Minería de Texto

Data Science (CC3084) - 2026

Minería de Texto

Semestre 02, 2026

Definición

La minería de texto (text mining) es el proceso de extraer información útil, patrones y conocimiento a partir de grandes colecciones de texto escrito en lenguaje natural.

El punto de partida es un problema: el texto no viene en filas y columnas. Un correo, una reseña, un expediente médico o un tweet no tienen columnas numéricas listas para un modelo. La minería de texto es el conjunto de técnicas que convierte ese texto en datos que un algoritmo puede analizar.

Por qué importa

- Se estima que la mayor parte de la información que generan las organizaciones es texto no estructurado: correos, contratos, tickets de soporte, notas clínicas, redes sociales.
- Ese texto contiene conocimiento que los métodos tabulares clásicos no pueden tocar directamente.
- Analizarlo manualmente no escala: nadie lee un millón de reseñas.

La idea central

Todo el campo se apoya en una sola transformación:

```
Texto crudo  ->  Texto limpio  ->  Números  ->  Modelo  ->  Conocimiento
```

Una vez que el texto es un vector de números, podemos aplicar sobre él cualquier método que ya conocemos: clasificación, regresión, agrupamiento, reducción de dimensionalidad.

Texto como Dato No Estructurado

Entender qué hace difícil al texto explica por qué existe una disciplina entera dedicada a él.

Características del texto

- No estructurado: no hay esquema, ni tipos de dato, ni columnas fijas.
- Alta dimensionalidad: un vocabulario típico tiene decenas de miles de términos, y cada término es una dimensión.
- Disperso (sparse): un documento usa una fracción mínima del vocabulario, así que casi todo el vector es cero.
- Ambiguo: la misma palabra cambia de significado según el contexto.
- Dependiente del orden.
- Ruidoso: faltas de ortografía, abreviaturas, emojis, mezcla de idiomas, jerga.

La ambigüedad del lenguaje

```
"Fui al banco a sacar dinero."        ->  institución financiera
"Me senté en el banco del parque."    ->  mueble
```

```
"Un banco de peces cruzó la bahía."   ->  grupo de animales
```

La misma cadena de letras significa tres cosas distintas. Solo el contexto lo resuelve, y eso es justo lo que un conteo de palabras no captura.

El orden también significa

```
"El perro mordió al hombre."
"El hombre mordió al perro."
```

Las dos frases tienen exactamente las mismas palabras y significan cosas opuestas. Cualquier método que solo cuente palabras las considera idénticas.

Consecuencia práctica

No podemos alimentar texto directamente a un modelo. Siempre hay una etapa de preprocesamiento y vectorización antes del aprendizaje, y buena parte del esfuerzo de un proyecto de minería de texto vive ahí.

Diferencias con Otros Métodos

La minería de texto se confunde con varias disciplinas vecinas. Vale la pena separarlas.

Minería de texto vs. minería de datos

| Aspecto | Minería de datos | Minería de texto | |---|---|---| | Entrada | Datos estructurados (tablas) | Texto libre en lenguaje natural | | Preprocesamiento | Limpieza, imputación, escalado | Tokenización, normalización, vectorización | | Dimensionalidad | Decenas o cientos de variables | Decenas de miles de términos

| | Densidad | Matriz densa | Matriz dispersa | | Ambigüedad | Baja: cada columna significa una cosa | Alta: polisemia, sarcasmo, contexto |

La minería de texto puede verse como minería de datos precedida por una etapa que fabrica las variables a partir del texto.

Minería de texto vs. procesamiento de lenguaje natural (NLP)

- NLP es el campo que estudia cómo hacer que una máquina procese lenguaje humano: análisis sintáctico, morfología, semántica, generación de texto.
- La minería de texto es una aplicación orientada a descubrir conocimiento en colecciones grandes de documentos.
- La minería de texto usa NLP como herramienta; NLP es más amplio y también persigue objetivos lingüísticos que no buscan descubrir patrones (traducción, generación, resumen).

Minería de texto vs. recuperación de información (IR)

- Recuperación de información responde: dada una consulta, ¿qué documentos son relevantes? Es lo que hace un buscador.
- Minería de texto responde: dada una colección, ¿qué patrones o conocimiento nuevo contiene?
- IR encuentra lo que ya está escrito; la minería de texto produce información que no estaba explícita en ningún documento individual.

```
Colección: 50,000 reseñas de un hotel

Recuperación:  "reseñas que mencionen el desayuno"
               -> devuelve las 3,200 reseñas que lo mencionan

Minería:       "¿de qué se queja la gente?"
               -> descubre que las quejas por ruido subieron
```

```
40% desde que abrió el bar del cuarto piso
```

Ninguna reseña individual dice eso. Es conocimiento que solo aparece al analizar la colección completa.

Minería de texto vs. minería web

La minería web abarca tres ramas: contenido, estructura (enlaces) y uso (navegación). La minería de texto se corresponde con la primera rama cuando el contenido es texto, pero se aplica también fuera de la web.

Resumen de la distinción

| Disciplina | Pregunta que responde | |---|---| | Recuperación de información| ¿Qué documento responde a mi consulta? | | NLP | ¿Cómo entiende la máquina esta oración? | | Minería de texto | ¿Qué patrones esconde esta colección? | | Minería de datos | ¿Qué patrones esconde esta tabla? |

Aplicaciones

La minería de texto está detrás de productos que usamos todos los días.

- Filtrado de spam: clasificar correos como deseados o no deseados.
- Análisis de opinión de clientes: reseñas de productos, encuestas abiertas, redes sociales.
- Clasificación y enrutamiento de tickets de soporte hacia el equipo correcto.
- Detección de noticias falsas y de contenido tóxico en plataformas.
- Análisis de expedientes clínicos para identificar síntomas, diagnósticos o efectos adversos.

- Revisión de contratos y documentos legales (legal tech).
- Análisis de currículums para preselección de candidatos.
- Vigilancia tecnológica y análisis de literatura científica.
- Predicción de mercados a partir de noticias financieras.
- Sistemas de recomendación basados en descripciones de contenido.

Todas parten de texto y terminan en una decisión: una etiqueta, un número, un grupo o un ranking.

El Proceso de Minería de Texto

El flujo de trabajo es estable, sin importar la aplicación.

1. Recolección: obtener los documentos (scraping, APIs, bases de datos, archivos).
2. Preprocesamiento: limpiar y normalizar el texto.
3. Representación: convertir texto en vectores numéricos.
4. Reducción de dimensionalidad: quedarse con lo informativo.
5. Modelado: aplicar clasificación, regresión, agrupamiento o modelado de tópicos.
6. Evaluación: medir el desempeño con métricas adecuadas.
7. Interpretación: traducir el resultado en conocimiento accionable.

En proyectos reales, las etapas 1 a 3 consumen la mayor parte del tiempo. La elección del algoritmo suele importar menos que la calidad de la representación.

Preprocesamiento

El objetivo es reducir el ruido y la variabilidad innecesaria del texto sin perder la señal. Es donde se decide qué información sobrevive hasta el modelo.

Las técnicas

- Limpieza de ruido
- Normalización
- Tokenización
- Detección de idioma y traducción
- Eliminación de palabras vacías
- Stemming y lematización
- N-gramas
- Corrección de errores de escritura

Limpieza de ruido

Quitar todo lo que no es lenguaje: etiquetas, enlaces, menciones, símbolos.

```
Original:  @tiendaXY Mira esto!!! <b>OFERTA</b>
           https://tienda.co/x3f □□□#descuento

Limpio:    Mira esto OFERTA descuento
```

Qué se borra depende de la tarea. En análisis de sentimientos los emojis cargan opinión y conviene conservarlos o traducirlos a palabras.

Normalización

Unificar formas que significan lo mismo para que el modelo las cuente como un solo término.

```
Mayúsculas:   "PRODUCTO"  "Producto"  "producto"   ->  producto
Acentos:      "cafe"      "café"      "CAFÉ"       ->  cafe
Puntuación:   "bueno!!!"  "bueno..."  "bueno"      ->  bueno
Números:      "2024"      "1998"                   ->  <NUM>
```

Cuidado: quitar acentos en español fusiona palabras distintas: papa y papá, esta y está, publico y público.

Tokenización

Partir el texto en unidades mínimas de análisis, llamadas tokens. Es la primera decisión estructural: define cuáles son las piezas con las que trabaja todo lo demás.

```
"El servicio no fue bueno."

Tokens:  [ El ] [ servicio ] [ no ] [ fue ] [ bueno ] [ . ]
```

Niveles de tokenización

```
Texto:     "No me gustó. El envío tardó mucho."

Oraciones: [ No me gustó. ]  [ El envío tardó mucho. ]

Palabras:  [ No ] [ me ] [ gustó ] [ El ] [ envío ] [ tardó ] [ mucho ]

Caracteres:[ N ] [ o ] [   ] [ m ] [ e ] ...
```

Palabras es lo habitual. Oraciones sirve para análisis de sentimientos por oración. Caracteres se usa en lenguas sin espacios y en detección de errores de escritura.

Tokenizar no es cortar por espacios

```
"Nueva York"        dos palabras, un solo concepto
"del"               contracción de "de" + "el"
"correo@uvg.edu.gt" un token, no cuatro
"3.14"              un número, no "3" y "14"
"rock and roll"     tres palabras, un solo nombre
"no-conformidad"    ¿uno o dos tokens?
```

El chino y el japonés se escriben sin espacios entre palabras, así que ahí el corte requiere un modelo entrenado, no una regla.

Tokenización por subpalabras

Es lo que usan los modelos modernos: parten las palabras raras en fragmentos frecuentes.

```
"inconstitucionalmente"

->  [ in ] [ constitucional ] [ mente ]
```

Ventaja: ninguna palabra queda fuera del vocabulario. Incluso una que el modelo nunca vio se representa como suma de fragmentos conocidos.

Detección de idioma

Una colección real casi nunca viene en un solo idioma. Antes de procesar hay que saber en cuál está cada documento.

```
"El producto llegó roto."           ->  español
"The product arrived broken."       ->  inglés
"O produto chegou quebrado."        ->  portugués
"El delivery llegó bien pero meh"   ->  español con inglés
```

Importa porque las listas de palabras vacías, los lematizadores y los léxicos de sentimiento son específicos de cada idioma. Aplicar los de español a un texto en

inglés produce basura.

Traducción

Llevar todo a un idioma común permite analizar la colección completa con un solo conjunto de herramientas.

```
Colección multilingüe          Traducida al español

"The room was filthy"     ->   "El cuarto estaba asqueroso"
"O quarto estava sujo"    ->   "El cuarto estaba sucio"
"La chambre était sale"   ->   "El cuarto estaba sucio"

Ahora las tres quejas se agrupan como un mismo problema.
```

A favor: un solo idioma significa un solo léxico, un solo modelo y resultados comparables entre países.

En contra: la traducción introduce errores propios y aplana matices. La ironía, la jerga local y los juegos de palabras rara vez sobreviven.

Alternativa moderna: usar un modelo multilingüe, que representa frases de distintos idiomas en un mismo espacio sin traducirlas.

Eliminación de palabras vacías (stopwords)

Las palabras vacías son términos muy frecuentes que aportan estructura gramatical pero casi ninguna información sobre el tema del documento.

```
Español:  el, la, los, de, que, y, a, en, un, por, con, para...
Inglés:   the, of, and, to, in, is, that, for, it, with...
```

Cómo se ve el filtrado

```
Original:   "el servicio de la tienda fue muy bueno y rápido"

Vacías:      el · de · la · fue · muy · y

Resultado:  "servicio tienda bueno rápido"
```

Reduce el vocabulario y el tamaño de la matriz, y deja solo los términos que hablan del contenido.

El peligro de quitarlas

```
Original:   "la película no me gustó nada"
Filtrado:   "película gustó"          <- ¡el sentido se invirtió!

Original:   "el cuarto no estaba sucio"
Filtrado:   "cuarto sucio"            <- dice lo contrario
```

En análisis de sentimientos, las negaciones (no, sin, nunca, tampoco) deben salir de la lista de palabras vacías. Son justo lo que invierte la polaridad.

La lista se ajusta al dominio

```
Colección de reseñas de hoteles:
  "hotel" aparece en el 98% de los documentos
  -> no distingue nada, funciona como palabra vacía

Colección de expedientes clínicos:
  "paciente", "consulta", "doctor"
  -> mismo caso: son ruido de fondo del dominio
```

Además de la lista estándar del idioma, conviene agregar los términos que saturan la colección concreta.

Stemming

Recortar la palabra a una raíz aproximada aplicando reglas de corte de sufijos. Es rápido y burdo: la raíz no tiene por qué ser una palabra real.

```
corriendo · corredor · corremos · correrá   ->   corr
jugando   · jugador  · juegos               ->   jug
niñas     · niños    · niñito               ->   niñ
```

Los errores del stemming

```
Corta de más (overstemming):
  universidad · universo · universal  ->  univers
  Tres conceptos distintos quedan como uno solo.

Corta de menos (understemming):
  fui · iba · ir
  El verbo irregular no se reduce: quedan como tres términos.
```

El español tiene mucha más flexión verbal que el inglés, así que el stemming es aquí más agresivo y más propenso a error.

Lematización

Llevar la palabra a su forma de diccionario, el lema, usando análisis morfológico y el contexto de la oración. Es más lento y más correcto.

```
corriendo · corrió · correrá   ->   correr
fui       · iba    · vamos     ->   ir
mejores   · mejor              ->   bueno
casas     · casita             ->   casa
```

Stemming frente a lematización

```
Palabra        Stemming     Lematización

corriendo      corr         correr
mejores        mejor        bueno
fuimos         fu           ir
estudios       estudi       estudio
```

| Criterio | Stemming | Lematización | |----------|----------------------|------------------------ | | Método | Reglas de corte | Diccionario y contexto | | Resultado | Puede no ser palabra | Siempre palabra real | | Velocidad | Muy rápida | Más lenta | | Precisión | Baja | Alta |

N-gramas

Tratar secuencias de n palabras consecutivas como una unidad, para recuperar parte del orden que se pierde al contar palabras sueltas.

```
Frase:      "no me gustó el servicio"

Unigramas:  no | me | gustó | el | servicio

Bigramas:   no me | me gustó | gustó el | el servicio

Trigramas:  no me gustó | me gustó el | gustó el servicio
```

Para qué sirven

```
Con unigramas:
  "no me gustó"     ->  no · me · gustó   (gustó parece positivo)

Con bigramas:
  "no me gustó"     ->  no me · me gustó  (la negación sobrevive)

Expresiones que solo existen juntas:
```

```
  "Nueva York" · "no obstante" · "servicio al cliente"
```

Costo: cada nivel multiplica el vocabulario. Pasar de unigramas a bigramas puede llevar de 20,000 a cientos de miles de términos.

Corrección y jerga

En redes sociales y chats, el mismo concepto aparece escrito de muchas formas.

```
Original:   "q buenoo stá el prodcto, 100% recomendadoo!!! xd"

Corregido:  "que bueno está el producto, recomendado"

Variantes del mismo término:
  "porfa" · "xfa" · "porfavor" · "por favor"   ->  por favor
  "buenisimoo" · "buenísimo" · "wenisimo"      ->  buenísimo
```

Sin esta normalización, cada variante ocupa una dimensión distinta del vocabulario y el modelo nunca junta suficientes ejemplos de ninguna.

El proceso completo sobre una frase

```
Original       @tienda Los ZAPATOS que compré NO llegaron!!! □□
               https://x.co/a

Limpieza       Los ZAPATOS que compré NO llegaron

Normalización  los zapatos que compre no llegaron

Tokenización   [los] [zapatos] [que] [compre] [no] [llegaron]

Vacías         [zapatos] [no] [llegaron]      (se conserva "no")

Lematización   [zapato] [no] [llegar]

Bigramas       [zapato no] [no llegar]
```

Advertencias

- Quitar palabras vacías puede destruir la señal: no y sin invierten la polaridad de una opinión.
- El stemming agresivo une palabras de significado distinto.
- Cada técnica depende del idioma: no se puede reutilizar la de otro sin revisarla.
- Las decisiones dependen de la tarea: no existe un preprocesamiento universal.

Cómo decidir cada paso

| Tarea | Palabras vacías | Reducción | N-gramas | |---|---|---|---| | Clasificar temas | Quitar | Lematizar | Unigramas | | Análisis de sentimientos | Conservar negaciones | Lematizar | Bigramas | | Modelado de tópicos | Quitar de forma agresiva | Lematizar | Unigramas | | Detección de autoría | Conservar todo | Ninguna | N-gramas de caracteres |

La pregunta siempre es la misma: lo que estoy borrando, ¿es ruido para esta tarea, o es justo la señal que necesito?

Representación del Texto

Es la etapa que convierte documentos en vectores. Tres enfoques principales.

Bolsa de palabras (Bag of Words)

Cada documento se representa por la frecuencia de cada término del vocabulario.

```
Vocabulario: [ buena · mala · película · actuación ]

"buena película"    ->  [ 1, 0, 1, 0 ]
"mala actuación"    ->  [ 0, 1, 0, 1 ]
"buena actuación"   ->  [ 1, 0, 0, 1 ]
```

Se pierde el orden por completo. Para este método, la película mordió al actor y el actor mordió a la película son el mismo documento.

TF-IDF

Corrige el problema de la bolsa de palabras: los términos frecuentes en todos los documentos aportan poco. TF-IDF pondera cada término por su frecuencia en el documento y penaliza los que aparecen en muchos documentos.

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \log\frac{N}{\text{df}(t)}$$

donde $\text{tf}(t,d)$ es la frecuencia del término t en el documento d, N es el número total de documentos y $\text{df}(t)$ es el número de documentos que contienen t.

Con un ejemplo

```
Colección: 1,000 reseñas de restaurantes

Término     Aparece en      Peso IDF    ¿Informativo?

comida       950 reseñas      muy bajo    no distingue nada
bueno        780 reseñas      bajo        casi nada
rancio        14 reseñas      muy alto    sí, mucho
cucaracha      3 reseñas      altísimo    sí, señal fuerte

En la reseña "la comida estaba rancia":
  comida -> peso bajo · rancia -> peso alto
```

El vector del documento queda dominado por rancia, que es exactamente lo que lo hace distinto de las otras 999 reseñas.

Embeddings

Representan cada palabra o documento como un vector denso de pocas dimensiones (100 a 1024) donde la cercanía geométrica refleja cercanía semántica.

- Word2Vec, GloVe, FastText: un vector fijo por palabra, aprendido del contexto.
- Embeddings contextuales (BERT y modelos derivados): el vector de una palabra cambia según la oración, lo que resuelve la ambigüedad.
- Embeddings de oración o documento (Sentence-BERT): un vector por texto completo, útil para similitud y agrupamiento.

Qué captura un embedding

```
Palabras más cercanas a "médico":
  doctor · enfermera · hospital · clínica · paciente

Palabras más cercanas a "rancio":
  podrido · echado a perder · vencido · descompuesto

Relaciones que el espacio aprende solo:
  rey - hombre + mujer  ~  reina
```

Para TF-IDF, médico y doctor son dos términos sin relación alguna. Para un embedding están casi en el mismo punto del espacio.

Embeddings contextuales

```
"Fui al banco a sacar dinero."
   banco  ->  cerca de: cajero · cuenta · sucursal

"Me senté en el banco del parque."
   banco  ->  cerca de: silla · plaza · jardín
```

Es la solución al problema de la ambigüedad que planteamos al inicio: la misma palabra recibe dos representaciones distintas.

Comparación

Representación	Dimensión	Captura orden	Captura semántica	Costo
Bag of Words	Vocabulario (alta)	No	No	Bajo
TF-IDF	Vocabulario (alta)	No	No	Bajo
Word2Vec/GloVe	100–300	No	Sí	Medio
BERT	768+	Sí	Sí, contextual	Alto

Clasificación de Texto

Es la tarea supervisada más común: asignar una etiqueta discreta a un documento.

```
"Gana dinero rápido, haz clic aquí"          ->  spam
"Reunión mañana a las 10 en el aula 302"     ->  no spam
```

Tipos de clasificación

```
Binaria      "Gana dinero rápido"        ->  spam / no spam

Multiclase   "Guatemala ganó 2 a 0"      ->  deportes
             "El quetzal se devaluó"     ->  economía
```

```
Multietiqueta
              "El ministro de salud
               renunció tras el escándalo"
                     ->  [ política, salud ]
```

Algoritmos habituales

- Naive Bayes multinomial: rapidísimo, muy buena línea base para texto, asume independencia entre términos.
- Regresión logística: sólida, interpretable a través de los coeficientes por término.
- SVM lineal: históricamente el mejor sobre TF-IDF en datos de alta dimensionalidad.
- Random Forest y Gradient Boosting: funcionan, pero sufren con matrices muy dispersas.
- Redes neuronales (CNN 1D, LSTM): capturan orden y patrones locales.
- Transformers ajustados (fine-tuning de BERT): el estado del arte actual cuando hay datos y cómputo.

Qué aprende el modelo

Un clasificador lineal aprende un peso por término. Los pesos son legibles y explican la decisión.

```
Clasificador de spam — términos con más peso

Hacia spam:      gratis · gana · clic · urgente · premio
Hacia no spam:   reunión · adjunto · informe · gracias
```

Poder mirar esa lista es lo que permite auditar el modelo y detectar que aprendió algo absurdo antes de ponerlo en producción.

Recomendación

Empieza siempre con TF-IDF más regresión logística o Naive Bayes como línea base. Un transformer solo se justifica si supera claramente esa referencia.

Predicción a partir de Texto

Clasificar no es lo único que se puede predecir. La predicción abarca cualquier salida estimada a partir del texto.

Predicción de valores numéricos (regresión)

```
"Llegó tarde y la caja venía abierta"          ->  2 estrellas
"Cumple, nada especial"                        ->  3 estrellas
"Excelente calidad, superó mis expectativas"   ->  5 estrellas
```

- Predecir el tiempo de resolución de un ticket a partir de su descripción.
- Estimar el precio de un inmueble usando la descripción del anuncio.

Predicción con texto como variable adicional

Muy común en la práctica: se combinan variables tabulares con variables derivadas del texto en un mismo modelo.

```
[ edad, ingreso, región ]  +  [ términos del comentario ]

              ->  modelo  ->  predicción
```

Predicción de eventos futuros

```
Ticket de hoy                          Predicción
```

```
"Es la tercera vez que escribo
 y nadie responde"            ->   alta probabilidad de
                                   cancelar el servicio
```

- Anticipar movimientos de mercado con noticias financieras.
- Detectar riesgo de deserción estudiantil en respuestas abiertas de encuestas.

Predicción de la siguiente palabra

```
"El clima hoy está muy ___"

  caluroso   40%
  frío       25%
  agradable  15%
  húmedo      8%
```

Es la tarea de modelado de lenguaje: estimar $P(w_t \mid w_1, \dots, w_{t-1})$. Es el objetivo de entrenamiento sobre el que se construyeron los modelos de lenguaje modernos.

Tareas no supervisadas asociadas

- Agrupamiento (clustering) de documentos similares.
- Modelado de tópicos: descubrir los temas latentes de una colección sin etiquetas.
- Detección de anomalías en texto.

```
Colección: 20,000 noticias sin etiquetar

Tópico 1   gol · partido · equipo · jugador · torneo    ->  Deportes
Tópico 2   paciente · síntoma · dosis · hospital        ->  Salud
Tópico 3   acción · mercado · inversión · bolsa         ->  Finanzas
Tópico 4   voto · congreso · partido · elección         ->  Política
```

Partido aparece en dos tópicos con sentidos distintos. La interpretación siempre requiere criterio humano.

Análisis de Sentimientos

El análisis de sentimientos (opinion mining) determina la actitud, opinión o emoción expresada en un texto.

```
"Me encantó, volvería sin dudarlo"     ->  positivo
"Llegó tarde y frío"                   ->  negativo
"El pedido llegó el martes"            ->  neutro
```

Qué se puede detectar

- Polaridad: positivo, negativo, neutro.
- Intensidad: qué tan fuerte es esa polaridad.
- Emociones específicas: alegría, enojo, tristeza, miedo, sorpresa.
- Subjetividad: si el texto expresa una opinión o un hecho.
- Intención: queja, elogio, solicitud, amenaza.

Polaridad e intensidad

```
"Está bien"                    ->  positivo débil
"Está bueno"                   ->  positivo
"Está buenísimo"               ->  positivo fuerte
"Es lo mejor que he probado"   ->  positivo muy fuerte
```

Un cliente que dice está bien y otro que dice es lo mejor que he probado no valen lo mismo para el negocio, aunque ambos sean positivos.

Niveles de análisis

- Nivel de documento: una sola polaridad para todo el texto.
- Nivel de oración: una polaridad por oración.
- Nivel de aspecto (ABSA): polaridad por cada aspecto mencionado. Es el más útil y el más difícil.

```
"La comida estaba deliciosa pero el servicio fue lentísimo
 y el lugar es carísimo."

Documento          ->  mixto
Aspecto comida     ->  positivo
Aspecto servicio   ->  negativo
Aspecto precio     ->  negativo
```

Una etiqueta única para todo el texto pierde información accionable. Al negocio le sirve saber qué aspecto concreto falla.

Enfoque 1: basado en léxicos

Se usa un diccionario que asigna una polaridad a cada palabra y se combinan las puntuaciones del texto.

```
Léxico:   excelente +3   ·   bueno +1
          malo      -1   ·   pésimo -3

"El servicio fue excelente pero la comida mala"
        +3                                -1     ->  +2  positivo
```

- Ventajas: no requiere datos etiquetados, es interpretable y funciona de inmediato.
- Desventajas: depende del dominio, no entiende contexto ni construcciones complejas.

- Herramientas: VADER (afinado para redes sociales en inglés), SentiWordNet, léxicos en español como ML-SentiCon o iSOL.

Enfoque 2: aprendizaje automático supervisado

Se entrena un clasificador con textos etiquetados, usando TF-IDF o embeddings.

```
Datos de entrenamiento

"Me encantó el servicio"          ->  positivo
"Nunca más vuelvo a este lugar"   ->  negativo
"Cumplió con lo prometido"        ->  neutro
   ... 10,000 ejemplos más ...
```

- Ventajas: se adapta al dominio y al vocabulario propio.
- Desventajas: necesita datos etiquetados, que son caros de producir.

Enfoque 3: modelos preentrenados y transformers

Se ajusta un modelo de lenguaje ya entrenado sobre un conjunto de sentimientos, o se usa uno ya afinado.

- Ventajas: mejor desempeño, entiende contexto, negación y construcciones largas.
- Desventajas: costo computacional y menor interpretabilidad.
- Para español existen modelos afinados sobre corpus de opinión de la región.

El reto de la negación

```
"El hotel está mal"            ->  negativo

"El hotel no está mal"         ->  positivo
                                   (la palabra "mal" sigue ahí)
```

```
"No puedo decir que no me
 haya gustado"                 ->  positivo, con doble negación
```

Un léxico simple falla en los tres casos. Aquí se ve por qué no conviene borrar las negaciones al filtrar palabras vacías.

El reto de la ironía

```
"Excelente, otra vez se cayó el sistema"
        ^ palabra positiva, mensaje negativo

"Qué maravilla esperar dos horas por un café frío"

"Me fascina que cancelen el vuelo sin avisar"
```

La ironía depende de conocimiento del mundo: hay que saber que un sistema caído es malo. Ningún conteo de palabras lo resuelve.

El reto del dominio

```
"impredecible"    en un auto     ->  negativo
                  en una novela  ->  positivo

"pequeño"         en un cuarto   ->  negativo
                  en un celular  ->  positivo

"largo"           en una espera  ->  negativo
                  en una batería ->  positivo
```

Un léxico o un modelo entrenado en otro dominio se traslada mal.

Otros retos

```
Comparación   "mejor que el anterior"
              no dice si el producto es bueno

Jerga local   "está de a huevo" · "qué chilero" · "me cae gordo"

Emojis        "Llegó el pedido □□  vs  "Llegó el pedido □□
              el texto es idéntico, la opinión es opuesta
```

Recomendación práctica

Un léxico sirve para explorar rápido. Para producción, entrena o ajusta un modelo con datos del dominio real, porque el vocabulario de opinión cambia radicalmente entre industrias.

Evaluación

Clasificación

| Métrica | Qué mide | |---|---| | Exactitud | Proporción de aciertos. Engañosa con clases desbalanceadas | | Precisión | De lo que predije positivo, cuánto lo era | | Recall | De lo positivo real, cuánto encontré | | F1 | Media armónica de precisión y recall | | Macro-F1 | Promedio de F1 por clase; trata igual a las clases raras |

El problema del desbalance

```
10,000 correos:  9,700 legítimos  ·  300 spam

Modelo tramposo: "todo es legítimo"
  Exactitud:  97%     parece excelente
  Recall spam: 0%     no detectó un solo spam
```

En texto el desbalance es la norma: el spam es minoritario, las quejas graves son minoritarias, las noticias falsas son minoritarias. Reportar solo exactitud oculta que el modelo falla justo en lo que importa. Usa F1, macro-F1 y la matriz de confusión.

Validación

- Partición estratificada para conservar la proporción de clases.
- Validación cruzada cuando hay pocos datos.
- Si el texto tiene componente temporal (noticias, tickets), la partición debe respetar el tiempo.
- El vocabulario se construye solo con el conjunto de entrenamiento, nunca con el de prueba.

Fuga de información

```
Incorrecto   construir el vocabulario con TODOS los documentos
             y después partir en entrenamiento y prueba

             -> el modelo ya vio términos de la prueba
             -> la métrica sale mejor de lo que será en producción

Correcto     partir primero, construir el vocabulario solo
             con el entrenamiento
```

Herramientas del Ecosistema

| Herramienta | Para qué sirve | |---|---| | NLTK | Enseñanza, tokenización, corpus, léxicos | | spaCy | NLP de producción: lematización, POS, entidades | | scikit-learn | Vectorización, clasificación, agrupamiento | | Gensim | Word2Vec,

LDA, similitud entre documentos | | Transformers | Modelos preentrenados y fine-tuning | | pysentimiento | Sentimientos y emociones en español |

Consideraciones Éticas

El texto es producido por personas, y eso trae responsabilidades.

- Privacidad: los textos suelen contener datos personales; anonimizar antes de analizar.
- Sesgo: los modelos aprenden los sesgos presentes en los datos de entrenamiento y los reproducen a escala.
- Consentimiento: que un texto sea público no significa que su autor autorizara su análisis.
- Transparencia: en decisiones que afectan a personas (contratación, crédito, moderación) debe poder explicarse el criterio.
- Uso indebido: vigilancia, manipulación de opinión, perfilado no consentido.

Resumen

- La minería de texto extrae patrones y conocimiento de texto no estructurado convirtiéndolo primero en datos numéricos.
- El texto es difícil por su alta dimensionalidad, dispersión, ambigüedad y dependencia del orden.
- Se distingue de la minería de datos por la entrada, de NLP por el objetivo y de la recuperación de información por lo que produce.
- El preprocesamiento decide qué información llega al modelo: limpieza, normalización, tokenización, idioma, palabras vacías y reducción de formas.
- Tokenizar es cortar el texto en unidades; no basta con separar por espacios.

- Quitar palabras vacías reduce el vocabulario, pero borrar las negaciones invierte el sentido.
- El stemming corta por reglas y la lematización usa diccionario y contexto.
- Las representaciones van de bolsa de palabras y TF-IDF hasta embeddings contextuales.
- La clasificación asigna etiquetas; la predicción abarca además valores numéricos, eventos futuros y la siguiente palabra.
- El análisis de sentimientos usa léxicos, aprendizaje supervisado o transformers, y sus mayores retos son la negación, la ironía y el dominio.
- Evalúa con F1 y macro-F1 ante clases desbalanceadas, y evita la fuga de información al construir el vocabulario.