

IRWA Project: Part 4 - User Interface & Web Analytics

Git URL: https://github.com/raquel-sb/IRWA_2023.git

1. Introducción

Este laboratorio constituye la finalización del proyecto en el que nos hemos enfocado durante todo el trimestre. Partimos de los avances de la primera, segunda y tercera práctica, es decir, el pre-procesamiento de nuestros tweets por un lado, la indexación-evaluación de los documentos y el sistema de búsqueda resultante por el otro, y, finalmente, la creación de un modelo de ranking. En esta cuarta parte nos centraremos en la creación de una interfaz gráfica para nuestro buscador, así como, en la aplicación de algunas analíticas web en esta misma.

Así pues, nuestra meta es, proporcionar una Aplicación Web (con Flask) para introducir una search query, mostrando los resultados de la búsqueda, así como, sus estadísticas de uso.

2. Implementaciones pasadas

En cuanto a las implementaciones realizadas para esta última parte del proyecto, hemos utilizado funciones ya implementadas en las anteriores partes del proyecto. Las funciones implementadas son las siguientes:

Fichero load_corpus.py

- **clean()**: esta función toma una cadena de texto como entrada y devuelve una versión limpia de la misma. Transforma el texto a minúsculas, elimina URL, hashtags y caracteres no alfanuméricos (incluidos guiones bajos), lo que da como resultado una cadena con solo caracteres alfanuméricos y espacios en minúsculas.
- **build_terms()**: esta función limpia, elimina stopwords en inglés y deriva palabras en el texto de entrada utilizando PorterStemmer de NLTK, devolviendo una lista de términos procesados.
- **separate_by_words()**: esta función toma una cadena de entrada como argumento. Utiliza una expresión regular para encontrar todas las palabras que comienzan con una letra mayúscula y van seguidas de letras minúsculas, luego une estas palabras con espacios. Si no se encuentran tales palabras, devuelve la cadena de entrada original.
- **getHashtagsFromTweet()**: esta función toma un tweet como entrada y devuelve una lista de hashtags utilizados en el tweet. Para ello, itera sobre el campo "hashtags" en las "entidades" del tweet, separa cada hashtag en palabras y las agrega a la lista.

- **prepare_hashtag_for_text():** esta función toma una lista de hashtags como entrada. Concatena todos los hashtags en una sola cadena, la convierte a minúsculas y luego divide esta cadena en palabras individuales que se devuelven como una lista.

Fichero algorithms.py: → esta sección ha sido optimizada respecto a nuestras implementaciones y entregas anteriores.

- **create_index_tfidf():** esta función toma un diccionario de documentos como entrada y devuelve cuatro diccionarios: index, tf, df e idf. Calcula los valores de frecuencia de documento inversa de término (TF-IDF) para cada término en cada documento, que es una técnica común en la recuperación de información para medir la importancia de un término en un documento dentro de un corpus. La función también realiza un seguimiento de las posiciones de cada término en los documentos.
- **rank_documents():** esta función clasifica una lista de documentos según su relevancia para una consulta determinada. Calcula las ponderaciones TF-IDF para cada término de la consulta y cada documento, y luego califica cada documento según su similitud de coseno con el vector de consulta. Luego, los documentos se clasifican en orden descendente de puntuación.
- **search_tf_idf():** esta función toma una consulta y tres parámetros relacionados con un corpus de documentos (index, idf, tf) y devuelve una lista clasificada de documentos del corpus que son relevantes para la consulta. Primero identifica los documentos que contienen términos de consulta y luego clasifica estos documentos según sus puntuaciones de frecuencia de términos-frecuencia de documentos inversa (TF-IDF).

3. Implementaciones nuevas

Por otro lado, como novedad, hemos implementado:

Fichero objects.py:

- **Class Document:** esta clase representa un tweet con atributos como id, título, tweet, tweet_preprocesado, nombre de usuario, fecha, hashtags, me gusta, retweets y URL. Incluye métodos para convertir el objeto a formato JSON (to_json) y representar el objeto como una cadena JSON cuando se imprime (__str__).
- **Class StatsDocument:** esta clase es una estructura de datos para almacenar información relacionada con el corpus, incluidos detalles como identificación, título, tweet, nombre de usuario, fecha, URL, recuento, diferencia horaria, consulta relacionada, hora de inicio de búsqueda, dirección IP, información del sistema operativo. y navegador. También incluye métodos para representar el objeto como una cadena JSON y actualizar el atributo de recuento.

- **Class ResultItem:** esta clase es un modelo para crear objetos que representan resultados individuales de una búsqueda o recopilación de datos. Cada objeto almacena información sobre el resultado, como identificación, título, tweet, nombre de usuario, fecha, me gusta, retweets, URL, search_id y clasificación.

Fichero load_corpus.py:

- **_load_corpus_as_dataframe():** esta función lee un archivo JSON de la ruta proporcionada en un DataFrame de pandas. Luego devuelve este DataFrame.
- **load_corpus():** esta función carga un conjunto de datos desde una ruta especificada en un DataFrame de pandas, luego itera sobre cada fila para crear un objeto Documento con varios atributos (como identificación, título, tweet, nombre de usuario, fecha, hashtags, me gusta, retweets y URL). Estos objetos de documento se almacenan en un diccionario _corpus con su identificación como clave. La función devuelve este diccionario.

Fichero algorithms.py:

- **search_in_corpus():** esta función toma una consulta y el corpus como entradas. Primero crea un índice TF-IDF para el corpus usando la función create_index_tfidf y luego aplica un algoritmo de clasificación usando la función search_tf_idf para calificar los documentos en el corpus según su relevancia para la consulta. La función devuelve estas puntuaciones.

Fichero search_engine.py:

- **Class SearchEngine:** esta clase es un motor de búsqueda educativo que tiene un método de búsqueda. Este método toma una consulta de búsqueda, una identificación de búsqueda y un corpus de documentos como entrada, realiza una búsqueda en el corpus basada en la consulta y devuelve una lista de objetos ResultItem, cada uno de los cuales representa un documento en el corpus que coincide con la consulta de búsqueda. junto con su puntuación de clasificación.

Fichero analytics_data.py:

- **Class AnalyticsData:** esta clase es un objeto de persistencia en memoria que contiene tablas de análisis, con un diccionario fact_clicks para contar los clics por ID de documento. También incluye un método save_query_terms que imprime el objeto y devuelve un número entero aleatorio entre 0 y 100000.
- **Class ClickedDoc:** Esta clase representa un documento en el que se ha hecho clic. Tiene atributos como doc_id, descripción, contador, diferencia de tiempo y rel_query, y métodos para convertir el objeto a formato JSON e imprimir el contenido del objeto como una cadena JSON.

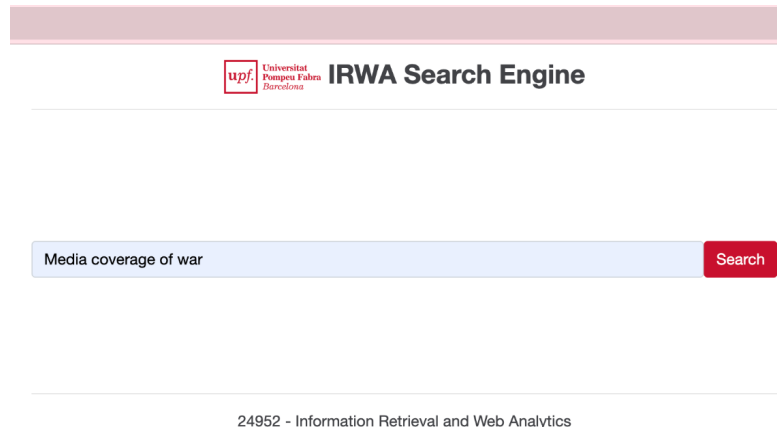
Fichero web_app.py:

- **Lines 19-53:** esta parte de código consiste en la implementación de una aplicación web Flask que utiliza un motor de búsqueda y datos analíticos. Carga un corpus de documentos desde un archivo JSON y anula el método predeterminado de la clase JSONEncoder para usar un método to_json personalizado si existe en el objeto que se está codificando.
- **Index:** se define una ruta para la URL de inicio ("/") que crea una sesión, recupera la información del navegador y la dirección IP del usuario y luego genera una plantilla HTML llamada "index.html" con un título de página "Bienvenido". Los datos de la sesión y la información del usuario también se imprimen en la consola con fines de depuración.
- **Search:** este código define una ruta/búsqueda de Flask que maneja solicitudes POST. Recupera una consulta de búsqueda de los datos del formulario, realiza una búsqueda utilizando un motor de búsqueda, almacena algunos datos en la sesión y finalmente genera una plantilla con los resultados de la búsqueda.
- **Doc_details:** este código define una ruta Flask /doc_details que responde a solicitudes HTTP GET. Recupera parámetros de consulta de la solicitud, crea un objeto ResultItem con estos parámetros, actualiza las estadísticas de clics para el documento y genera una plantilla con los detalles del documento.
- **Stats:** este código es una ruta de aplicación web Flask que calcula y muestra estadísticas sobre las interacciones del usuario con documentos en un motor de búsqueda. Realiza un seguimiento de información como el número de clics en cada documento, la diferencia horaria entre el inicio de la búsqueda y la hora actual, la consulta relacionada, la dirección IP, el sistema operativo y el navegador del usuario, y luego presenta estos datos en una plantilla.
- **Dashboard:** este código define una ruta/tablero que maneja las solicitudes GET. Recupera y clasifica documentos según la cantidad de clics que recibieron y luego muestra una página de panel con esta lista ordenada de documentos.
- **Sentiment:** este código es una aplicación web Flask con dos rutas. La primera ruta, /sentiment, muestra un formulario al usuario. La segunda ruta, también /sentiment pero que acepta solicitudes POST, toma la entrada del usuario del formulario, analiza su sentimiento utilizando la herramienta de análisis de sentimiento VADER de NLTK y luego muestra la puntuación de sentimiento al usuario en el mismo formulario.

Remarcar que, web_app.py está vinculado a diferentes ficheros html, los cuales nos han ayudado a mostrar todos nuestros resultados de forma visual y agradable para el usuario, incluso con ayuda de gráficos.

3.1. User Interface

En cuanto a la página de búsqueda, hemos creado una página web principal con un cuadro de búsqueda en el centro para que el usuario introduzca cierta consulta. Una vez introducida la consulta deberá clicar el botón de Search.



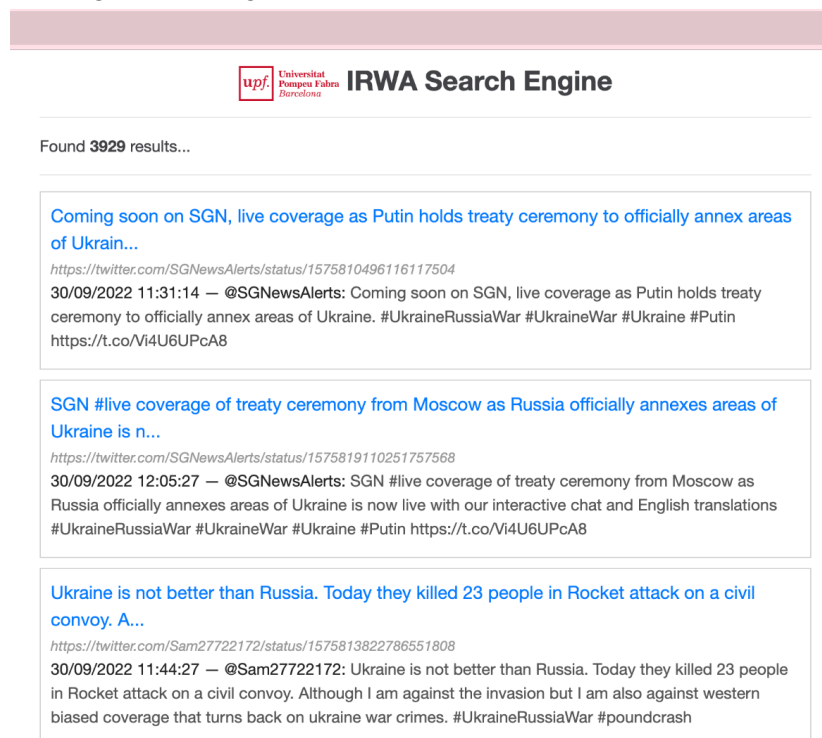
upf. Universitat Pompeu Fabra Barcelona

IRWA Search Engine

Media coverage of war Search

24952 - Information Retrieval and Web Analytics

Una vez el usuario introduce la consulta, obtiene una lista con los tweets, tal y como podemos ver en la siguiente imagen:



upf. Universitat Pompeu Fabra Barcelona

IRWA Search Engine

Found 3929 results...

[Coming soon on SGN, live coverage as Putin holds treaty ceremony to officially annex areas of Ukrain...](#)
<https://twitter.com/SGNewsAlerts/status/1575810496116117504>
30/09/2022 11:31:14 — @SGNewsAlerts: Coming soon on SGN, live coverage as Putin holds treaty ceremony to officially annex areas of Ukraine. #UkraineRussiaWar #UkraineWar #Ukraine #Putin <https://t.co/Vi4U6UPcA8>

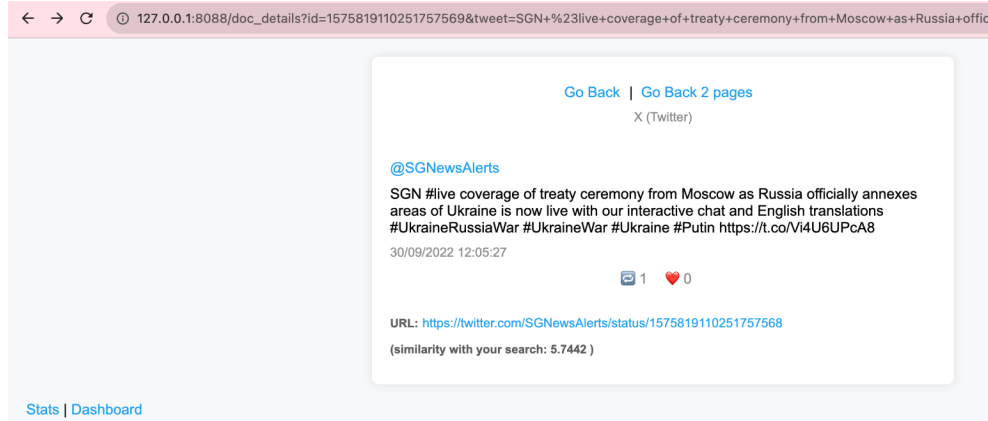
[SGN #live coverage of treaty ceremony from Moscow as Russia officially annexes areas of Ukraine is n...](#)
<https://twitter.com/SGNewsAlerts/status/1575819110251757568>
30/09/2022 12:05:27 — @SGNewsAlerts: SGN #live coverage of treaty ceremony from Moscow as Russia officially annexes areas of Ukraine is now live with our interactive chat and English translations #UkraineRussiaWar #UkraineWar #Ukraine #Putin <https://t.co/Vi4U6UPcA8>

[Ukraine is not better than Russia. Today they killed 23 people in Rocket attack on a civil convoy. A...](#)
<https://twitter.com/Sam27722172/status/1575813822786551808>
30/09/2022 11:44:27 — @Sam27722172: Ukraine is not better than Russia. Today they killed 23 people in Rocket attack on a civil convoy. Although I am against the invasion but I am also against western biased coverage that turns back on ukraine war crimes. #UkraineRussiaWar #poundcrash

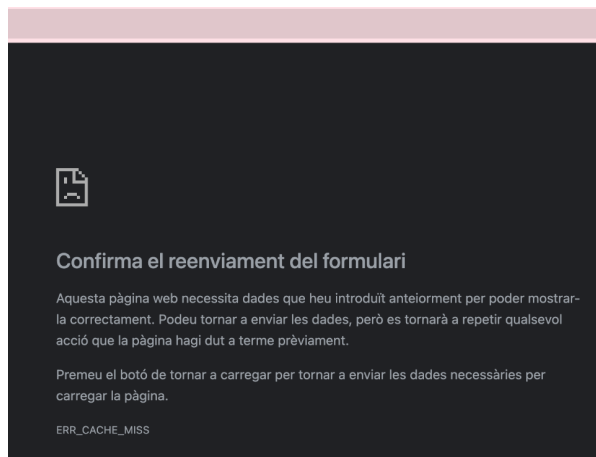
En la lista de tweets el usuario ve en primer lugar y de color azul los 100 primeros caracteres del tweet, seguidamente y en color gris el enlace, y, finalmente en color negro, fecha y hora, autor y todo el tweet.

Seguidamente, por un lado, si el usuario hace click en el enlace del tweet, este le llevará al tweet original.

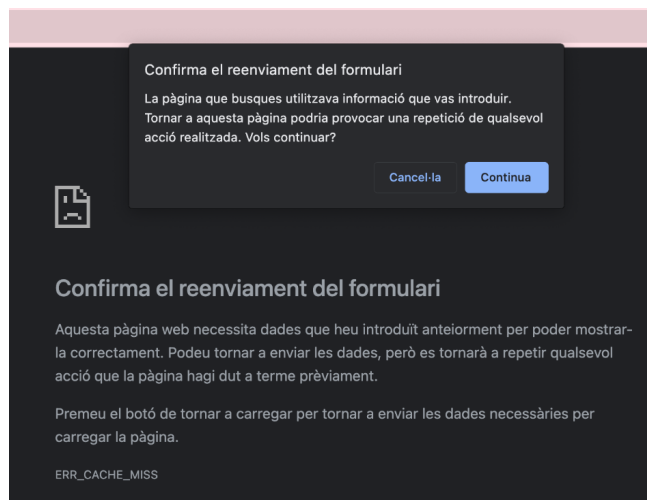
Por otro lado, si el usuario hace click en el título de un tweet de la lista se dirige a una página que podrá ver el tweet en su totalidad, así como el usuario que lo ha publicado, la fecha y hora, el enlace del tweet. También se visualiza el número de retweets que ha recibido, así como el número de likes y el score que se le ha asignado al tweet en cuestión según la relevancia que tenga relacionado con la query.



Hemos detectado en uno de los ordenadores que cuando se hace click en Go back para volver a la página de resultados de tweets, aparece la siguiente pantalla:



Pero si actualizas la página y le das a continuar, ha retrocedido correctamente y nos sitúa en la página deseada.



3.2. Web Analytics

En este apartado hemos decidido implementar las siguientes estadísticas, las cuales se pueden ver dando clic a “Stats” de forma esquemática o “Dashboard” de forma gráfica. Hay que tener en cuenta que clicks on documents y ranking of clicked documents ya estaban implementadas.

- **Clicked docs:** en esta parte vemos aquellos tweets que el usuario ha querido consultar con más detalle. Además podemos ver también el número de veces que ha visitado ese mismo tweet, el tiempo que ha tardado el usuario en hacer clic en el documento de resultado y volver a la página de resultados y, finalmente, la query a la cual está relacionada el tweet.
- **Searched queries:** aquí podemos observar las queries que ha introducido el usuario para realizar las búsquedas. Además se ve reflejado el número de palabras que contiene la query.
- **User context:** en esta parte podemos ver la siguiente información: buscador utilizado, hora y fecha del día, IP y sistema operativo.

document.search_id



IRWA Search Engine

[Go Back](#) | [Go Back 2 pages](#) | [Go Back 3 pages](#)

Quick Stats:

Clicked docs

(3 visits) — id: 1575819110251757569 — SGN #live coverage of treaty ceremony from Moscow as Russia officially annexes areas of Ukraine is now live with our interactive chat and English translations
#UkraineRussiaWar #UkraineWar #Ukraine #Putin <https://t.co/Vi4U6UPcA8>
Dwell time: 015 seconds
Related query: Media coverage of war

(2 visits) — id: 1575842840768569344 — The spokesman of the Eastern group of troops Cherevaty reported that the encirclement of the Russian group near Lyman in Donetsk region is "at the stage of completion" #UkraineRussiaWar <https://t.co/drAsg9PDes>
Dwell time: 009 seconds
Related query: Eastern separatists groups

(1 visits) — id: 1575684580295970817 — Here's how the war in Ukraine 🇺🇦 is impacting world trade and investment acc to @wef #SupplyChain #Procurement #economy #UkraineRussiaWar #foodcrisis #logistics <https://t.co/NNCRHhTFsW>
Dwell time: 026 seconds
Related query: Humanitarian impact

(1 visits) — id: 1575818037130731520 — ! Official comment on the situation in Lyman from the military 🗨️
"The operation to encircle the Russian group in Lyman is "at the stage of completion", - the representative of the Eastern group Serhiy Chereviiy. #UkraineWillWin #UkraineWar #UkraineRussiaWar #Russian <https://t.co/vuKHUpOfOf>
Dwell time: 009 seconds
Related query: Eastern separatists groups

Searched queries

Query: Media coverage of war
Number of terms: 4

Query: Eastern separatists groups
Number of terms: 3

Query: Humanitarian impact
Number of terms: 2

User context

Browser: Chrome
Time of the day: 19:14:57
Date: 2023-12-01
IP address: 127.0.0.1
OS: Mac OS X 10.15.7

Browser: Chrome
Time of the day: 19:13:15
Date: 2023-12-01
IP address: 127.0.0.1
OS: Mac OS X 10.15.7

Browser: Chrome
Time of the day: 19:08:17
Date: 2023-12-01
IP address: 127.0.0.1
OS: Mac OS X 10.15.7

Browser: Chrome
Time of the day: 19:13:15
Date: 2023-12-01
IP address: 127.0.0.1
OS: Mac OS X 10.15.7

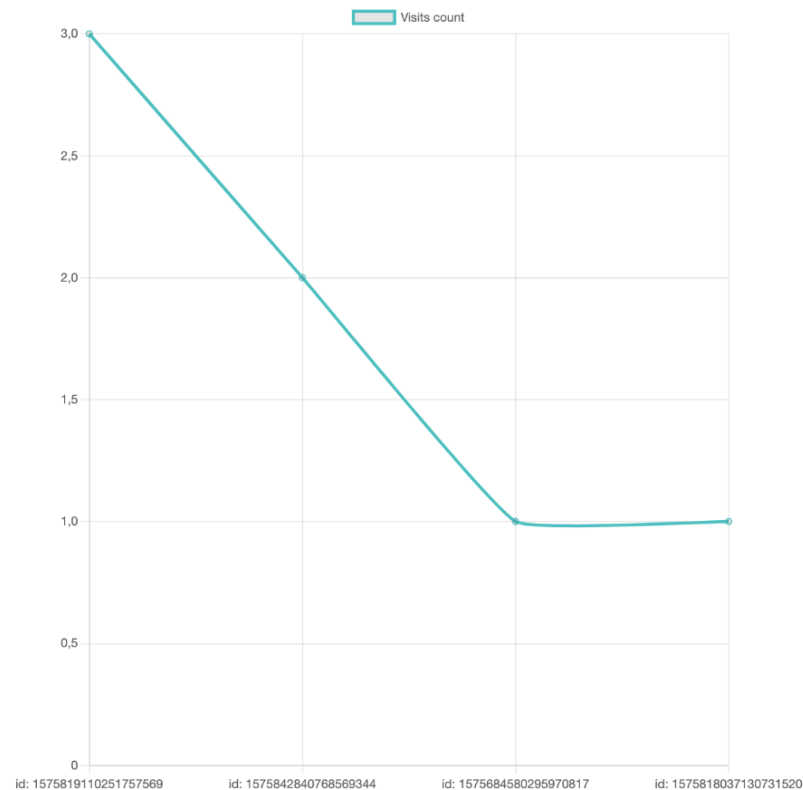
24952 - Information Retrieval and Web Analytics

- **Ranking of visited documents:** se observa un gráfico con el número de visitas que se han realizado a los diferentes tweets. En el eje x se observa los diferentes identificadores de los tweets y en el eje y el número de visitas realizadas. En la parte inferior del gráfico vemos un listado con esta misma información, pero añadiendo el contenido del tweet.



[Go Back](#) | [Go Back 2 pages](#) | [Go Back 3 pages](#)

Ranking of Visited Documents



Print Python data for verification with graph above...

(3 visits) — id: 1575819110251757569 — SGN #live coverage of treaty ceremony from Moscow as Russia officially annexes areas of Ukraine is now live with our interactive chat and English translations #UkraineRussiaWar #UkraineWar #Ukraine #Putin <https://t.co/Vi4U6UPcA8>

(2 visits) — id: 1575842840768569344 — The spokesman of the Eastern group of troops Cherevaty reported that the encirclement of the Russian group near Lyman in Donetsk region is "at the stage of completion" #UkraineRussiaWar <https://t.co/drAsg9PDDes>

(1 visits) — id: 1575684580295970817 — Here's how the war in Ukraine 🇺🇦 is impacting world trade and investment acc to @wef #SupplyChain #Procurement #economy #UkraineRussiaWar #foodcrisis #logistics <https://t.co/NNCRHhTFsW>

(1 visits) — id: 1575818037130731520 — ! Official comment on the situation in Lyman from the military 🗣️ "The operation to encircle the Russian group in Lyman is "at the stage of completion", - the representative of the Eastern group Serhiy Chereviy. #UkraineWillWin #UkraineWar #UkraineRussiaWar #Russian <https://t.co/vuKHUpOfF>

- **NLTK sentiments:** en esta parte, podemos visualizar el emoticono triste o feliz a partir de la connotación que le sugiere la query introducida. En la captura podemos ver que si introducimos war, reacciona de forma negativa con una cara triste.

127.0.0.1:8088/sentiment

NLTK Sentiments

Type a sentence, click on the submit button and wait for your prediction.

Envia

