ij}(x_i - x_j)^2$$

2. L es simétrica y positiva semi-definida.
3. L satisface $L \cdot \mathbb{1}_n = 0$
4. Los valores propios de L satisfacen: $\lambda_i \in \mathbb{R}$ of L , $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

Lema 1. Sea $G = (V, E)$ un grafo y $A = a_{ij}$ su matriz Laplaciana . Entonces el segundo valor propio más pequeño, λ_2 , satisface:

$$\lambda_2 \leq \frac{2|E|}{n-1} \quad (45)$$

Demostración. Se tiene que:

$$\lambda_2 = \min_x x^t A x \quad (46)$$

Sea ahora $\hat{A} = A - \lambda_2(I - n^{-1}\mathbb{1}_{n \times n})$. Sea $y \in E_n$ tal que $y = \alpha_1\mathbb{1} + \alpha_2x$ donde $x \in \mathbb{R}^n$. Tenemos de (46) que $\hat{A}\mathbb{1} = 0$, y por tanto:

$$y^t \hat{A}y = \alpha_2^2 x^t \hat{A}x = \alpha_2^2(x^t Ax - \lambda_2) \geq 0$$

Lo que implica que \hat{A} es positiva y por tanto de la ecuación (46) se tiene:

$$0 \leq \min_i a_{ii} - \lambda_2(1 - n^{-1})$$

Y de aquí se sigue que:

$$\lambda_2 \leq \frac{n}{n-1} \min_i a_{ii}$$

Ahora, ahora bien, y sabemos que $n \min_i v_i \leq \sum_i v_i = 2|E|$. De donde obtenemos:

$$\lambda_2 \leq \frac{n}{n-1} \min_i a_{ii} \leq \frac{2|E|}{n-1}$$

□

Lo que el lema anterior nos dice es que el segundo valor propio más pequeño de L está relacionado con la bi-partición del grafo asociado a L . De esta forma, podemos definir el algoritmo de la relajación espectral.

1. Dada la matriz W de vectores de palabra se obtiene $\mathfrak{A} = WW^t$ que representa una matriz de adyacencia. A partir de \mathfrak{A} se obtiene una matriz diagonal $\mathfrak{D} = \text{diag}(\{\sum_j^n A_{ij}\}_{i=1}^n)$ y se determina la matriz Laplaciana como:

$$L = \mathfrak{D} - \mathfrak{A} \quad (47)$$

2. Se obtienen los valores, $\lambda_1, \lambda_2, \dots, \lambda_n$, propios de L y sus vectores propios asociados v_1, v_2, \dots, v_n .
3. Se determina un k apropiado, que representa la nueva dimensión y se eligen los k valores propios más pequeños.
4. Dados los k valores propios más pequeños, se forma una matriz W' de tamaño $n \times k$ a partir de la transposición de los k vectores propios asociados a los k valores propios más pequeños.
5. Cada renglón en la matriz W' representa un vector de una palabra w tal que $w \in \mathbb{R}^n$.

16. Clasificación y agrupamiento de textos

En esta sección, abarcaremos diferentes algoritmos que tienen que ver con la clasificación y agrupamiento de textos. En general el esquema que se seguirá aquí será el **modelo general de aprendizaje a partir de ejemplos**. Que puede verse representado por la figura 4.

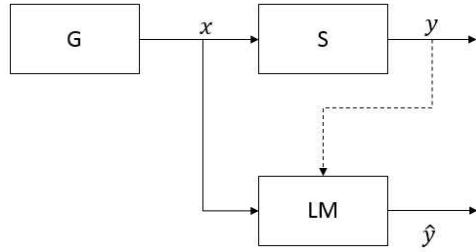


Figura 4: Modelo general de aprendizaje a partir de ejemplos.

Aquí, G es el **generador**: éste se encarga de producir un ejemplo x a partir de una distribución de los datos. S es el **supervisor** que determina la clase y a la que x pertenece. Los ejemplos generados por G se llevan a una **máquina de aprendizaje**, ML cuyo objetivo es determinar el \hat{y} tal que $\|y - \hat{y}\| \leq \epsilon$ para un ϵ pequeño.

En este modelo, podemos ver que ML no siempre aprende a partir de las y generadas por S . Por tanto, tendremos dos conjuntos de aprendizaje: supervisado y no supervisado. A continuación pasaremos a revisar cada uno de éstos.

17. Aprendizaje supervisado

El aprendizaje supervisado es aquel en que nuestra máquina de aprendizaje cuenta con un conjunto dado por el supervisor; es decir, un conjunto supervisado.

Definición 67 (Conjunto supervisado). *Un conjunto supervisado es un conjunto de la forma:*

$$\mathcal{S} = \{(x_i, y_i) : x_i \in G, y_i \in S, i = 1, \dots, n\}$$

En este caso le damos a nuestra máquina de aprendizaje ejemplos que cuentan con un vector x y una supervisión y . Si pensamos en los modelos ocultos de Markov, tendremos un modelo de aprendizaje supervisado. Aquí, se tiene un conjunto $\{(w_i, t_i)\}_{i=1}^l$, donde $w_i \in G$ son las palabras ejemplo y $t_i \in S$ es la etiqueta o clase que le corresponde a cada palabra. En este caso, podemos usar los HMM para clasificar otros elementos, diferentes de las etiquetas POS, que tengan un comportamiento estocástico.

17.1. Clasificación de idioma

Una de las tareas comunes y que puede solucionarse de manera simple es la clasificación del idioma de un texto. Debe notarse que en una lengua determinada se permiten determinadas combinaciones de letras. Por ejemplo, en español es raro tener la cadena 'th' mientras que para el inglés es común. También es común que en español aparezcan letras acentuadas o caracteres como 'ñ', lo que en inglés u otras lenguas no es común. Por ejemplo, en italiano se tiene 'gn' para representar este sonido y en portugués 'nh'.

Es común, para obtener mejores resultados, caracterizar un lenguaje a partir de sus trigramas de letra; es decir, se toman grupos de 3 letras. Estos trigramas son los ejemplos y la supervisión consiste en las etiquetas del lenguaje correspondiente. De esta forma, el conjunto de entrenamiento luce de la siguiente manera:

$$\mathcal{S} = \{(w_{1,3}, L) : w_{1,3} \text{ es un trígrama, } L \in \mathcal{L}\}$$

donde \mathcal{L} es el conjunto de etiquetas de lenguajes con los que se trabajará, por ejemplo 'español', 'inglés', 'italiano', 'francés', etc.

Sea entonces $w_{1,3} \in \mathcal{S}$ un elemento de entrenamiento al que le corresponde un $L \in \mathcal{L}$. Entonces, existe una dependencia estocástica de tal forma que podemos definir una distribución $P(L|w_{1,3})$ definida sobre \mathcal{L} . Precisamente, buscamos la forma de determinar esta distribución. Lo que tenemos es precisamente el conjunto \mathcal{S} , por lo que podemos determinar $P(w_{1,3}|L)$. Por tanto, para realizar la clasificación de idiomas bien podemos utilizar el algoritmo de **Bayes ingenuo**.

El algoritmo de Bayes ingenuo se basa fuertemente en el teorema de Bayes que asume que la probabilidad de un evento A dado B puede determinarse por el conocimiento de la probabilidad de B dado A , la probabilidad a priori de A y la probabilidad de la evidencia; es decir B . El teorema de Bayes asevera que:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (48)$$

En este caso, usamos la definición clásica para determinar la probabilidad a priori de un idioma:

$$P(L) = \frac{C(L) + \lambda}{|\mathcal{S}| + \lambda N}$$

Mientras que para obtener la probabilidad de un tri-grama dado el lenguaje tendremos que determinar:

$$P(w_{1,3}|L_i) = \frac{C((w_{1,3}, L_i) + \lambda)}{|\mathcal{L}| + \lambda N}$$

Cabe hacer notar que en las dos ecuaciones anteriores ya estamos aplicando un smoothing de Lindstone. En general, se puede ver que estamos realizando una especie de matriz de transición. Sin embargo, en este caso se nota que no es cuadrada. Sin embargo, se debe cumplir la condición de $\sum_{i=1}^n P(w_{1,3}|L_i) = 1$.

Por tanto, lo que el algoritmo de Bayes busca es maximizar $P(L|w_{1,3})$. Debe tomarse en cuenta que en la ecuación (48) la probabilidad de la evidencia $P(w_{1,3})$ puede omitirse, pues no tiene peso en la comparación; podemos entonces que $P(L|w_{1,3}, \dots, w_{n,3}) \propto P(L) \prod_{i=1}^n P(w_{i,3}|L)$. Se tiene entonces que buscamos:

$$\hat{L} = \arg \max_j P(L_j) \prod_{i=1}^n P(w_{i,3}|L_j) \quad (49)$$

En particular, el algoritmo se puede definir de la siguiente forma:

1. Dado un corpus etiquetado, obtener los tri-gramas de carácter correspondientes, $w_{1,3}$, y asociarlos a la etiqueta de lenguaje L .
2. Dado el conjunto $\{(w_{i,3}, L_i)\}_{i=1}^l$ obtener el modelo de entrenamiento compuesto los conjuntos $\{P(w_{i,3}|L_i)\}_{i=1}^l$ y $\{P(L_i)\}_{i=1}^k$.
3. Dado un elemento de evaluación obtener sus trigramas y obtener \hat{L} evaluando para cada lenguaje.

17.2. Diagnóstico de enfermedades

Supóngase ahora otro problema: se quieren diagnosticar enfermedades a partir de textos médicos para determinar la probabilidad de un posible diagnóstico. En los textos médicos contamos con la descripción de los síntomas del paciente. Para crear un modelo de aprendizaje, es entonces necesario tener un corpus de entrenamiento donde a una serie de síntomas se les asocie una enfermedad.

Supongamos que tenemos un corpus compuesto por textos médicos. Lo primero que debemos hacer es extraer los síntomas de estos textos. Para esto, se proponen algunos métodos: 1) Buscar patrones dentro del texto que indiquen dónde se encuentran los candidatos a síntomas, y extraer estos candidatos a partir de determinadas expresiones regulares; 2) utilizar un algoritmo de extracción de palabras clave (que veremos más adelante); 3) generar una ontología de los síntomas y las enfermedades, para obtener los datos necesarios a partir de ésta.

Supóngase entonces que se tienen los siguientes conjuntos:

- $\mathcal{E} = \{E_1, \dots, E_k\}$ un conjunto de nombres de enfermedades.
- $\mathcal{X} = \{x_1, \dots, x_n\}$ un conjunto de síntomas.
- $\mathcal{S} = \{(x_i, E_i) : x_i \in \mathcal{X}, E_i \in \mathcal{E}\}$ el conjunto de entrenamiento.

Por tanto, nuestro modelo de aprendizaje tiene que ser generado a partir de \mathcal{S} , el cual es un conjunto supervisado. En este caso, podemos utilizar el algoritmo de **árboles de decisión**.

Un árbol de decisión es un método simple de aprendizaje de máquina cuyo objetivo es la creación de un árbol. Es un modelo jerárquico que partitiona recursivamente el espacio de características. La mayor ventaja de este algoritmo es que es altamente interpretable.

Definición 68 (*Árbol*). *Un árbol es un grafo no-dirigido G que satisface:*

1. *G es conexo*
2. *G es acíclico*
3. *Si $e_i, e_j \in G$, $i \neq j$ son nodos, entonces $\exists! v_k \in G$ camino que conecta e_i y e_j .*

Cabe señalar que en este caso nuestro árbol debe contar con una raíz. En el caso del árbol de decisión, se deben determinar dos tipos de nodos:

1. Nodo interno. Es un nodo que está conectado por más de un camino. Estos nodos son los nodos de decisión.
2. Nodo terminal. Es un nodo que está conectado por un sólo camino y que cuenta con una etiqueta de salida. Representa las hojas del árbol (finales).

En general, el algoritmo de árboles de decisión consiste en los siguientes pasos:

1. A partir del conjunto de entrenamiento \mathcal{S} se busca crear un árbol que sea el más pequeño posible. Siendo $G = (E, V)$ donde E es el conjunto de nodos y V de caminos, se toma $e_1 = \mathcal{S}$.
2. Se busca particionar el nodo $e_1 = \mathcal{S}$ de tal forma que se obtenga una división homogénea de los datos. Esto se logra a partir de minimizar una función de costón I . Para esto, nos podemos basar en diferentes funciones, de las que mencionamos dos.

Índice Gini.

$$G(X) = \frac{\sum_{i,j} |x_i - x_j|}{2n^2} \quad (50)$$

Entropía.

$$H(X) = - \sum_i P(x_i) \log_2 P(x_i) \quad (51)$$

3. Para determinar el elemento que representará el nodo de una clase se calcula:

$$\arg \min_i I(E_j | x_i)$$

que en el caso de la entropía se puede calcular como la información mútua.

4. Repetimos el paso anterior para construir el árbol óptimo. Cuando ya no necesitamos hacer un corte quiere decir que llegamos a la hoja y esta debe etiquetarse. El algoritmo termina cuando ya no hagamos más cortes.

Lo que queremos es que el corte realizado por el algoritmo sea puro, pues a partir de cortes puros, necesitaremos menos cortes para poder llegar a las hojas del árbol.

17.3. Perfilación de autor

Supóngase que se tiene una serie de textos de diferentes autores. A partir de estos, se puede obtener información sobre el autor de dicho texto. Estos datos pueden ser la edad, sexo, nivel académico, etc. Para tal tarea, se debe tener de antemano las necesidades que tenemos, qué es lo que queremos averiguar sobre el autor de un texto y para qué fines lo queremos.

Supongamos que queremos clasificar textos en dos grupos: si el autor es mujer o si el autor es hombre. Para esto, necesitamos un conjunto de etiquetas $T = \{h, m\}$ donde h corresponde a hombre y m a mujer. Entonces, nuestro conjunto de entrenamiento es:

$$\mathcal{S} = \{(x_i, t_i) : x_i \in \mathbb{R}^n, t_i \in T\}$$

Dado que nuestras categorías son binarias, podemos ver el problema como el determinar si un punto pertenece o no a la clase 1. Por tanto, para esta tarea podemos aplicar un algoritmo de Support Vector Machines o SVM. Para esto, necesitamos introducir varios conceptos.

17.3.1. Vectores de soporte

El algoritmo de vectores de soporte es un tipo de aprendizaje que se basa en una clasificación por hiperplanos. El algoritmo de vectores de soporte está pensado para un conjunto cuya partición sea binaria. Es decir, dado los datos de $\mathcal{X} = \{x_1, \dots, x_n\}$ y las etiquetas $hombre = -1$ y $mujer = 1$, el conjunto de entrenamiento es:

$$\mathcal{S} = \{(x_i, t_i) : x_i \in \mathbb{R}^n, t_i \in \{-1, 1\}\}$$

De tal forma que podamos definir un funcional $F : \mathcal{X} \rightarrow \{-1, 1\}$ tal que:

$$F(x) = \begin{cases} +1 & \text{si } x \text{ mujer} \\ -1 & \text{si } x \text{ hombre} \end{cases} \quad (52)$$

Entonces, se tiene que encontrar un vector $w \in \mathbb{R}^n$ tal que:

$$\langle w, v \rangle + b = 0 \quad (53)$$

donde v es un vector del hiperplano de separación. En otras palabras, el vector w es un vector perpendicular a v , mientras que $v \in H$ es un vector del hiperplano H que separa a los puntos con la etiqueta -1 y $+1$.

Para determinar este vector w que determinará el hiperplano se utiliza un proceso de optimización. Podemos verlo como el siguiente proceso:

1. $w = (0 \ 0 \ 0)$
2. $\forall (x, t) \in S:$
 $\hat{y} = \langle x, w \rangle + b > 0$
 $\epsilon = t - \hat{y}$

3. Si $\epsilon \neq 0$:

$$\forall x_i \in x, x_i \in \mathbb{R}:$$

$$w_i = w_i + \eta \cdot \epsilon \cdot x_i$$

Esta es una forma sencilla de definir un hiperplano de tal forma que los datos estén separados. A esta simple forma de generar una clasificación se le conoce como **perceptrón**.

Definición 69 (Perceptrón). *Un perceptrón es un algoritmo de aprendizaje supervisado caracterizado por el funcional $F : \mathbb{R}^n \rightarrow \mathbb{R}$ tal que:*

$$F(x) = \begin{cases} 1 & \text{si } \langle w, x \rangle + b > 0 \\ 0 & \text{si } \langle w, x \rangle + b \leq 0 \end{cases} \quad (54)$$

donde $w \in \mathbb{R}^n$ es el vector que define al funcional y b es el sesgo.

Sin embargo, el perceptrón formula una separación débil de los datos, pues no optimiza la separación de los datos por un hiperplano y por tanto no minimiza el error por sobreajuste. En realidad, pueden existir diferentes funcionales que determinen diferentes hiperplanos para la separación de un mismo conjunto de datos. Por ejemplo, en la Figura 5 se ven varios hiperplanos que pueden separar los puntos etiquetados con 0 de los puntos etiquetados con 1.

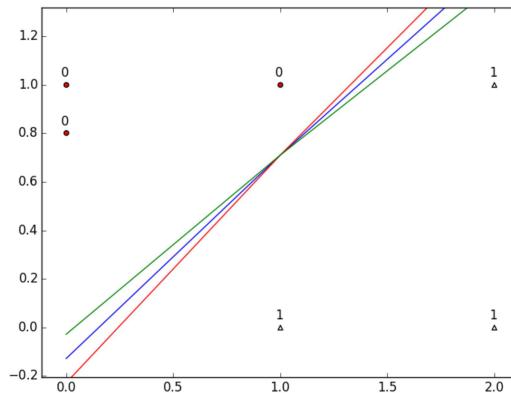


Figura 5: Separación de datos a partir de diferentes hiperplanos.

La idea de las máquinas de vectores de soporte es determinar un único hiperplano que minimice el error por sobreajuste. Para esto, se tienen que ver los vectores que estén más cercanos al hiperplano y crear un margen a partir de estos vectores. Es decir, aquellos vectores que cumplen $\min_i \{||x - x'_i||\}$ con x'_i con una etiqueta diferente a x . O en otras palabras

Definición 70 (Vector soporte). *Un vector soporte, x , es aquel que cumple la igualdad:*

$$|\langle w, x \rangle + b| = 1 \quad (55)$$

Es decir, los vectores soporte son los que se encuentran en los márgenes que delimitan al hiperplano. El hiperplano, por otra parte se encuentra a una distancia del origen igual a $\frac{|b|}{\|w\|}$.

Proposición 8. *La distancia del hiperplano al origen es:*

$$\frac{|b|}{\|w\|} \quad (56)$$

Demostración. Tenemos que b es una constante que determina una transalución de la función. El hiperplano está formado por $\{x | \langle w, x \rangle = 0\}$. Por tanto tenemos que la distancia está dada por:

$$\begin{aligned} \|x - 0\| &= (\langle w, x \rangle - |b|) - \langle w, 0 \rangle \\ \implies \langle w, x \rangle &= |b| \\ \implies \langle \frac{w}{\|w\|}, x \rangle &= \frac{|b|}{\|w\|} \end{aligned}$$

Corolario 1. *La distancia entre los hiperplanos formados por los vectores de soporte es:*

$$\frac{2}{\|w\|} \quad (57)$$

Entonces, el algoritmo de SVM busca determinar el mejor hiperplano de separación de datos en base a los vectores de soporte, de tal forma que el mejor hiperplano será el que se encuentre en medio de los hiperplanos generados por los vectores de soporte. Esta idea se muestra gráficamente en la figura 6.

Por tanto, el problema de encontrar el mejor hiperplano está restringido a la condición:

$$|\langle w, x \rangle + b| \geq 1 \quad (58)$$

Y finalmente, queremos que el margen entre los vectores de soporte y el hiperplano sea mínimo. De aquí obtenemos que el problema de optimización para SVM es encontrar:

$$\min \|w\| : \forall x \in \mathcal{X} |\langle w, x \rangle + b| \geq 1 \quad (59)$$

De tal forma que obtengamos la función de decisión dada por:

$$F(x) = \text{sgn}(\langle w, x \rangle + b) \quad (60)$$

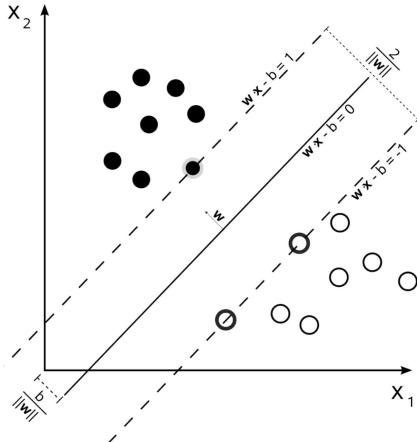


Figura 6: Hiperplano de decisión dado por los vectores d esoporte

17.3.2. Funciones kernel

Definición 71 (Función kernel). *Sea \mathcal{X} un espacio vectorial y sea \mathcal{H} un espacio isomórfico a \mathcal{X} , llamado el espacio de rasgos. Una función $\phi : \mathcal{X} \rightarrow \mathcal{H}$ dada por $x \mapsto x := \phi(x)$ es una función núcleo, k , con respecto al producto interno, $\langle \cdot, \cdot \rangle$ en \mathcal{H} si:*

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \quad (61)$$

Entonces, lo que debemos buscar son funciones kernel que creen un espacio de rasgos que represente bien las características que queremos resaltar. Supóngase que un vector $x \in \mathcal{X}$ compuesto por las entradas $x = (x_1, \dots, x_n)$. Entonces podemos definir:

Definición 72 (Monomiales). *Supóngase que el vector x la mayor cantidades de información está contenida en el producto de grado d , esto es:*

$$\phi_d(x) := (x_{i_1} \cdot \dots \cdot x_{i_d})_{i=1}^n$$

Estos productos son llamados los rasgos monomiales de las entradas de x de grado d .

Proposición 9 (Kernel polinomial). *Defínase $\phi_d : \mathbb{R}^n \rightarrow \mathbb{R}^n$ como la función $x \mapsto \phi_d(x)$, entonces la función kernel sobre ϕ_d , llamada kernel polinomial, Está dada por:*

$$k(x, x') = \langle x, x' \rangle^d \quad (62)$$

Demostración. Se tiene que:

$$\begin{aligned}
k(x, x') &= \langle \phi_d(x), \phi_d(x') \rangle \\
&= \sum_{i_1=1}^n \dots \sum_{i_d=1}^n x_{i_1} \cdot \dots \cdot x_{i_d} \cdot x'_{i_1} \cdot \dots \cdot x'_{i_d} \\
&= \sum_{i_1=1}^n x_{i_1} \cdot x'_{i_1} \dots \sum_{i_d=1}^n x_{i_d} \cdot x'_{i_d} \\
&= \left(\sum_{i=1}^n x_i \cdot x'_i \right)^d \\
&= \langle x, x' \rangle^d
\end{aligned}$$

□

Esta proposición determina una serie de funciones kernel: por ejemplo, si $d = 1$ la función kerne es un producto interno. Tenemos que tomar otro punto a consideración; esto es que podamos definir funciones kernel con la propiedad de ser definidos positivos. Para esto, antes introduciremos resultados previos.

Definición 73 (Matriz de Gram). *Dada una función $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ tal que x_1, \dots, x_m , entonces la matriz $K \in \mathbb{R}^{m \times m}$ cuyas entradas son de la forma:*

$$K_{ij} = k(x_i, x_j)$$

Es la matriz de Gram de k con respecto a \mathcal{X} .

Definición 74 (Matriz positiva-definida). *Sea $K \in \mathbb{R}^{m \times m}$ una matriz de Gram y $x \in \mathbb{R}^m$ un vector. Se dice que K es positiva-definida si:*

$$x^t K x \geq 0$$

Como trabajamos en el caso real, la condición positiva-definida implica que $\sum_{i,j} x_i x_j K_{ij} \geq 0$. Además, K es una matriz simétrica dado un producto interno. Entonces, dado una matriz simétrica positiva-definida tenemos que su eigenvalores son no-negativos. A partir de esto, entonces, podemos definir la condición positiva-definida para una función kernel.

Definición 75 (Kernel positivo-definido). *Sea $\mathcal{X} \neq \emptyset$ un conjunto y $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ una función kernel. Si se tiene la matriz de Gram $K = (k_{ij}) = k(x_i, x_j)$ positiva definida, entonces k es positivo definido.*

Debe notarse que ser una matriz positiva definida implica dos condiciones:

- $k(x_i, x_i) \geq 0$, es decir, positividad en la diagonal, y
- $k(x_i, x_j) = k(x_j, x_i)$; es decir, simetría.

18. Aprendizaje no supervisado

A diferencia del aprendizaje supervisado, el aprendizaje no supervisado no recibe ninguna señal del supervisor. De tal forma, la máquina tiene que generar una predicción a partir únicamente de los datos del generador. Se cuenta entonces con un conjunto no supervisado \mathcal{U} .

Definición 76 (Conjunto no supervisado). *Sea $G = \{x_1, \dots, x_n\}$ el conjunto de elementos lingüísticos producidos por el generador; entonces, un conjunto no supervisado es:*

$$\mathcal{U} = \{x_i : x_i \in G, i = 1, \dots, n\} \quad (63)$$

18.1. Agrupamiento de documentos

El agrupamiento de documentos es una tarea muy común en muchos sistemas de recuperación de información. En general, repasaremos varios métodos para esto. Antes que nada, debemos definir el concepto de agrupamiento:

Definición 77 (Agrupamiento). *Dado un conjunto de objetos X , un agrupamiento $\mathfrak{C} = \{C_i : C_i \subseteq X, i = 1, \dots, k\}$ es una particiónn de X tal que*

1. $\bigcup_{C_i \in \mathfrak{C}} C_i = V$
2. $\forall C_i, C_j \in \mathfrak{C} : C_i \cap C_j = \emptyset$ for $i \neq j$

A partir de esto podemos definir diferentes algoritmos de agrupamiento para obtener la partición deseada.

18.1.1. K-medias

Sea $\mathcal{X} = \{d_1, \dots, d_n\} \subseteq \mathbb{R}^n$ el conjunto de las representaciones vectoriales de los documentos de un corpus. Para particionar estos datos, antes que nada debemos determinar bajo qué criterio deseamos realizar la separación; ésta puede ser por área temática, por género textual, por autor, etc. Esta decisión debe estar codificada en la representación vectorial y por las medidas de similitud utilizadas.

La forma más inmediata de visualizar el problema de agrupamiento es bajo la idea de que cada vector es un punto en un espacio vectorial. Por tanto podemos definir una métrica sobre ellos. Una métrica debe cumplir las propiedades que se dan en la siguiente definición.

Definición 78 (Métrica). *Sea \mathcal{X} un espacio vectorial. Una función $\delta : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ es una métrica si:*

1. $\forall x, y \in \mathcal{X} (\delta(x, y) \geq 0)$
2. $\forall x \in \mathcal{X} (\delta(x, x) = 0)$
3. $\forall x, y \in \mathcal{X} (\delta(x, y) = \delta(y, x))$

$$4. \forall x, y, z \in \mathcal{X} (\delta(x, z) \leq \delta(x, y) + \delta(y, z))$$

A partir de una mátrica podemos determinar la distancia de un documento a otro en un espacio vectorial. El algoritmo de k-medias se basa en esta premisa para realizar agrupamientos sobre los documentos. A continuación describimos el algoritmo:

1. Se selecciona un k que es el número de grupos que se desean como grupos de los documentos.
2. Se generan k puntos en \mathbb{R}^n aleatorios. Estos son los llamados *centroïdes*, c_j .
3. Dados los puntos d_1, \dots, d_n se calcula $\delta(d_i, c_j), i = 1, \dots, n, j = 1, \dots, k$ y se asignan los puntos d_i al centroide más cercano, para crear los grupos \mathcal{C}_j .
4. Por cada grupo $\mathcal{C}_j = \{d_{j_1}, \dots, d_{j_r}\}, r = 1, \dots, n$ Se recalcula un centroide de la siguiente forma:

$$c'_j = \frac{1}{r}(d_{j_1} + \dots + d_{j_r})$$

5. Se repite el paso 3 bajo los nuevos centroides. Es decir, se clacula $\delta(d_i, c'_j)$ y se reasignan los puntos en k grupos nuevos.
6. El algoritmo termina hasta que no se hagan más asignaciones.

18.1.2. Algrupamiento jerárquico

Otro método para determinar grupos a partir de un conjunto de documentos es a partir del agrupamiento jerárquico. Éste se divide en dos tipos: aglomerativo y divisivo.

Aglomerativo. Asume que se tienen n grupos, donde n es igual al número de elementos. Entonces va encontrando similitudes hasta formar un sólo grupo.

Dvisivo. Al contrario del aglomerativo, este asume un sólo grupo y encuentra disimilitudes hasta separar los datos completamente.

En este caso, podemos revisar un algoritmo aglomerativo de *single-linkage*. Este es un algoritmo simple que crea un dendograma a partir de los documentos. Éste consiste en los siguientes pasos.

1. Se comienza con los n grupos tales que $\mathcal{C}_1 = \{d_1\}, \dots, \mathcal{C}_n = \{d_n\}$; es decir, cada documento forma un grupo.

2. En el siguiente paso buscamos la cercanía entre grupos a partir de:

$$D(\mathcal{C}_i, \mathcal{C}_j) := \min_{d^i, d^j} \delta(d^i, d^j)$$

de tal modo que $d^i \in \mathcal{C}_i$ y $d^j \in \mathcal{C}_j$. Los pares de grupos más cercanos van a emerger en un sólo grupo \mathcal{C}' .

3. Se repite el paso anterior hasta que se obtenga un sólo grupo.

Este algoritmo jerárquico nos da un dendograma, que es una estructura jerárquica de los datos. Para hacer los grupos, debemos seleccionar donde hacer el corte óptimo.

18.2. MajorClust

Los siguientes dos algoritmos que revisaremos ven el problema de agrupamiento de forma distinta. Ahora la perspectiva geométrica se reemplaza por una perspectiva de grafos.

El algoritmo de MajorClust busca particionar un grafo; esto es, torna la definición de agrupamiento en la siguiente.

Definición 79 (Agrupamiento sobre grafos). *Si $G = (V, E, \phi)$ es un grafo donde $V = \{v_1, \dots, v_m\}$ es el conjunto de nodos del grafo, $E = \{E_1, \dots, E_m\}$ es el conjunto de vértices y $\delta : V \times V \rightarrow \mathbb{R}^+$ es una función de peso, entonces, un agrupamiento sobre G es el conjunto $\mathcal{C} = \{G(C_i) : G(C_i) \subseteq G, i = 1, \dots, k\}$ donde $G(C_i)$ son particiones del grafo.*

Básicamente el algoritmo propone que dado un grafo G , tal que $G = \langle V, E, \phi \rangle$ (donde V denota los nodos del grafo, que en este caso son los documentos; E , las aristas entre los documentos delimitados por una función de similitud δ , que puede verse como la distancia que separa los nodos), el grafo G se descompone en una colección de grupos C , tal que $C = \{C_1, \dots, C_k\}$, donde C_i representan los grupos de documentos con alta similitud, entonces:

$$\Lambda(C) = \sum_{i=1}^k |C_i| \cdot \lambda_i \quad (64)$$

Donde λ_i es una función de máxima atracción que designa la conectividad entre las aristas de los grupos, $G(C_i)$. Esto se puede definir como el mínimo de aristas que se necesitan eliminar para desconectar el grafo $G(C_i)$, esto es:

$$\lambda_i = \min \sum_{(v,u) \in E'} \delta(v, u) \quad (65)$$

Donde $E' \subset E$; es decir, se busca minimizar la función de similitud (o bien la distancia) entre dos aristas pertenecientes a dos nodos o documentos u y v .

El algoritmo, consiste en los siguientes pasos:

1. Crear un grafo donde $E = \{d_1, \dots, d_n\}$ y las aristas conectando cada nodo estén pesadas por la métrica δ . Defínase $i = 0$.
2. En la iteración $i + 1$ defínase una función $c : V \rightarrow \mathbb{N}$ tal que $\forall v \in V (c(v) = i + 1)$, donde $i + 1$ es la etiqueta del grupo al que v pertenece, $G(C_{i+1})$.
3. Por cada $u \in V$ si $\sum_{v,u} \delta(v, u) \geq \sum_{w,u} \delta(w, u), \forall w \in V$, entonces defínase $c^*(v) := i + 1$.
4. Si $c^*(v) \neq c(v)$ reasígnase $c^*(v) := c(v)$ y vuélvanse a repetir los pasos anteriores. De lo contrario, el algoritmo termina.

18.3. Spectral clustering

Spectral clustering es una técnica que se basa grandemente en la relajación espectral vista más arriba. En realidad, no se trata de un algoritmo de agrupamiento como tal, pero tiene ciertas ventajas en una tarea de agrupamiento. En primer lugar, tenemos el resultado:

$$\lambda_2 \leq \frac{|E|}{n - 1}$$

Que nos dice que el segundo eigenvalor más pequeño está ampliamente relacionado con la bipartición del grafo. Además, tenemos el siguiente resultado, del que todavía no hemos hablado.

Proposición 10. *Si G es un grafo no-dirigido cuya matriz de adyacencia sea positiva definida, entonces la multiplicidad k del eigenvalor 0 de L es el número de componentes conectados.*

Demostración. Si $k = 1$ entonces el grafo está completamente conectado. Sea x un eigenvector con eigenvalor 0, de donde:

$$0 = x^t L x = \sum_{i,j} a_{ij} (x_i - x_j)^2$$

Y ya que a_{ij} es no negativo para todo i, j , entonces si v_i conectado con v_j vértices, implica que $x_i = x_j$ por lo que $x = \mathbf{1}$, y esto implica que es el único eigenvector con eigenvalor 0. Es decir, la multiplicidad de $\lambda_k = 0$ es uno.

Ahora asúmase que es cierta la afirmación para k componentes conectados. Entonces, asúmanse $k + 1$ componentes L_1, \dots, L_{k+1} . Sin pérdida de generalidad, ordénense los vértices de acuerdo al componente conectado al que pertenecen, entonces:

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_{k+1} \end{pmatrix}$$

Por hipótesis de inducción sabemos que la multiplicidad del eigenvalor 0 es igual a k . Por un argumento similar al que se ha hecho arriba, llegamos a probar que para L_{k+1} el eigenvalor 0 corresponde a un eigenvector $\mathbb{1}_{k+14}$. Por lo que la multiplicidad ahora es igual a $k + 1$. \square

Para crear el grafo, se pueden usar los tres métodos ya especificados:

1. Completely connected
2. ϵ connected
3. $k - nn$ connected

Sin embargo, para determinar los pesos y saber que la proposición 10 se cumple, necesitamos una función kernel positiva definida. Sabemos que K es la matriz de Gram con respecto a k . Definamos una función kernel como:

$$k(x_i, x_j) = \frac{1}{K_{ii} + K_{jj} + K_{ij} + K_{ji}} \quad (66)$$

Ya que K es simétrica y positiva semidefinida se puede descomponer tal que $K = QDQ^t$ donde Q es una matriz ortogonal formada por los eigenvectores de K y D es la matriz diagonal de eigenvalores. Entonces, $K_{ij} = (QD^{\frac{1}{2}})_i(QD^{\frac{1}{2}})_j^t$. Entonces, buscando $\delta(x_i, x_j) = k(x_i, x_j)^{-\frac{1}{2}}$ entonces:

$$k(x_i, x_j) = d(x_i, x_j)^2 \quad (67)$$

$$= K_{ii} + K_{jj} - K_{ij} - K_{ji} \quad (68)$$

$$= ((QD^{\frac{1}{2}})_i - (QD^{\frac{1}{2}})_j)^2 \quad (69)$$

Es decir, cada entrada de K es del tipo $\frac{1}{\delta^2}$.

Definición 80 (Kernel euclídeo inverso). *Sea σ un parámetro (generalmente la varianza de los datos) entonces el kernel euclídeo inverso está definido por:*

$$k(x, x') = \frac{\sigma^2}{||x - x'||^2} \quad (70)$$

Otra función kernel que podemos definir, comúnmente utilizada para estas tareas es el kernel gaussiano.

Definición 81 (Kernel gaussiano). *Sea σ un parámetro (generalmente la varianza de los datos) entonces el kernel gaussiano se define como:*

$$k(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2} ||x_i - x_j||^2\right) \quad (71)$$

A partir del cálculo de estos kernel, podemos calcular la matriz de adyacencia. En una sección anterior vimos que podemos definir un grafo completamente conectado. Un grafo ϵ **conectado** es aquel que para un ϵ dado, los vértices v_i y v_j están conectados si $k(v_i, v_j) < \epsilon$.

Para el grafo **k-nn conectado**, se hace una algoritmo de k -vecinos, y sólo estos k -vecinos más cercanos están conectados. El algoritmo de $k - nn$ para spectral clustering se describe a continuación.

1. Dados los vectores $w_1, \dots, w_n \in \mathbb{R}^n$, se elige $1 \leq k \in \mathbb{N}$ que representará el número de vecinos de un vector w_i .
2. Por cada w_i vector se calcula $\delta(w_i, w_j), j = 1, \dots, n - 1$ y se ordenan sus valores.
3. Se toman los k primeros w_j que serán los elementos conectados a w_i , lo demás son elementos desconectados.

De esta forma, el algoritmo de espectral clusterin se describe de la siguiente forma:

1. A partir de los datos d_1, \dots, d_n se crea una matriz de adyacencia que puede ser:
 - Completamente conectada
 - ϵ conectada
 - k-nn conectada
2. Se obtiene la matriz Laplaciana, L como se ha señalado más arriba.
3. De la matriz L se obtienen los eigenvectores y eigenvalores. Se crea una nueva matriz con los r eigenvectores transpuestos correspondientes a los r eigenvalores más pequeño.
4. A partir de esta nueva representación de los datos, se aplica un algoritmo de agrupamiento (usualmente k-medias).

A. Proceso de Gram Schmidt

Sea $V \in \mathbb{R}^{n \times n}$ una matriz conformada por una base no ortonormal de \mathbb{R}^n . Sea entonces cada vector renglón $w_1, \dots, w_n \in V$. Para obtener una base ortonormal de \mathbb{R}^n a partir de los vectores renglón de V se aplica un proceso de Gram-Schmidt.

Definición 69 (Proyección ortogonal). *Sean w_1 y w_2 vectores. La proyección ortonormal de w_2 sobre w_1 está definida como:*

$$\text{proj}(w_1, w_2) := \frac{\langle w_2, w_1 \rangle}{\|w_1\|^2} w_1 \quad (52)$$

A partir de la proyección, entonces, definiremos el proceso de ortogonalización por Gram-Schmidt de la siguiente forma:

$$v_j = w_j - \sum_{i=1}^{j-1} \text{proj}(v_i, w_j) \quad (53)$$

donde v_j es la representación ortogonal de w_j . Para obtener los vectores normales, e_j , a partir de los vectores ortogonales, se hará:

$$e_j := \frac{e_j}{\|e_j\|} \quad (54)$$

Ejemplo 10. Téngase la siguiente base de \mathbb{R}^3 : $\beta = \{(1, 0, 0), (1, 1, 0), (1, 1, 1)\}$ aplicando el proceso de Gram-Schmidt se obtiene:

$$\begin{aligned} v_1 &= (1, 0, 0) \\ v_2 &= w_2 - \text{proj}(v_1, w_2) \\ &= (1, 1, 0) - (1, 0, 0) \\ &= (0, 1, 0) \\ v_3 &= w_3 - \text{proj}(v_1, w_3) - \text{proj}(v_2, w_3) \\ &= (1, 1, 1) - (1, 0, 0) - (0, 1, 0) \\ &= (0, 0, 1) \end{aligned}$$

Estos vectores ya son normales por los que no es necesario dividirlos entre su norma. Finalmente, nuestra nueva base es $\beta^\perp = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$.