Diplomado de Ciencia de Datos

Módulo VII S1: Minería de texto

Jorge de la Vega Góngora

Museo Interactivo de Economía

Introducción

Spotify trends could help us gauge the public mood - Bank of England

Chief economist says 'taste in books, TV and radio may also offer a window on the soul'

▲ Euphoric times? Janelle Monae celebrates the launch of her new album at a Spotify event. Photograph: Christopher Polk/Getty Images for Spotify

Central bankers seeking to understand what's really happening in the economy might want to forget about market research surveys and get hip to the number of Taylor Swift downloads instead, the chief economist at the Bank of England has suggested.

Twitter-based Risk Index

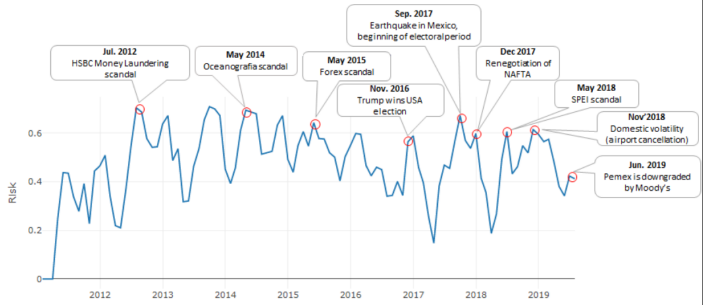

Figura: Fuente: Rho, et.al: A Sentiment-based Risk Indicator for the Mexican Financial Sector, Banco de México Working Papers, May 2021

Minería de texto I

- El texto es uno de los tipos de datos más comunes.
- Con la evolución de la era digital, la información en forma de texto ha tomado más y más relevancia.
- El objetivo de la *minería de textos*, como en el caso de la minería de datos, es extraer información útil de *fuentes de datos documentales* a través de la identificación y exploración de patrones interesantes.
- Las *fuentes de datos* son colecciones de documentos, esencialmente palabras o combinación de palabras, que usualmente no se encuentran en bases de datos sino en fuentes *no estructuradas*.
- Por ejemplo, las empresas, las bibliotecas, oficinas públicas, etc. acumulan mucha información en documentos y guardan más información en forma de texto.

Definición

La minería de textos o analítica de textos, es un proceso de conocimiento intensivo en el que un usuario interactúa con una colección de documentos a lo largo del tiempo usando un conjunto de herramientas analíticas. Es una especialización de la ciencia de datos.

Aplicaciones de la minería de texto I

- Algunas aplicaciones:
  - Recuperación de la información
  - Análisis de redes sociales
  - Filtrado de *spam*, detección de noticias falsas
  - Análisis de sentimiento
  - Procesamiento de lenguaje Natural
- Algunas aplicaciones en específico:
  - **Servicio al cliente**: puede ayudar a la identificación de publicaciones en los medios sociales viables para una organización.
  - **Recursos humanos**: percepciones de candidatos sobre la organización o para ligar descripciones de trabajo con CVs.
  - **Empresas privadas**: como una herramienta más de ciencia de datos, puede dar elementos importantes para la modelación predictiva.
  - **Inteligencia**: revisión de textos externos para proveer recomendaciones profundas a una organización.
  - **Política y economía**: análisis de sentimientos, tendencias de mercado (*bearish* o *bullish*). Análisis de diversidad léxica (eg. Lula vs Bolzonaro en discursos)
  - **Finanzas**: Construir índices basados en las noticias de datos (índices de pánico, de incertidumbre)

Aplicaciones de la minería de texto II

- **Encuestas**: Para medir la prominencia de una campaña.
- Para identificar a los evangelistas de una marca e impactar la modelación de la propensión de un cliente.
- Dar información relevante de las encuestas de satisfacción de los clientes.
- Analizar la reacción de los consumidores a los cambios de tendencia de los mercados de un producto.
- Analizar el sentimiento que se transfiere en los reportes de las empresas.

Proceso de Minería de Textos

Pasos de la Minería de Textos

1. Extracción de los datos para el análisis de un *corpus* de datos no estructurados.
2. Preparación de los datos crudos para un análisis cuantitativo. Estandarización del texto, vectorización.
3. Análisis exploratorio. Análisis de la estructura de los datos, identificación de patrones (como temas comunes).
4. Especificación de una regla para clasificar el sentimiento de los documentos en las bases de datos y aplicación de un *clasificador*: algoritmo que aplique de manera automática la regla de clasificación.
5. Construcción de indicadores basados en los *inputs* del paso de clasificación.
6. Probar indicadores con respecto a indicadores alternativos.

Herramientas en R

Los siguientes paquetes son útiles para desarrollar *textmining* en R:

- `pdftools`: Permite extraer el texto disponible en archivos pdf y algunos metadatos relevantes del archivo. También permite hacer algunas conversiones de archivos en pdf. En Mac necesita `brew poppler`.
- `corpus`: Permite crear objetos de tipo *corpus*, que son las conexiones de documentos, y permite hacer algunas operaciones relacionadas con el corpus, como tokenizar, filtrar y calcuar algunas estadísticas.
- `wordcloud` y `wordcloud2`: Paquetes para crear nubes de palabras, tanto básicas como con formatos diferentes.
- `tidytext`: Version de tipo tidy para el análisis de textos. Creado por Julia Silge.
- `tm`: Paquete que incluye varias de las funcionalidades para el análisis de textos en un sólo paquete, que permite importar datos, manejo de corpus, preprocesamiento, gestión de metadatos y creación de matrices de términos de un documento.

Estructuras de datos

- Las fuentes de documentos pueden ser diversas:
  - archivos de texto
  - páginas web
  - hojas de cálculo
  - archivos pdf
  - etc.

  No necesariamente tienen que ser del mismo tipo y usualmente son datos no estructurados.
- Para darle estructura al texto, usualmente se considera la siguiente jerarquía de estructuras sobre las que se organizan los datos:
  - La estructura de datos fundamental en el análisis de texto es un objeto `TextDocument`.
  - La estructura que agrupa un conjunto de documentos de texto se conoce como objeto de tipo `Corpus`.
  - La siguiente estructura es el objeto `TextRepository` que es una colección de objetos Corpus.
- Una vez que se tiene una de estas estructuras, el siguiente paso es procesar los datos para simplificarlos y normalizarlos.

Procesamiento de datos

- Las transformaciones más importantes son las siguientes:
  - Eliminar símbolos de puntuación
  - Eliminar números
  - Quitar espacios innecesarios
  - Convertir todas las palabras a minúsculas (para no distinguir entre Casa y casa, por ejemplo).
  - Quitar algunas palabras que no son relevantes para el análisis de texto. Esto incluye, por ejemplo, artículos, preposiciones, conectivos, etc. A este tipo de palabras se le conoce como *stopwords*. También hay este tipo de palabras que son de dominio específico.
- En algunos contextos, es conveniente quedarse sólo con la raíz de las palabras, para no distinguir entre plural o singular, o variaciones de una raíz. Por ejemplo: "financiero", "financieros", "financiando", "finanzas", tienen la misma raíz "finan". Este tipo de operación se le llama derivación de raíz, o enraizamiento (`stemming` en inglés).

Procesamiento de datos I

- Una vez que se tienen los datos 'limpios', se elabora la **bolsa de palabras** o **matriz de términos-documentos**, como se le denomina a la versión de datos estructurados.
- Esta matriz es la representación de los datos donde el texto se almacena con una forma de estructura, estableciendo un puente entre los datos no estructurados y los datos estructurados.

```
library(tm)
a <- c("Este es un pequeño ejemplo para construir una estructura de texto", "Sólo contiene tres oraciones", "pequeño ejemplo")
cuerpo <- Corpus(VectorSource(a))
M <- TermDocumentMatrix(cuerpo)
inspect(M)
```

```
<<TermDocumentMatrix (terms: 12, documents: 3)>>
Non-/sparse entries: 14/22
Sparsity           : 61%
Maximal term length: 10
Weighting          : term frequency (tf)
Sample             :
            Docs
Terms        1 2 3
  construir  1 0 0
  contiene   0 1 0
  ejemplo    1 0 1
  este       1 0 0
  estructura 1 0 0
  oraciones  0 1 0
  para       1 0 0
  pequeño    1 0 1
  texto      1 0 0
  una        1 0 0
```

La forma de leer la salida de esta inspección es la siguiente:

- El conjunto de datos tiene 12 términos (palabras) que aparecen al menos una vez en tres documentos (oraciones en este caso).
- Hay 22 celdas en la matriz de frecuencias, de las cuales 12 son no ceros.
- La esparcidad muestra el porcentaje de celdas que son 0 respecto al total de celdas (22/(22+14)=0.61)
- La longitud maximal de término, se refiere a el mayor número de caracteres de uno de los términos en la dt, en este caso, la palabra 'estructura', que tiene 10 caracteres.

- La entrada de la matriz es alguna función de ponderación del término en los documentos. Esto se hace para poder reducir los términos que tienen poco peso, ya que usualmente los textos pueden ser altamente dimensionales. Algunos ejemplos de ponderaciones de los términos son:
  - la frecuencia de los términos en cada documento

Procesamiento de datos III

- la frecuencia del documento por la frecuencia inversa del documento: $TF - IDF = f \cdot \log(N/d)$, donde N es el número de documentos, d es el número de documentos en donde el término aparece, y f es su frecuencia.

```
    M2 <- TermDocumentMatrix(cuerpo, control = list(weighting = weightTfIdf))
    inspect(M2)
```

```
<<TermDocumentMatrix (terms: 12, documents: 3)>>
Non-/sparse entries: 14/22
Sparsity           : 61%
Maximal term length: 10
Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
Sample             :
             Docs
Terms               1         2         3
  construir  0.19812031 0.0000000 0.0000000
  contiene   0.00000000 0.3962406 0.0000000
  ejemplo    0.07312031 0.0000000 0.2924813
  este       0.19812031 0.0000000 0.0000000
  estructura 0.19812031 0.0000000 0.0000000
  oraciones  0.00000000 0.3962406 0.0000000
  para       0.19812031 0.0000000 0.0000000
  pequeño    0.07312031 0.0000000 0.2924813
  sólo       0.00000000 0.3962406 0.0000000
  tres       0.00000000 0.3962406 0.0000000
```

Procesamiento de datos IV

- función de peso binaria: sólo indica si el documento aparece en un documento, ignorando si aparece en otros y su frecuencia.

```
M2 <- TermDocumentMatrix(cuerpo, control = list(weighting = weightBin))
inspect(M2)
```

```
<<TermDocumentMatrix (terms: 12, documents: 3)>>
Non-/sparse entries: 14/22
Sparsity           : 61%
Maximal term length: 10
Weighting          : binary (bin)
Sample             :
            Docs
Terms        1 2 3
  construir  1 0 0
  contiene   0 1 0
  ejemplo    1 0 1
  este       1 0 0
  estructura 1 0 0
  oraciones  0 1 0
  para       1 0 0
  pequeño    1 0 1
  texto      1 0 0
  una        1 0 0
```