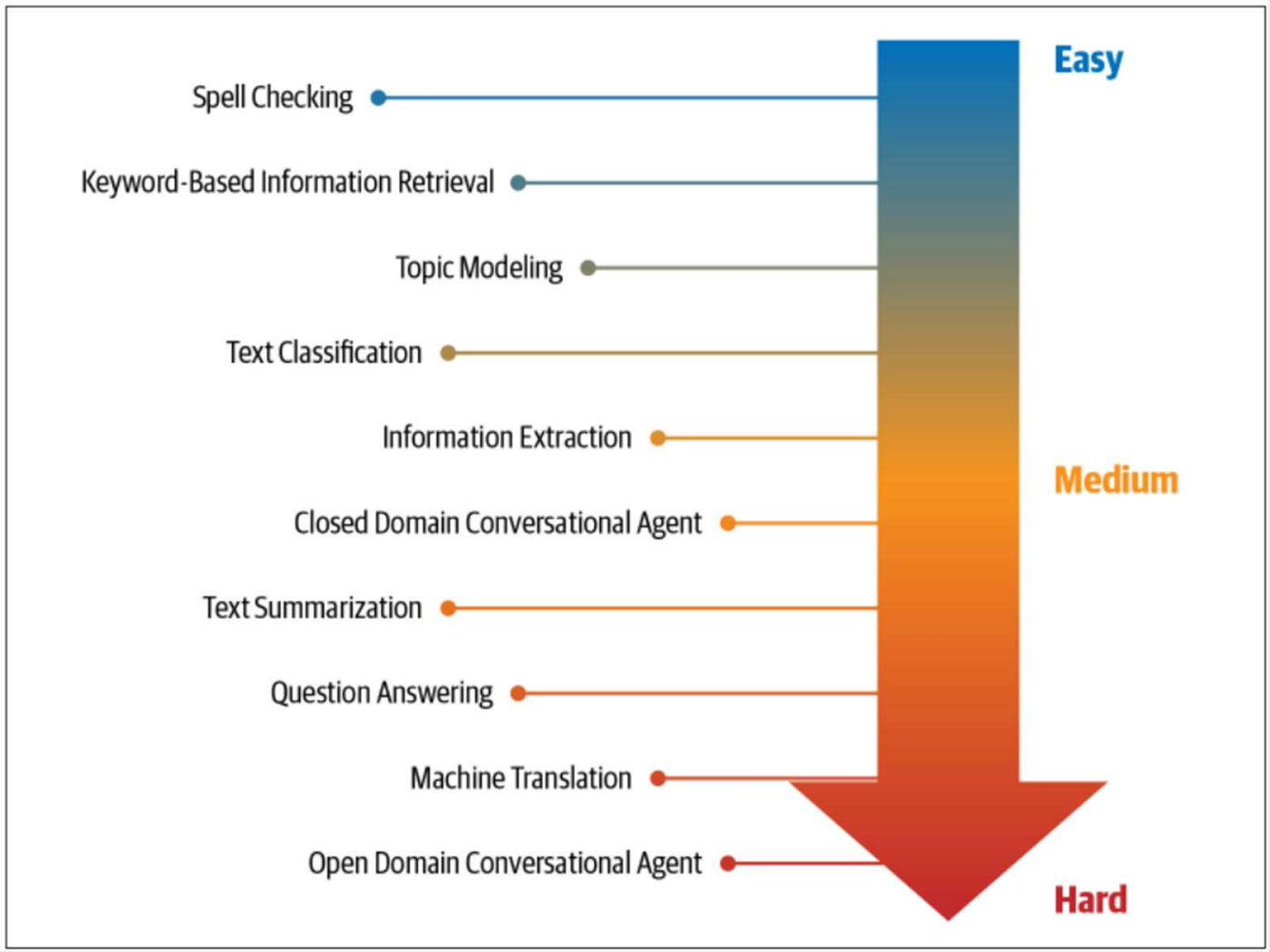
El procesamiento del lenguaje natural (NLP – Natural Language Processing)

Preguntas disparadoras:

- ¿Qué cosas parecen fáciles para nosotros pero son difíciles para una máquina?
- ¿Cómo afecta la ambigüedad?
- ¿Qué papel juega el contexto?


Spell Checking
Keyword-Based Information Retrieval
Topic Modeling
Text Classification
Information Extraction
Closed Domain Conversational Agent
Text Summarization
Question Answering
Machine Translation
Open Domain Conversational Agent
Easy
Medium
Hard

Glosario

- Corpus
  - Conjunto de textos que se utilizan para analizar, entrenar o evaluar modelos.
  - Ejemplo: noticias, libros, comentarios, tweets, artículos científicos.
- Documento
  - Unidad individual dentro del corpus (por ejemplo, un artículo, un email, una reseña).
- Oración
  - Secuencia de palabras con estructura gramatical completa.
  - Puede terminar en punto, signo de exclamación o interrogación.

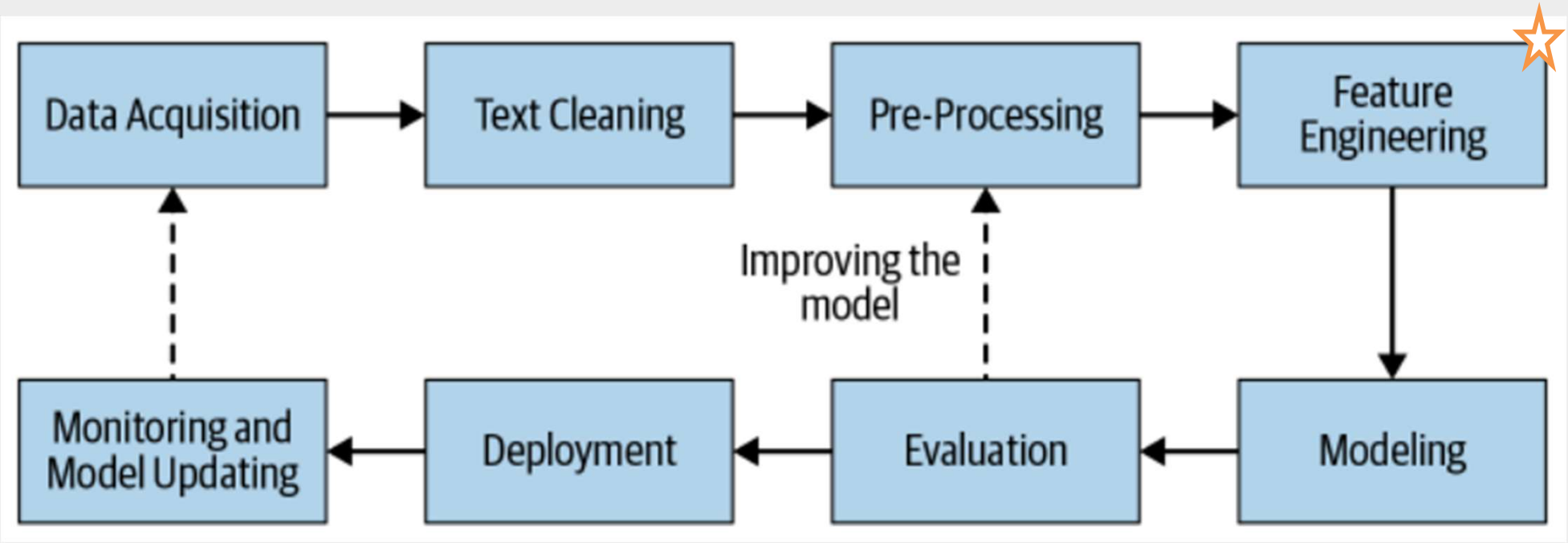
Glosario

- Token
  - Unidad básica de análisis: puede ser una palabra, puntuación o símbolo.
  - Proceso de tokenización: dividir el texto en tokens.
- Entidad
  - Elemento con significado específico: persona, lugar, organización, fecha.
- Polisemia
  - palabra que admite dos o más significados
  - Ejemplo: banco (silla, entidad financiera), bota (calzado, botar)

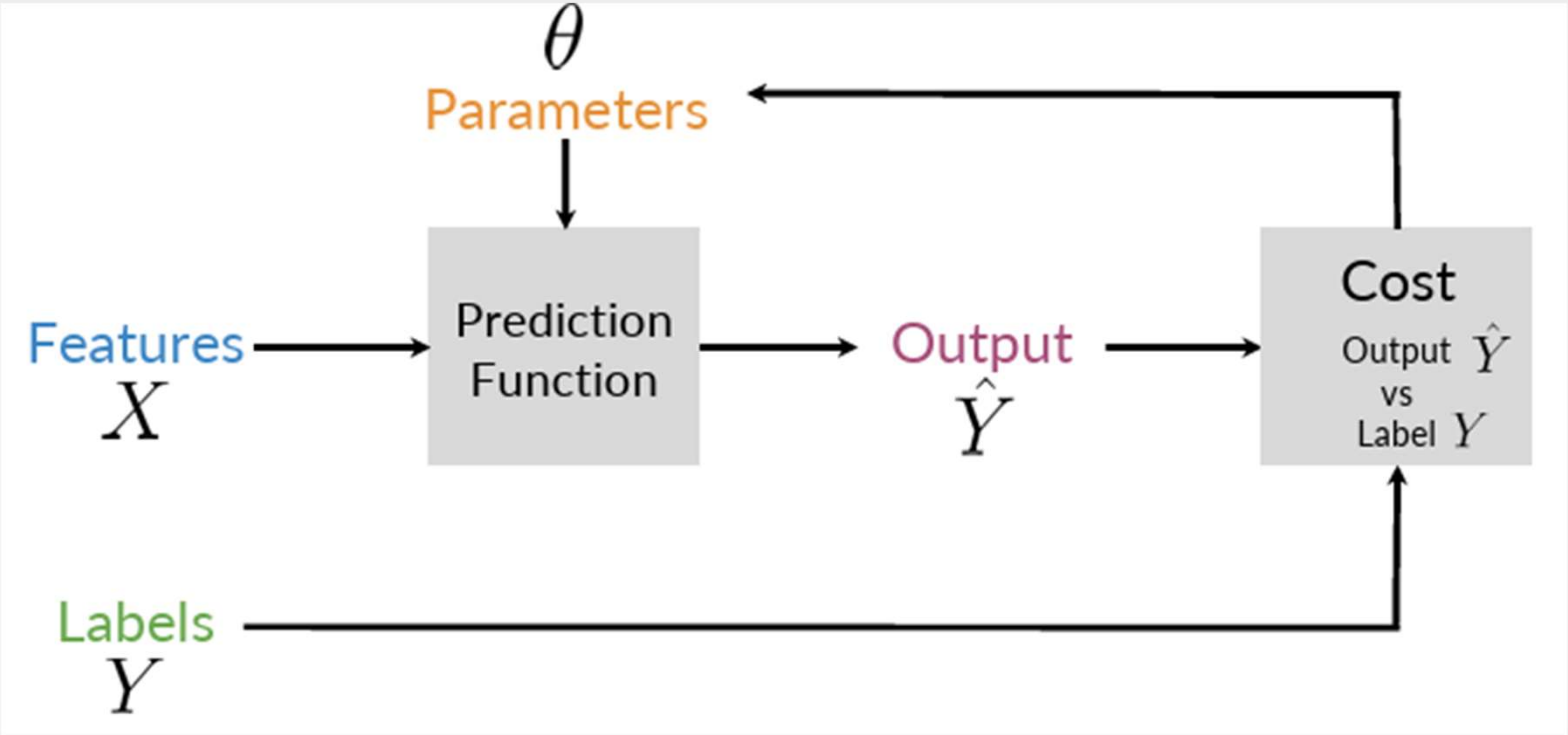
POS (part-of-speech)

- POS Tagging: asignar una categoría gramatical a cada palabra de un texto:
  - sustantivo, verbo, adjetivo, adverbio, pronombre, etc.
  - Usos
    - Análisis sintáctico
    - Extracción de información
    - Reconocimiento de entidades
    - Desambiguación de palabras
    - Traducción automática

- Pipeline genérico de NLP



- Aprendizaje Supervisado: ☆
  - Enfoque clásico



- Adquisición de datos
  - Datos propios
  - Datasets públicos
  - Scraping
  - Estrategias de Data Augmentation
    - Reemplazo por sinónimos
    - Back translation
    - Reemplazo de palabras basadas en TF-IDF
    - …

Basics - regex

- Las expresiones regulares (regex) son patrones usados para encontrar combinaciones de caracteres dentro de cadenas.
- Se utilizan para búsqueda, validación y procesamiento de texto.

- Meta-caracteres

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"planet$"
*	Zero or more occurrences	"he.*o"
+	One or more occurrences	"he.+o"
?	Zero or one occurrences	"he.?o"
{}	Exactly the specified number of occurrences	"he.{2}o"
|	Either or	"falls|stays"
()	Capture and group	

- Secuencias especiales

Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

- Sets

Set	Description
[arn]	Returns a match where one of the specified characters (`a` , `r` , or `n`) is present
[a-n]	Returns a match for any lower case character, alphabetically between `a` and `n`
[^arn]	Returns a match for any character EXCEPT `a` , `r` , and `n`
[0123]	Returns a match where any of the specified digits (`0` , `1` , `2` , or `3`) are present
[0-9]	Returns a match for any digit between `0` and `9`
[0-5][0-9]	Returns a match for any two-digit numbers from `00` and `59`
[a-zA-Z]	Returns a match for any character alphabetically between `a` and `z` , lower case OR upper case
[+]	In sets, `+` , `*` , `.` , `|` , `()` , `$` , `{}` has no special meaning, so `[+]` means: return a match for any `+` character in the string

- Ejercicios (usando Python o regex101.com)

- Librerías NLTK-spaCy

Característica	NLTK	spaCy
Enfoque	Investigación / Educación	Producción / Industria
Facilidad de uso	Más compleja, granular	API simple y directa
Velocidad	Lenta	Muy rápida
Recursos incluidos	Muchos corpus y herramientas	Menos, pero bien integrados
Tareas comunes (POS, NER, etc.)	Requiere configuración manual	Preentrenadas y listas para usar
Unidad principal	Strings	Docs y tokens como objetos

Preprocesamiento - Pasos básicos previos

- Transformación a minúsculas
- Sentence segmentation (*)
- Word tokenization (*)

(*) Tener en cuenta que la tokenización de oraciones o palabras dependerá de idiomas/contextos. Ej: N.Y.!

Stop Words

- palabras que se consideran poco informativas
- se suelen eliminar durante el preprocesamiento del texto
- aparecen con mucha frecuencia, pero aportan poca información semántica
- Dependen del idioma

Stop Words

- Ejemplo:
  - Artículos: el, la, los, un
  - Preposiciones: de, en, a, con
  - Conjunciones: y, o, pero
  - Verbos comunes: ser, estar, haber, tener
  - Pronombres: yo, tú, él, nosotros, etc

Stemming

- Reducir una palabra a su raíz o "stem", eliminando sufijos o terminaciones
- no considera reglas gramaticales complejas

Stemming

- Ejemplo

Palabra original	Stem (raíz)
playing	play
played	play
happily	happi
runner	run
national	nation

Stemming

- ¿Para qué se usa?
  - Reducir la dimensionalidad del texto en tareas de clasificación o recuperación
  - Agrupar variantes morfológicas de una misma palabra
  - Acelerar procesamiento en sistemas donde no se necesita precisión lingüística (como motores de búsqueda simples).

Lematización

- Reducir una palabra a su forma base o canónica, conocida como lema.

¿Para qué se usa?

- Normalizar texto para análisis semántico o clasificación.
- Mejorar resultados en sistemas de recuperación de información.
- Reducir la dimensión del vocabulario en modelos de ML/NLP.

Lematización

- Ejemplo

Palabra original	Lema	Tipo de palabra
cantando	cantar	verbo
corriendo	correr	verbo
mejores	mejor	adjetivo
niños	niño	sustantivo
fue	ser	verbo irregular

- La lematización depende del tipo de palabra

Palabra	Tipo (POS)	Lema
banco	sustantivo	banco
banco	verbo	bancar
mejores	adjetivo	mejor
mejores	verbo (mejorar)	mejorar

- Stemming vs. Lematización

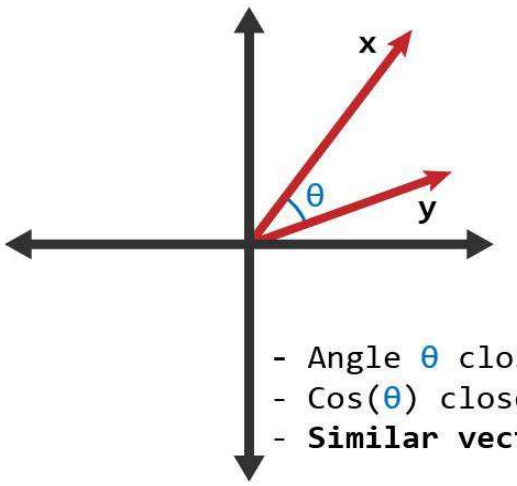
Característica	**Lematización**	**Stemming**
Basado en reglas lingüísticas	**Sí**	No
Produce palabras reales	**Sí**	No necesariamente
Más preciso	**Generalmente sí**	Aproximado
Más lento	Más costoso computacionalmente	**Rápido**

Vector Space Models

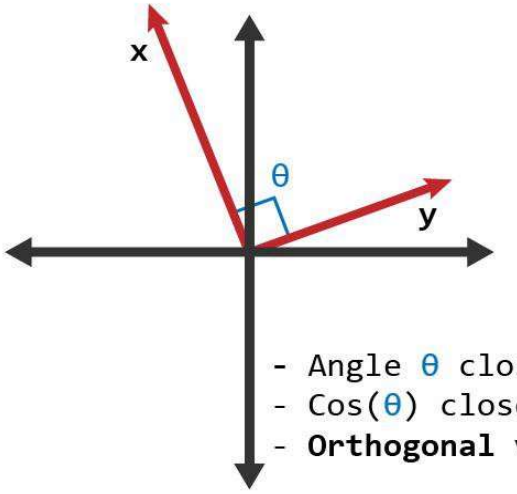
- Representan textos como vectores en un espacio n-dimensional
- Cada dimensión puede corresponder a:
  - una palabra/token (modelo BoW)
  - una frecuencia ponderada (TF-IDF)
  - una dirección semántica (embeddings)
  - …
- Permiten calcular similitud entre textos

- Similaridad entre representación de texto
  - Distancia de coseno

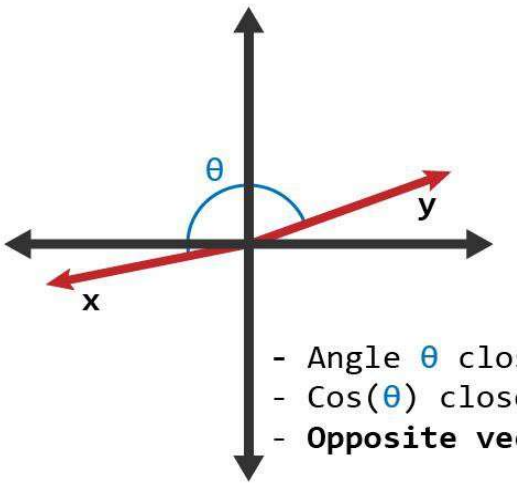
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



- Angle θ close to 0
- Cos(θ) close to 1
- **Similar vectors**



- Angle θ close to 90
- Cos(θ) close to 0
- **Orthogonal vectors**



- Angle θ close to 180
- Cos(θ) close to -1
- **Opposite vectors**

One-hot enconding (solo a efectos de completitud)

- representa palabras/tokens como vectores binarios
- cada dimensión representa una palabra del vocabulario
- la posición activa (valor 1) indica la presencia de esa palabra

- Ejemplo
  - ["el gato duerme“, "el perro ladra“, "el gato maúlla allí "]
  - vocabulario = ["el", "gato", "duerme", "perro", "ladra", "maúlla", "allí"]
  - "el gato duerme" sería:

```
[
 [1, 0, 0, 0, 0, 0],  # el
 [0, 1, 0, 0, 0, 0],  # gato
 [0, 0, 1, 0, 0, 0]   # duerme
]
```

  - Qué sucede con "el gato maúlla allí " ?

Bag of Words (BoW)

- convierte documentos en vectores de frecuencia:
  - cuenta la frecuencia de cada palabra del vocabulario
  - representa el documento como un vector de frecuencias.

- Ejemplo
- "el gato duerme“
- "el perro ladra“
- Vocabulario: ["el", "gato", "duerme", "perro", "ladra"]

Documento	el	gato	duerme	perro	ladra
"el gato duerme"	1	1	1	0	0
"el perro ladra"	1	0	0	1	1

Bag of N-Grams

- Extensión del modelo Bag of Words (BoW)
- Representa el texto como una colección de N-gramas: secuencias de *N* palabras consecutivas
- Captura información de contexto local y orden parcial de palabras

- (1) "El ciudadano no cumplió con su deber“
- (2) "El no ciudadano cumplió con su deber“
- BoW

Frase	el	Ciudadano	No	cumplió	con	su	deber
(1)	1	1	1	1	1	1	1
(2)	1	1	1	1	1	1	1

- Bag-of-Bigrams (BoN, N = 2)

Frase	el ciudadano	ciudadano no	no cumplió	cumplió con	con su	su deber	el no	no ciudadano	ciudadano cumplió
(1)	1	1	1	1	1	1	0	0	0
(2)	0	0	0	1	1	1	1	1	1

TF-IDF

- Term Frequency – Inverse Document Frequency
- técnica para representar documentos como vectores numéricos que reflejan la importancia relativa de cada palabra

- **TF (Frecuencia de término)**
  Cuántas veces aparece un término en un documento

$$\mathrm{TF}(t, d) = \frac{\text{frecuencia de } t \text{ en } d}{\text{número total de palabras en } d}$$

- **IDF (Frecuencia inversa de documento)**
  Penaliza las palabras comunes en muchos documentos.

$$\mathrm{IDF}(t) = \log\left(\frac{N}{1 + \text{número de documentos que contienen } t}\right)$$

- TF-IDF final: $\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$

- Reduce el peso de palabras frecuentes y poco informativas
  - Aumenta el peso de palabras específicas e importantes en cada documento (mayor discriminabilidad)
  - Reduce el peso de palabras comunes entre documentos