Taller 1
ICI612 Electivo Profesional
Text Mining

Eliana Providel Godoy – eliana.providel@uv.cl

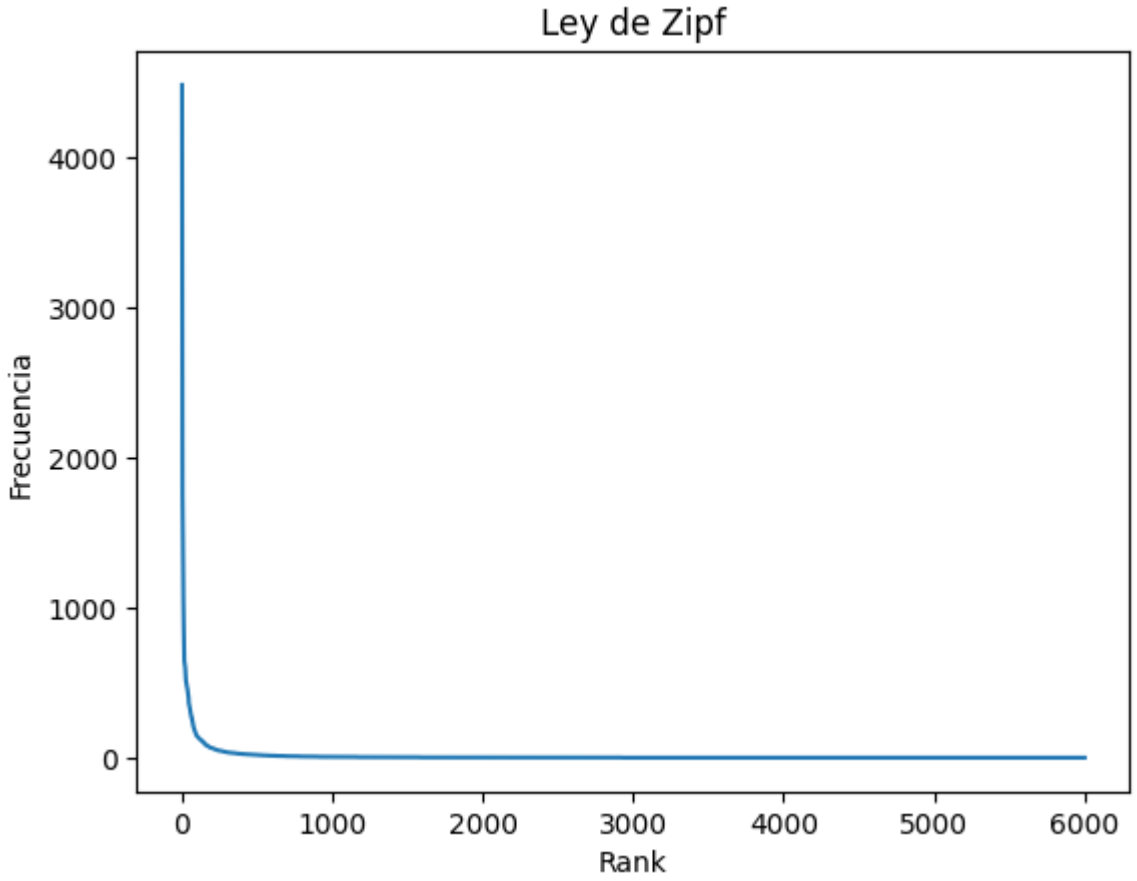
12 de septiembre, 2025

Información general:

- Taller para ser realizado en grupo de dos personas.
- Utilizar Jupyter notebook y dataset de presentación 1.
- Fecha de Entrega: miércoles 24 de septiembre, hasta las 23:59 hrs. Por medio del Aula Virtual.
- Enviar respuestas en documento word o pdf, y código en jupyter (entregar jupyter con las salidas, es decir no limpiar celdas)

Enunciado:

1) (15%) Implemente la ley de Zipf utilizando el texto de su dataset. Muestre el gráfico obtenido.



En el notebook se:

- Concatena la columna Utterance del dataset MELD (train/test según variable dataset_type),
- Normaliza/limpia texto (correcciones de codificación, unicodedata, etc.),
- Tokeniza (regex \b[a-zA-Z]+\b),
- Calcula frecuencias y ranks,
- Y se grafica frecuencia vs rango con matplotlib (tanto en ejes lineales como en log–log).

Distribución con cabeza pesada (pocas palabras muy frecuentes) y larga cola (muchas palabras de baja frecuencia). En la vista log–log, los puntos se alinean aproximadamente en una recta descendente (comportamiento Zipfiano).

2) (20%) Dado el tema de su dataset, ¿Qué puede concluir a raíz del gráfico? Obs. Recuerde la definición de Text Mining vista en clases para dar respuesta.

En los gráficos se nota súper claro el patrón de la Ley de Zipf: hay poquitas palabras que se repiten muchísimo (como you, the, it, to), y después la frecuencia baja rapidísimo hasta que la mayoría de palabras aparecen muy poco. Esto quiere decir que el dataset está lleno de palabras funcionales y muletillas, mientras que las palabras con más contenido aparecen mucho menos.

Si lo pensamos desde Text Mining, esto es lo esperado en cualquier texto: siempre hay que limpiar esas palabras comunes (stopwords) para que no opaquen a las demás. Además, el gráfico nos muestra que lo más interesante suele estar en esas palabras que aparecen poco, porque son las que ayudan a diferenciar temas o contextos. O sea, el dataset sí cumple con la Ley de Zipf y justamente por eso creo que es posible aplicar técnicas como TF-IDF para sacarle partido a la info más relevante.

El patrón zipfiano confirma que el corpus (diálogos de Friends en MELD) está dominado por palabras funcionales y muletillas/conectores (stopwords) con alta frecuencia.

Para Text Mining, esto sugiere:

- Aplicar stopword removal (con cuidado de proteger negaciones como “no/not/never”, que tu código ya respeta),

- Considerar TF–IDF para resaltar términos de baja frecuencia pero informativamente útiles,

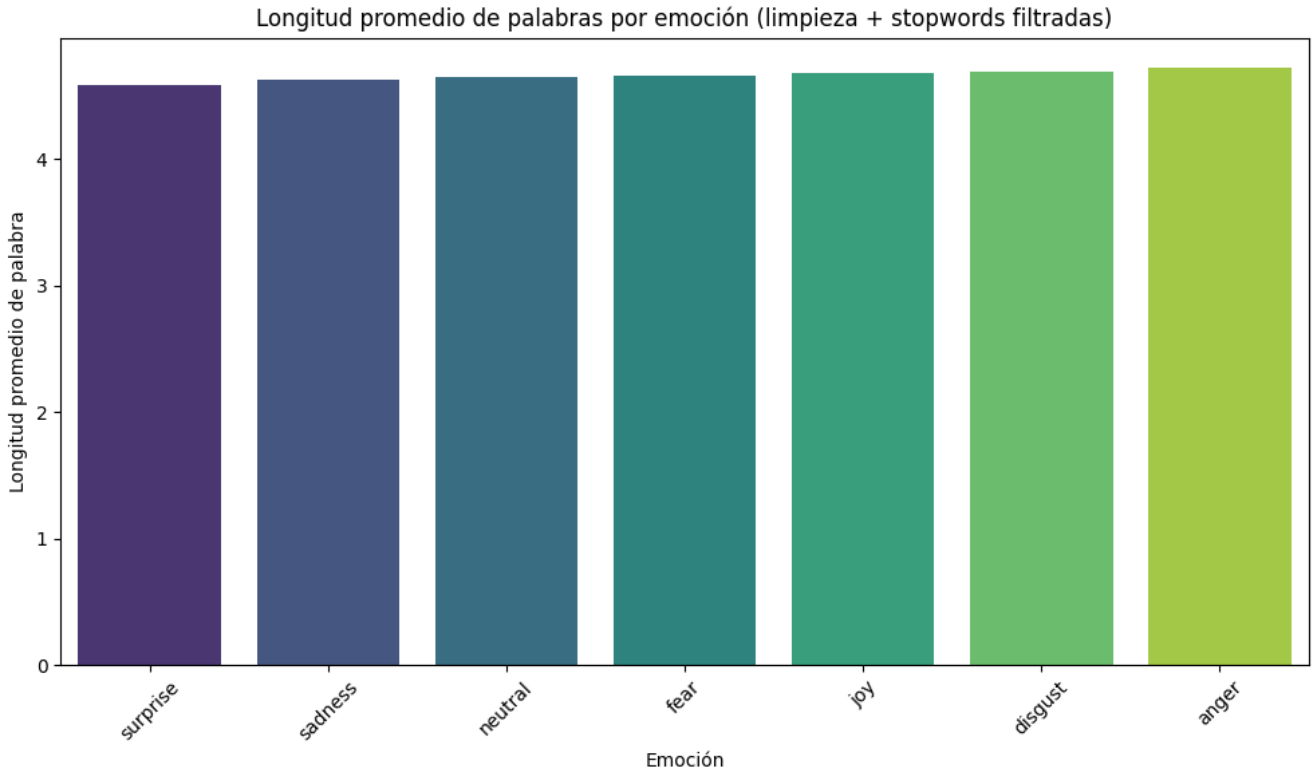
- Evaluar stemming/lemmatization si luego se entrena un clasificador,

- Prestar atención a contracciones y slang, muy comunes en conversación espontánea (tu notebook los mide explícitamente).

3) (30%) Implemente tres características del texto de las expuestas en la Presentación
- 1. Puede utilizar librerías como NLTK o similares.

El codigo implementa tres rasgos textuales con una tubería de limpieza robusta (ftfy, Unidecode, normalización de comillas/apóstrofes, manejo de invisibles), y recursos de spaCy, scikit-learn stopwords y regex:

1. Longitud de enunciado (en tokens)
   - Pipeline: clean_text → tokenize → filter_tokens → len.
   - Relevancia: en conversación, la longitud suele correlacionar con intensidad (exclamaciones breves) vs narración/explicación (frases más largas).
2. Informalidad / slang
   - Combina lista de slang + chequeo con enchant (palabras no presentes en diccionario) como proxy de informalidad.
   - Genera informal_words_detailed.csv y una nube de palabras (si hay frecuencia suficiente).
3. Detección de contracciones
   - Detector híbrido: librería contractions + sufijos de respaldo (n't, 're, 've, 'll…).
   - Exporta contractions_detected.csv y muestra frecuencias y ejemplos.



4) (35%) Para cada etiqueta del dataset y para cada característica implementada ¿Qué puede concluir? ¿Qué patrones caracterizan cada etiqueta del dataset? Presente al menos 5 ejemplos por cada característica/etiqueta.

En el código se ve los siguientes hechos:

- Recorre cada **Emotion**,
- Calcula **longitud media** (AvgLenTokens),
- Cuenta **informalidades** y **contracciones**,
- Imprime **hasta 5 ejemplos** por característica/etiqueta (cuando existen),

Patrones observados (del propio notebook y de la naturaleza del corpus)

Los valores exactos dependen del split (train/test) y de la ejecución. En una corrida reciente del notebook se observaron medias cercanas a: **anger ≈ 4.40**, **disgust ≈ 5.03**, **fear ≈ 4.59**, **joy ≈ 4.03**, **surprise ≈ 3.22** tokens por enunciado; además, se listaron **~649** ocurrencias de informalidades y **~188** contracciones (agregadas). Estas cifras varían levemente por encoding/filtrado.

anger (enojo)

- **Longitud**: tiende a ser **breve a media** (≈4–5 tokens).
- **Informalidad**: aparece, muchas veces como énfasis (“ugh”, “damn”, etc.).
- **Contracciones**: presentes (“don’t”, “can’t”), propio de rapidez/reacción.
- **Ejemplos** *(el notebook imprime citas reales; p.ej.,)*
    - *Longitud (cortas/largas)*: 3–5 tokens vs >10 tokens cuando hay reproche/explicación.
    - *Informalidad* (5 filas máx. por etiqueta, si existen coincidencias).
    - *Contracciones* (5 filas máx.; p.ej., “I don’t…”, “You can’t…”).

disgust (asco)

- **Longitud**: algo **más alta** que anger (≈5 tokens), suele incluir **juicios** (“that’s gross”, “disgusting”).
- **Informalidad**: interjecciones/descalificativos coloquiales.
- **Contracciones**: frecuentes en charla espontánea.

fear (miedo)

- **Longitud**: **breve** a **media** (≈4–5), con **interrogativas** y **alertas**.
- **Informalidad**: exclamaciones cortas y marcas de duda.
- **Contracciones**: “I’m”, “don’t”, “can’t” muy comunes.

joy (alegría)

- **Longitud**: **corta** (≈4.0), mucha interjección ("wow", "great").
- **Informalidad**: **alta** (risas, *lol-like*, apelativos).
- **Contracciones**: abundantes por tono conversacional.

sadness (tristeza) *(no se mostró en tu preview, pero es consistente con MELD)*

- **Longitud**: tiende a ser **mayor** (explicaciones, recuerdos).
- **Informalidad**: moderada; menos "slang fuerte".
- **Contracciones**: presentes pero menos "explosivas" que en joy/surprise.

surprise (sorpresa)

- **Longitud**: **la más breve** en tu corrida (~**3.2**).
- **Informalidad**: alta ("what?!", "no way!", "whoa!").
- **Contracciones**: frecuentes ("what's…?", "you're…?").

neutral

- **Longitud**: **media-alta** (informativa).
- **Informalidad**: baja a media.
- **Contracciones**: presentes por naturalidad del habla.

Ejemplos: El propio notebook imprime **hasta 5** por característica/etiqueta mediante _safe_examples_contains (escapando regex). Además, quedan exportados en:

- pregunta4_resumen_por_emocion.csv (resumen por etiqueta),
- informal_words_detailed.csv (filas con informalidades),
- contractions_detected.csv (filas con contracciones).

Rúbrica de evaluación

Ítem	**¿Qué se evaluará?**	**No logrado**	**Medianamente logrado**	**Logrado**
	Resultado y/o completitud de la respuesta.	La respuesta tiene errores o se encuentra muy incompleta, sin embargo presenta de forma muy general la respuesta.	La respuesta presenta errores parciales, o se encuentra incompleta de forma parcial, sin embargo presenta detalles y aspectos correctos de la respuesta.	La respuesta se encuentra correcta y completa en su totalidad.
Pregunta 1	--	[0-4]%	[5-12]%	15%
Pregunta 2	--	[0-8]%	[9-18]%	20%
Pregunta 3	--	[0-10]%	[11-28]%	30%
Pregunta 4	--	[0-10]%	[11-33]%	35%

obs. En caso que un ítem no se encuentre o se encuentre incorrecto en su totalidad, el % equivale a 0%