Protocolo para el armado de dataset para finetunings de LLM

por Karen Palacio

email: karen.palacio.1994@gmail.com | | tel: +549 3513374932 | | github: karen-pal (Karen Palacio) · GitHub | | instagram: Karen Palacio (🌞) (@kardaver) • Instagram photos and videos | | youtube: Karen Palacio - YouTube | | personal page: https://karen-pal.github.io/about/ | | CV: CV | | soundcloud: https://soundcloud.com/kardaver

Introducción

Este documento está destinado a servir como guía para el armado de un dataset de texto de manera colaborativa a su grupo.
Esta guía es necesaria porque para poder correr procesos automáticos-programáticos de manera eficiente necesitamos que los datos que creemos por separado sigan el mismo formato.
Para que ustedes se imaginen - el texto no es una cuestión estandarizada de sistema operativo a sistema operativo. Por ej existen carácteres de texto que solo existen en Windows, otros que solo existen en mac, etc. Incluso cada editor de texto ingresa sus propios carácteres de formateo que el resto de los editores no reconocen. De hecho existen carácteres invisibles, famosos en el ecosistema Windows por entorpecer procesos de análisis de texto. Ninguno de sus editores de texto van a mostrarle a ustedes estos carácteres molestos, pero sí me van a saltar a mí cuando programe los procesos para el entrenamiento posterior. Queremos que eso pase lo mínimo posible - ya que sanitizar texto es un proceso que conlleva mucho tiempo.

Metodología

Notas previas

Es importante que sigan estas indicaciones. Si no las siguen, probablemente a su momento yo me de cuenta. Si el resultado que me entregan está suficientemente sucio como llevarme una semana limpiarlo, se los voy a devolver y van a tener que hacer la entrada de datos de nuevo desde cero. Lo evitemos!

Descripción

Para el anotado de datos en Windows **usaremos el programa bloc de notas**. Esto es: no usaremos editores de texto avanzados - como notepad++, vscode, o *mucho menos* word,

excel, google docs, google spreadsheets, etc. Cada uno de esos editores de texto aumenta la probabilidad de ensuciar el dataset.

Si usan Mac/Apple no tienen bloc de notas, entonces usen **TextEdit**. Esto es: no usaremos editores de texto avanzados - como notepad++, vscode, o *mucho menos* word, excel, google docs, google spreadsheets, etc. Cada uno de esos editores de texto aumenta la probabilidad de ensuciar el dataset.
Van a tener que configurarlo para que esté en texto plano, ya que por defecto usa un tipo de texto que ensuciaría el dataset. Para usarlo en modo de **texto plano** hagan esto:

- Abrir **TextEdit**.
- Ir a **Formato** en la barra de menú.
- Seleccionar **Convertir a texto sin formato**.
- mientras estemos haciendo esto les recomiendo configurar el programa para que siempre abra archivos como texto plano desde **Preferencias**.

*> Si esto se les complica pueden instalar el editor **gedit** que tiene la configuración que necesitamos por defecto, y no tienen que configurar nada.*

Si usan Linux usen **gedit**. No hace falta configurar más nada. Esto es: no usaremos editores de texto avanzados - como notepad++, vscode, o *mucho menos* word, excel, google docs, google spreadsheets, etc. Cada uno de esos editores de texto aumenta la probabilidad de ensuciar el dataset.

La forma de llevar a cabo la escritura es **escribiendo directamente**. **No copien y peguen** de otra fuente, ya que en ese proceso pueden estar ensuciando el dataset y no lo van a detectar ustedes (yo sí).

Vamos a estar usando **el tipo de archivo csv**. Un archivo csv es un archivo con datos separados por comas. Si les sirve lo pueden pensar como tablas con columnas y filas. Cada dato separado por coma corresponde a una columna, cada línea del archivo es una fila. Les voy a dar ejemplos de cómo se ve eso.

Cada integrante escribirá sus datos en un solo archivo que deben llamar

datos_nombre_integrante.csv

Si se les vuelve engorroso escribir todo en un solo archivo por alguna razón, pueden hacer varios archivos, por ej

datos_nombre_integrante_1.csv
datos_nombre_integrante_2.csv
… etc

Solo no se olviden luego de pasarme todos los archivos!

No agreguen caracteres especiales (tildes, virgulilla, etc) al nombre del archivo. No agreguen espacios en blanco en el nombre del archivo. **Mantengan el nombre del archivo lo más llano posible, usando guion bajo en vez de espacios.**

En este archivo vamos a escribir datos en tres columnas separadas por comas: instruction, input, output.

donde
instruction: refiere a la tarea que quieran que el modelo realice. Usualmente es un verbo. Puede ser directamente una pregunta también.
input: El contenido de entrada (si aplica).
output: La respuesta esperada.

Los datos dentro de cada columna siempre tienen que estar entre comillas dobles.
Por ejemplo

datos_karen_1.csv
```
"instruction","input","output"
"Traducir al inglés el siguiente texto","Hola, ¿cómo estás?","Hello, how are you?"
```

Siempre vamos a agregar esa linea de `"instruction","input","output"` arriba de todo, una sola vez por archivo. Yo la voy a necesitar para leer el archivo con código, y a ustedes les puede servir como guía para no olvidarse qué va en cada columna y por ej no ponerlas en otro orden.

Si lo quieren ver como una tabla sería:

instruction	input	output
`Traducir al inglés el siguiente texto`	`Hola, ¿cómo estás?`	`Hello, how are you?`

Podemos ver lo importante que son las comillas dobles para poder entender cómo está delimitado el dato cuando usamos comas en nuestros textos del dataset pero también las usamos como separadores.

A veces por el tipo de instrucción que usamos no necesitamos input, en ese caso introducimos dos comillas dobles, especificando que es una oración vacía.
Es decir: `""` **= oración vacía**.

datos_karen_1.csv
```
"instruction","input","output"
"Traducir al inglés el siguiente texto","Hola, ¿cómo estás?","Hello, how are you?"
"cuál país es el campeón del mundo de futbol","","Argentina"
```

Si lo representáramos como tabla:

instruction	input	output
`Traducir al inglés el siguiente texto`	`Hola, ¿cómo estás?`	`Hello, how are you?`

cuál país es el campeón del mundo de futbol?		Argentina

Como vemos, cuando queremos agregar un dato más usamos la tecla enter para indicar que lo que sigue es un nuevo dato.

Unos ejemplos para que se imaginen el tipo de instrucciones que ustedes pueden indicar:

datos_karen_2.csv

```
"instruction","input","output"
"Expandir el siguiente sueño de manera creativa","Me senté a cenar y mi gato me hablaba.","Cuando le pregunté si me quería me dijo que yo era lo mejor que le pasó en su vida"
```

datos_karen_3.csv

```
"instruction","input","output"
"¿Qué soñaste hoy?","","Soñé que me senté a cenar y mi gato me hablaba. Cuando le pregunté si me quería me dijo que yo era lo mejor que le pasó en su vida"
```

Queda en ustedes juntarse y definir el giro poético que quieran darle a este dataset: si se ponen de acuerdo de usar los mismos verbos/instrucciones, o si cada une sigue su criterio.

Respecto a cantidad, sería óptimo tener **al menos 20 datos por cada une. No hay un máximo**, no se preocupen por eso - la única forma de generar diferencia es si ustedes aumentan órdenes de magnitud o sea múltiplos de 10 - por ej si en vez de 100 datos me mandan 1k, 10k, etc - ahí sí va a haber un cambio en los tiempos de cómputo.

A medida que los tengan me van avisando así hacemos chequeo de formato. Pueden también escribir uno cada une y mandarme el archivo para asegurarnos que lo están haciendo bien. **Me los mandan por mail a karen.palacio.1994@gmail.com .**
No hay un máximo a menos que estén pensando usar un método automático de crear este dataset - en ese caso no hagamos más de 10k datos entre todes y no me manden un resultado final de muchos datos sin antes chequear temas de formato - hagan 20 datos por ej, me lo mandan a ver si estamos siguiendo bien el formato y luego sí generen.

Contacto

Si tienen dudas en alguno de estos puntos mandarme por mail a karen.palacio.1994@gmail.com .