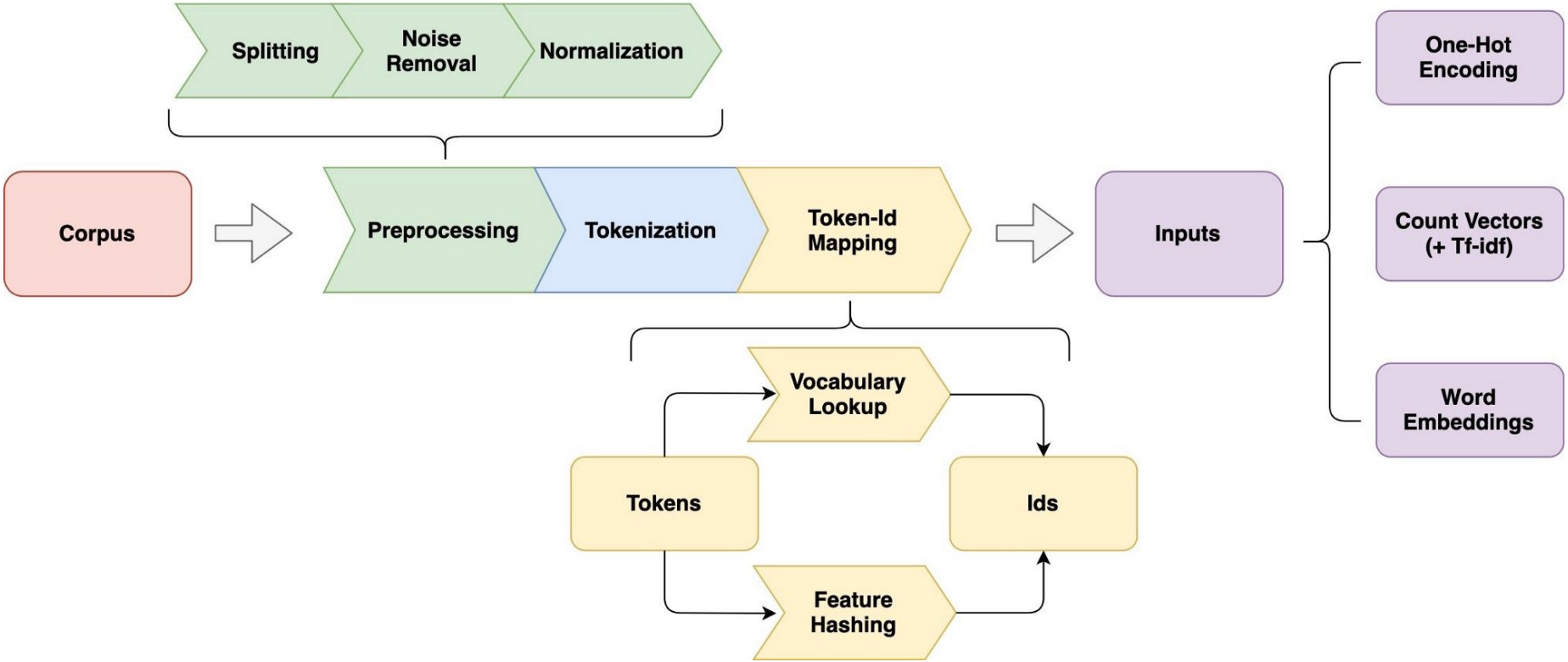
Pre-Procesamiento y Representación de texto

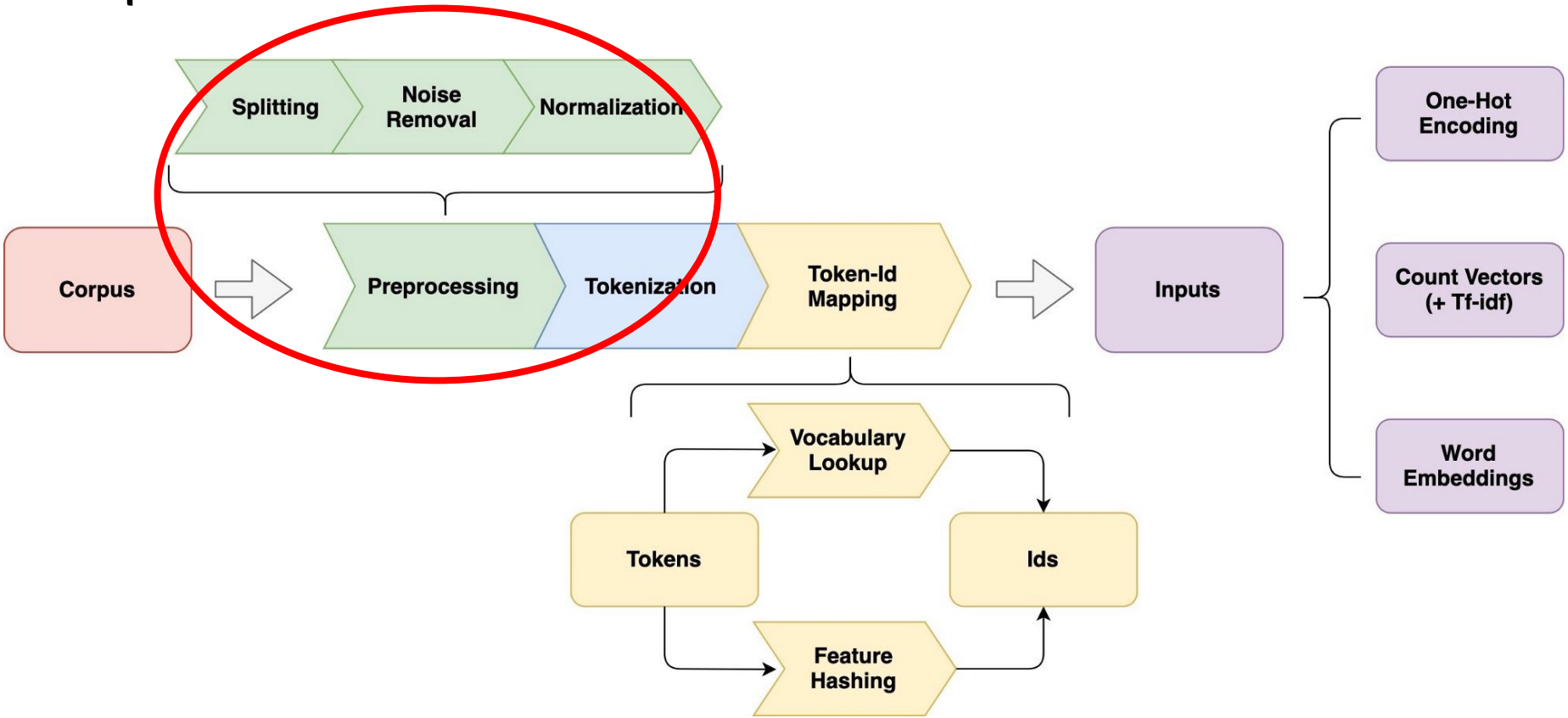
Andrés Rosso

Pipeline NLP



Pre-procesamiento

Preprocesamiento



Qué es ?

- El preprocesamiento de texto es la práctica de limpiar y preparar los datos de texto para obtener una mejor representación.
  - Más rica semánticamente
  - Menos ambigua
  - Que facilite su representación computacional.

- NLTK, Spacy y RE son librerías comunes de Python que se utilizan para manejar muchas tareas de preprocesamiento de texto.

Grandes Tareas

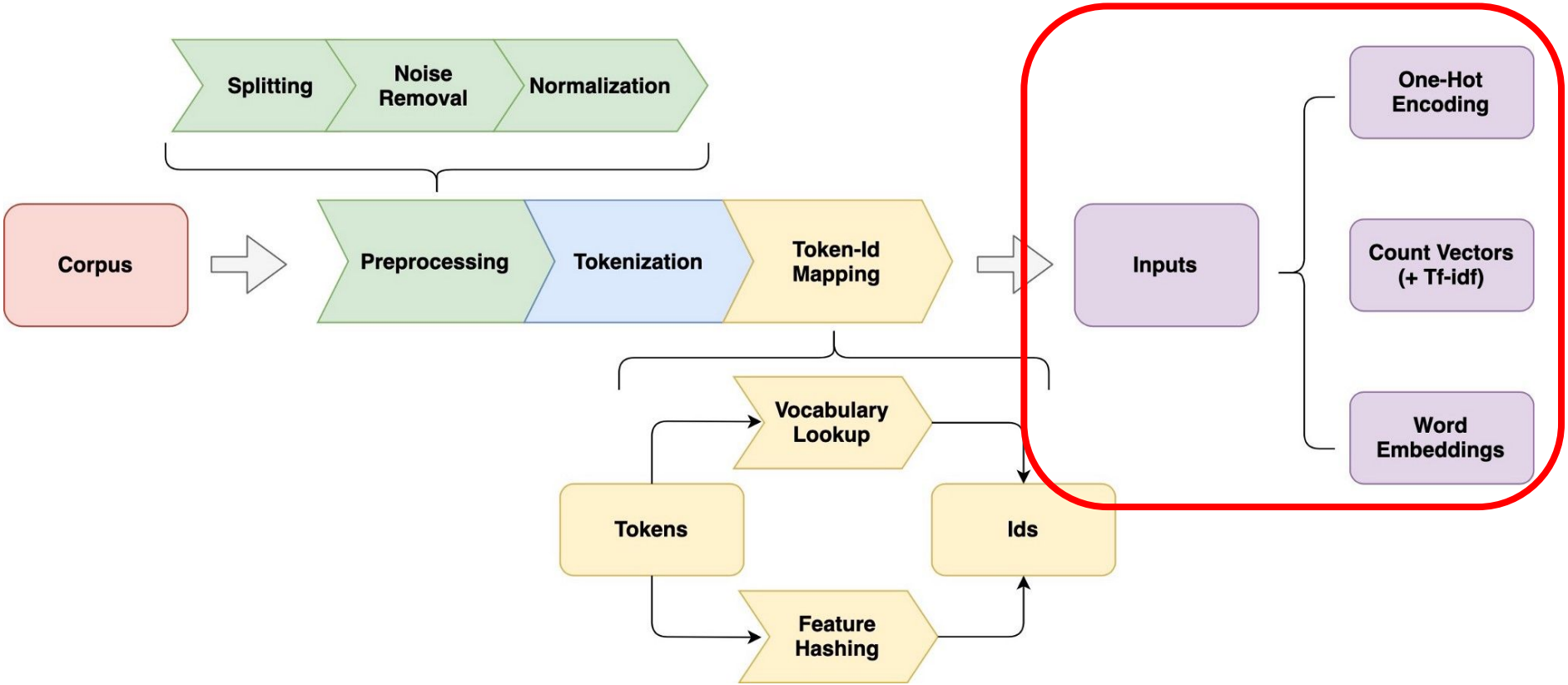
- **Eliminación de ruido**: eliminar el formato del texto.
- **Tokenización**: dividir el texto en componentes pequeños.
- **Normalización**: abarca varias sub-tareas como: lematización, cambio a minúsculas y eliminación de stop-words, etc.
- **Stemming**: proceso heurístico que corta los extremos de las palabras con la esperanza de lograr representarla como su raíz.
- **Lematización**: devolver la forma base o raíz común de una palabra.
- **Eliminación de palabras irrelevantes (stop-words)**: eliminar palabras de una cadena que no proporcionan ninguna información sobre el tono de una declaración.
- **PoS tagging:** asignar una parte del discurso a cada palabra en una cadena.

Subtareas del Preprocesamiento

- Conversión a minúsculas
- Eliminación de puntuaciones
- Eliminación de palabras frecuentes
- Eliminación de palabras raras
- Stemming
- Lematización
- Eliminación de emoticonos/emojis
- Conversión de emoticonos/emojis en palabras
- Eliminación de URLs/Emails/Números/HTML
- Corrección ortográfica

Representación del Texto

Representación



Representación a Nivel de Carácter

La representación a nivel de caracteres de un texto consiste en secuencias de caracteres longitud 1, 2, 3, n.

Por ejemplo:

- Los 1-gramas también se llaman unigramas.
- Los 2-gramas también se llaman bigramas o digramas.
- Los 3-gramas también se llaman trigramas.

N-Grams y Probabilidad

- Con N-gramas queremos calcular: P(w|h), donde w se refiere a una palabra y h a un conjunto de palabras previas o su historia

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

- Si queremos calcular la probabilidad de que: tina aparezca después de anita lava la, sería:

$$P(\text{tina} \mid \text{anita lava la}) = \frac{C(\text{anita lava la tina})}{C(\text{anita lava la})}$$

N-Gramas Simplificación

- El cálculo de la probabilidad completa es costoso. Supondremos una breve historia *<<suposición de Markov>>*.
- Un bigrama aproxima:
  - P(tina | anita lava la) ≈ P(tina | la)

Fortalezas y Debilidad N-Gramas

La representación tiene varios puntos fuertes importantes:

- Es muy robusta ya que evita la morfología del lenguaje
- Útil, por ejemplo, para la identificación de lenguas
- Capta patrones simples a nivel de caracteres
  - Detección de spam
- Para tareas semánticas más profundas, la representación es demasiado débil

Representación a Nivel de Palabra

- La palabra es una unidad bien definida en las lenguas occidentales - por ejemplo, el chino tiene una noción diferente de unidad semántica.
- La representación más común de texto más utilizada.
- Existen muchos paquetes de software que ayudan a obtener la representación.

Propiedades de las Palabras

Pueden inducir ruido por:

- **Homonimia**: misma forma, pero diferente significado
  - **Vino**
    - Forma del verbo venir.
    - Bebida.
- **Polisemia:** misma forma diferentes acepciones relacionadas.
  - **Cresta**
    - **Parte del cuerpo de algunos animales que crece generalmente sobre la cabeza.**
    - **Cumbre de una ola.**
    - **Cumbre de una montaña.**
- **Sinonimia**: forma diferente, mismo significado (por ejemplo, cantante, vocalista)
  - Conceptual, referencial, contextual, de connotación.
- **Hiponimia**: una palabra denota una subclase de otra (por ejemplo, desayuno, comida)

Stop-words

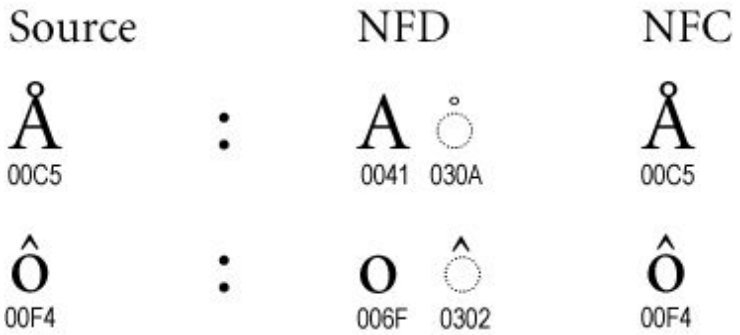
Desde el punto de vista no lingüístico, no aportan información

- Tienen un papel principalmente funcional
- Normalmente las eliminamos para ayudar a que los métodos funcionen mejor

Las palabras reservadas dependen de la lengua.

Normalización

Como tenemos muchas codificaciones de caracteres, a menudo no es trivial identificar una palabra y escribirla de forma única.



Stemming (1/2)

Las diferentes formas de una misma palabra suelen ser problemáticas para el análisis de los datos de texto, ya que tienen una ortografía diferente y un significado similar (por ejemplo, aprende, aprendía, aprender,...)

El stemming es un proceso de transformación de una palabra en su raíz (forma normalizada)

Stemming (2/2)

Para el inglés se utiliza principalmente Porter stemmer en http://www.tartarus.org/~martin/PorterStemmer/

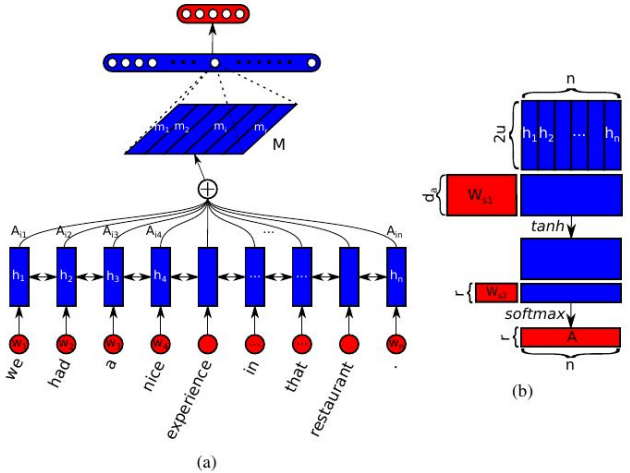
Ejemplo de reglas en cascada utilizadas en el Porter stemmer inglés

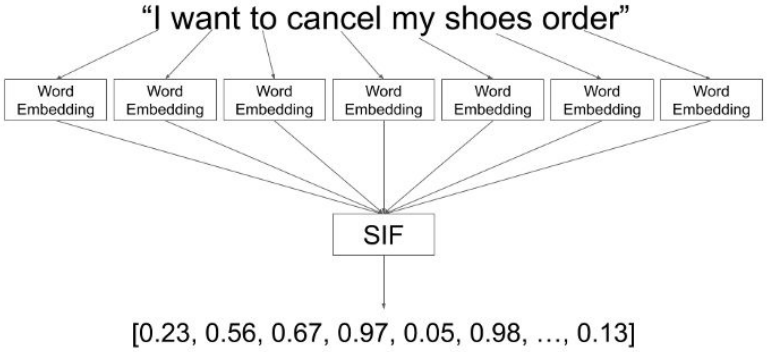
ATIONAL ->	ATE	relational -> relate
TIONAL ->	TION	conditional -> condition
ANCI ->	ANCE	hesitanci -> hesitance
IZER ->	IZE	digitizer -> digitize
ABLI ->	ABLE	conformabli -> conformable
ALLI ->	AL	radicalli -> radical
ENTLI ->	ENT	differentli -> different
ELI ->	E	vileli -> vile
OUSLI ->	OUS	analogousli -> analogous

Representación a Nivel de Frase

En lugar de tener sólo palabras sueltas podemos tratar con frases completas.

- Captura el contexto, no únicamente la palabra.
- Permite comparar vectores a nivel de frase, sin tener que hacer agregación de las palabras representadas.
- Puede capturar el orden de las palabras.





Representación por Taxonomías/thesaurus

- El tesauro tiene la función principal de conectar diferentes palabras con el mismo significado en un solo sentido (sinónimos).
- Los hiperónimos pueden relacionar los sentidos de las palabras.
- Con el uso de sinónimos e hiperónimos, compactamos los vectores.
- El tesauro general más utilizado es WordNet

Wordnet consta de 4 bases de datos (sustantivos, verbos, adjetivos y adverbios)

Cada base de datos consta synsets, con sus respectivos sinónimos.

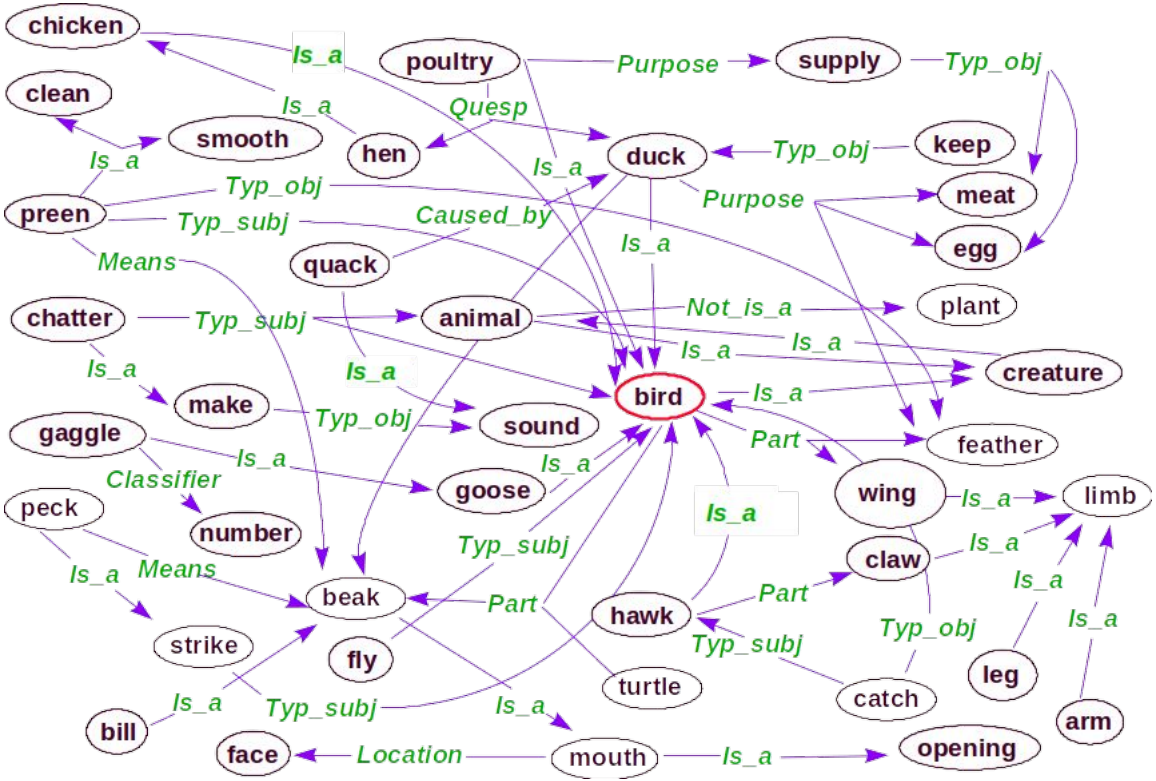
Category	Unique Forms	Number of Senses
Noun	94474	116317
Verb	10319	22066
Adjective	20170	29881
Adverb	4546	5677

Wordnet

WordNet Noun Relations

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

Wordnet (relaciones/sentidos)



Modelo de bolsa de palabras (BoW)

El modelo de bolsa de palabras (BoW) es la forma más sencilla de representación del texto en números. Al igual que el término, podemos representar una frase como un vector de bolsa de palabras.

Ejemplo crítica de películas:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

Modelo de bolsa de palabras (BoW)

Primero vamos a construir un vocabulario a partir de todas las palabras únicas de las tres críticas anteriores. El vocabulario se compone de estas 11 palabras: 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'.

Ahora podemos tomar cada una de estas palabras y marcar su aparición en las tres reseñas de películas anteriores con 1s y 0s. Esto nos dará 3 vectores para 3 críticas:

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Inconvenientes

- Si las nuevas frases contienen nuevas palabras, el tamaño de nuestro vocabulario aumentará y, por tanto, la longitud de los vectores también.
- Los vectores también contendrían muchos 0s, lo que daría lugar a una matriz dispersa.
- No conservamos ninguna información sobre la gramática de las frases ni sobre el orden de las palabras en el texto.

TF-IDF (Term Frequency-Inverse Document Frequency)

- Estadística numérica que pretende reflejar la importancia de una palabra en un documento perteneciente a una colección de documentos.
- **tf** : medida de la frecuencia con la que un término, **t**, aparece en un documento **d**.
- Aquí, en el numerador, **n** es el número de veces que el término "**t**" aparece en el documento "**d**". Así, cada documento y cada término tendrían su propio valor de **TF**.

$$tf_{t,d} = \frac{n_{t,d}}{Number\ of\ terms\ in\ the\ document}$$

Review 2: This movie is not scary and is slow

- TF('this') = 1/8
- TF('movie') = 1/8
- TF('is') = 2/8 = 1/4
- TF('very') = 0/8 = 0
- TF('scary') = 1/8
- TF('and') = 1/8
- TF('long') = 0/8 = 0
- TF('not') = 1/8
- TF('slow') = 1/8
- TF('spooky') = 0/8 = 0
- TF('good') = 0/8 = 0

IDF (Inverse Document Frequency)

IDF es una medida de la importancia de un término.

Necesitamos el valor de la IDF porque calcular sólo la TF no es suficiente para comprender la importancia de las palabras.

IDF('this') = log(número de documentos/número de documentos que contienen la palabra 'this')

IDF('this') = log(3/3) = log(1) = 0

$$idf_t = \log \frac{number\ of\ documents}{number\ of\ documents\ with\ term\ 't'}$$

Review 2: This movie is not scary and is slow

- IDF('movie',) = log(3/3) = 0
- IDF('is') = log(3/3) = 0
- IDF('not') = log(3/1) = log(3) = 0.48
- IDF('scary') = log(3/2) = 0.18
- IDF('and') = log(3/3) = 0
- IDF('slow') = log(3/1) = 0.48

TF for Movie Review

Term	Review 1	Review 2	Review 3	TF (Review 1)	TF (Review 2)	TF (Review 3)
This	1	1	1	1/7	1/8	1/6
movie	1	1	1	1/7	1/8	1/6
is	1	2	1	1/7	1/4	1/6
very	1	0	0	1/7	0	0
scary	1	1	0	1/7	1/8	0
and	1	1	1	1/7	1/8	1/6
long	1	0	0	1/7	0	0
not	0	1	0	0	1/8	0
slow	0	1	0	0	1/8	0
spooky	0	0	1	0	0	1/6
good	0	0	1	0	0	1/6

IDF

Term	Review 1	Review 2	Review 3	IDF
This	1	1	1	0.00
movie	1	1	1	0.00
is	1	2	1	0.00
very	1	0	0	0.48
scary	1	1	0	0.18
and	1	1	1	0.00
long	1	0	0	0.48
not	0	1	0	0.48
slow	0	1	0	0.48
spooky	0	0	1	0.48
good	0	0	1	0.48

Palabras como "this", "y", "es", etc., se reducen a 0 y tienen poca importancia; mientras que palabras como "scary", "long", "good", etc. son palabras con más importancia y, por tanto, tienen un valor más alto.

TF-IDF

Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.000	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	0	0.48	0.068	0.000	0.000
not	0	1	0	0.48	0.000	0.060	0.000
slow	0	1	0	0.48	0.000	0.060	0.000
spooky	0	0	1	0.48	0.000	0.000	0.080
good	0	0	1	0.48	0.000	0.000	0.080

TF-IDF

Para calcular el score **TF-IDF** de cada palabra del corpus.

$$(tf\_idf)_{t,d} = tf_{t,d} * idf_t$$

Las palabras con una puntuación más alta son más importantes, y las que tienen una puntuación más baja son menos importantes

Review 2: This movie is not scary and is slow

TF-IDF(‘this’, Review 2) = TF(‘this’, Review 2) * IDF(‘this’) = 1/8 * 0 = 0

- TF-IDF(‘movie’, Review 2) = 1/8 * 0 = 0
- TF-IDF(‘is’, Review 2) = 1/4 * 0 = 0
- TF-IDF(‘not’, Review 2) = 1/8 * 0.48 = 0.06
- TF-IDF(‘scary’, Review 2) = 1/8 * 0.18 = 0.023
- TF-IDF(‘and’, Review 2) = 1/8 * 0 = 0
- TF-IDF(‘slow’, Review 2) = 1/8 * 0.48 = 0.06

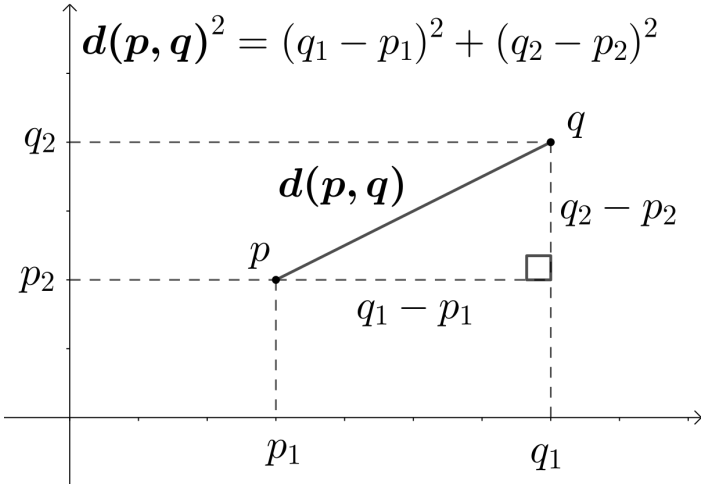
Similitud Textual en Espacios Vectoriales

Espacio Vectorial

- Tenemos un espacio vectorial |V|-dimensional
- Los términos son los ejes del espacio
- Los documentos son puntos o vectores en este espacio
- El espacio es de muy alta dimensionalidad:
  - Cientos de millones de dimensiones cuando se aplica esto a un motor de búsqueda web
  - Se trata de un vector muy disperso -la mayoría de las entradas son cero

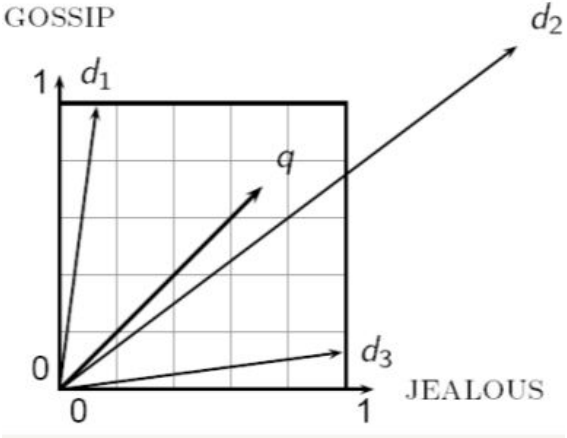
Opciones de medida?

- ¿Cómo podemos medir la proximidad de los documentos en este espacio?
- Primera opción: distancia entre dos puntos
  - ¿Distancia euclidiana?



Por qué es una mala idea la distancia Euclideana?

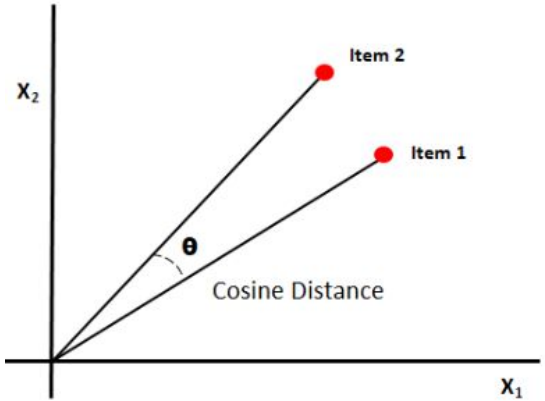
- La distancia euclidiana entre d1 y d2 es grande aunque aunque la distribución de los términos sean similares.
- Para coincidir la norma d los vectores debería ser igual.



Similitud Coseno

- Es un producto punto normalizado.
- Mide la proyección de un documento respecto a otro.
- Acotado

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$



Conclusiones

- La bolsa de palabras sólo crea un conjunto de vectores que contienen el recuento de ocurrencias de palabras en el documento, mientras que el modelo TF-IDF contiene información sobre las palabras más importantes y las menos importantes también.
- Los vectores de la bolsa de palabras son fáciles de interpretar. Sin embargo, el TF-IDF suele funcionar mejor en los modelos de aprendizaje automático.
- Aunque tanto la Bolsa de Palabras como el TF-IDF han sido populares en su propio sentido, aún quedaba un vacío en lo que respecta a la comprensión del contexto de las palabras. Detectar la similitud entre las palabras "spooky" y "scary", o traducir nuestros documentos dados a otro idioma, requiere mucha más información sobre los documentos.