https://doi.org/10.46744/bapl.202502.007
e-ISSN:2708-2644

Bol. Acad. peru. leng. 78. 2025 (235-260)

USO DE PYTHON EN EL ANÁLISIS DE LA DISPONIBILIDAD LÉXICA

Use of Python in the analysis of lexical availability

Emploi de Python pour l'analyse de la disponibilité lexicale

Marco Antonio Pérez Durán
Universidad Autónoma de San Luis Potosí, San Luis Potosí, México
marco.duran@uaslp.mx
https://orcid.org/0000-0003-0854-3109

Alejandro César Rico Martínez
Instituto Tecnológico Superior de San Luis Potosí, Capital, México
alejandro.rico@tecsuperiorslp.edu.mx
https://orcid.org/0000-0003-0146-7393

Alejandra Haydee Moreno Sánchez
Universidad Autónoma de San Luis Potosí, San Luis Potosí, México
alejandrhamors@gmail.com
https://orcid.org/0009-0003-4203-6124

Resumen:

El presente trabajo describe la operacionalización de la fórmula matemática de disponibilidad léxica de López Chávez y Strassburger Frías (2000) mediante el uso del lenguaje de programación Python. La metodología empleada se enmarca en un enfoque cualitativo-explicativo de carácter ejemplificativo, lo cual permite observar cómo Python puede incorporarse plenamente en los estudios de disponibilidad léxica.

 https://doi.org/10.46744/bapl.202502.007

Python optimiza de manera significativa el procesamiento de los datos, acortando el tiempo necesario y facilitando el análisis, según evidencian los resultados.

Palabras clave: disponibilidad léxica, fórmula matemática, índice de disponibilidad léxica, lenguaje de programación, Python.

Abstract:

This paper describes the operationalization of López Chávez and Strassburger Frías' (2000) mathematical formula for lexical availability using the Python programming language. The methodology used is framed within a qualitative-explanatory approach of an illustrative nature, which allows us to observe how Python can be fully incorporated into lexical availability studies. Python significantly optimizes data processing, shortening the time required and facilitating analysis, as evidenced by the results.

Key words: lexical availability, mathematical formula, lexical availability index, programming language, Python.

Résumé :

Le présent travail décrit l'opérationnalisation de la formule mathématique de disponibilité lexicale de López Chávez et Strassburger Frías (2000) au moyen du langage de programmation Python. La méthodologie adoptée s'inscrit dans une approche qualitative-explicative à caractère exemplificatif, ce qui permet d'observer comment Python peut être pleinement intégré dans les études de disponibilité lexicale. D'après nos résultats, Python optimise significativement le traitement des données, en permettant une réduction des temps et rendant plus facile l'analyse.

Mots clés : disponibilité lexicale, formule mathématique, indice de disponibilité lexicale, langage de programmation, Python.

Recibido: 26/05/2025 **Aprobado:** 03/10/2025 **Publicado:** 31/12/2025

1. Introducción

Hoy en día, la relación entre competencia léxica e inteligencia artificial (IA desde este momento) es cada vez más estrecha, especialmente en el ámbito del procesamiento del lenguaje natural (NLP desde este momento). Esta conexión se sustenta en los beneficios tanto educativos como tecnológicos que surgen de la combinación de ambos para el estudio y análisis de los niveles de la lengua. Por medio de esta conexión, se pueden generar métodos y programas basados en los resultados de la NLP que fortalezcan la competencia léxica, entendida esta como el conocimiento del vocabulario en el que se relacionan formas con significados, los cuales se agrupan y se almacenan en el lexicón mental (Lahuerta y Pujol, 1996, citado en Gómez Molina, 2004).

Para estudiar la competencia léxica, se requiere de dos aspectos fundamentales: frecuencia y disponibilidad léxica. La frecuencia influye en el aprendizaje y consolidación del vocabulario: cuanto más se escucha o se lea una palabra, mayor será la probabilidad de adquirirla. A través de la frecuencia léxica, se han podido identificar leyes cuantitativas que describen de manera sistemática la frecuencia de palabras, la longitud, la densidad, entre otras características del léxico (Capsada Blanch y Torruella Casañas, 2017; Hernández y Ferrer i Cancho, 2019; Rasinger, 2008/2019) que forman parte de la riqueza léxica (Ávila, 1988; Ávila Muñoz, 2014; López Morales, 2011) y que se apoyan en la estadística para reforzar y mejorar la descripción matemática sobre los estudios del vocabulario (Alvar Ezquerra, 2005; Davies, 2005; Romero-Pérez *et al.*, 2018).

En cuanto a la disponibilidad léxica, su estudio se originó en Francia a mediados del siglo pasado. Su objetivo era y es recopilar y

https://doi.org/10.46744/bapl.202502.007

analizar el léxico disponible de una comunidad lingüística mediante estímulos cognitivos (pruebas de fluencia semántica) que activan el lexicón del hablante. Para analizar la información recuperada de esas pruebas (llamadas cuestionarios de disponibilidad léxica), se ha empleado una fórmula matemática que calcula el índice de disponibilidad léxica (IDL desde este momento). Este cálculo se basa tanto en la frecuencia como en el orden de aparición de las palabras en el cuestionario de disponibilidad léxica.

¿Cómo se obtiene el IDL de un léxico disponible? Para poder calcularlo, la fórmula ha sido integrada en diversos programas informáticos (LexiDisp, DispoLex y DispoCen) diseñados para el procesamiento y análisis de los datos de manera rápida y eficiente. LexiDisp, el primer programa de cómputo que incorporó la fórmula matemática de López Chávez y Strassburger Frías (1991), fue desarrollado por la Universidad de Alcalá en 1990 (García Marcos, 1997; Moreno Fernández *et al.*, 1995). Tiempo después apareció DispoLex, un *software* diseñado de manera conjunta por la Universidad de Concepción (Chile) y la Universidad de Salamanca (España) (Bartol Hernández y Hernández Muñoz, 2004), el cual cuenta con una plataforma que incluye diferentes herramientas para el análisis del léxico disponible (cálculo del IDL; análisis y obtención de la frecuencia de las palabras; generación y obtención de datos estadísticos, sobre todo). Y recientemente ha aparecido DispoCen (*software* basado en R), que permite hacer el cálculo tanto de la disponibilidad léxica como de la centralidad léxica, y que se sustenta en la hipótesis de que las palabras más comunes y compartidas por los miembros de la comunidad lingüística presentarán una alta centralidad léxica y viceversa (Ávila Muñoz *et al.*, 2021).

En el contexto mexicano, hay evidencia de que se han desarrollado *softwares* para el estudio de la disponibilidad léxica. El programa más conocido es el denominado Vocablos, cuya finalidad consistió en

identificar tipos de vocablos comunes en diferentes colectividades escolares (Reyes Valdés *et al.*, 2021). También el grupo de investigadores de Reyes Valdés, Flores Treviño y Ojeda Castañeda ha utilizado el lenguaje de programación R para ampliar el análisis de la disponibilidad léxica.

Reconociendo las aportaciones hechas por los *softwares* previamente utilizados en esta clase de estudios, en el presente trabajo de investigación se ha empleado el lenguaje de programación Python para analizar y procesar datos de disponibilidad léxica. Desde este enfoque se espera una actualización más rápida y completa para el procesamiento de los datos informáticos en esta área, sin la necesidad de programas de cómputo específicos. Trasladar la fórmula matemática de disponibilidad léxica al campo del NLP constituye una de las primeras integraciones al análisis y desarrollo de sistemas inteligentes basados en IA (Muñoz-Basols *et al.*, 2024). Esta incorporación no solo amplía la posibilidad de análisis automatizado, sino que contribuye a la reducción de la brecha digital que aún existe en el procesamiento de datos lingüísticos. El objetivo de este artículo ha sido describir la operacionalización de la fórmula matemática para medir la disponibilidad léxica según López Chávez y Strassburger Frías (2000) en lenguaje de programación Python. Su eficiencia se evidencia con un ejemplo del estudio *Análisis de disponibilidad léxica de matemáticas en estudiantes de educación superior de San Luis Potosí capital* (2024), de Moreno Sánchez.

Se optó por Python debido a que es un lenguaje de programación ampliamente conocido y utilizado en el desarrollo de *softwares*, sobre todo en los campos de la ciencia de datos y aprendizaje automático (por sus siglas en inglés, *Machine Learning*, ML). Esto representa una innovación en dicha área de estudio, ya que Python utiliza bibliotecas especializadas como TensorFlow, desarrollada por Google, que facilita la implementación de modelos de aprendizaje automático. Su facilidad de acceso y su creciente adopción en entornos académicos y tecnológicos

 https://doi.org/10.46744/bapl.202502.007

lo convierten en una opción ideal para ejecutar análisis lingüísticos, incluido el estudio de la disponibilidad léxica.

2. Marco teórico

La lingüística computacional tiene su origen en dos momentos clave en la evolución de las tecnologías de procesamiento del lenguaje entre 1940 y 1950. El primero se centra en el desarrollo de la teoría de los autómatas propuesta por Turing (1948), con la cual se generó un modelo matemático expresado por medio de un algoritmo. El segundo momento se ubica en el desarrollo de los modelos probabilísticos (o la teoría de la información) propuestos por Shannon y Weaver (1949), con los que se desarrollaron los sistemas de comunicación a partir de leyes matemáticas que rigieran la transmisión y el procesamiento de la información (Alfonseca, 2000). De esta manera, desde que aparecieron las tecnologías del lenguaje (LT desde este momento), la relación entre ser humano y computadora ha generado un binomio indisociable para el avance del conocimiento en los ámbitos culturales, sociales, académicos, entre muchos otros.

Las tecnologías de la información se presentan como una técnica para el desarrollo de aplicaciones informáticas relacionadas con el tratamiento del lenguaje y del habla (Pierrel y Romary, 2000). Como resultado del desarrollo de estas aplicaciones, las computadoras no solo leen, analizan y procesan información, sino que también generan interacciones lingüísticas con los humanos. Además, las LT (o ingeniería lingüística), que forman parte de la lingüística computacional, se han centrado desde un principio en la IA a través del procesamiento del lenguaje humano. Esto se puede observar en que más del 80 % de la población global utiliza, sin darse cuenta, algunas de las aplicaciones de las LT —que se comercializan a través de la tecnología—: reconocimiento de voz, asistentes inteligentes, traducción automática, *chatbots*, resumen de textos, búsqueda de información, subtitulado automático,

generación de corpus lingüísticos (orales o escritos), síntesis del discurso, análisis del lenguaje, generación de lenguaje natural, comprensión del lenguaje, etc. Con las LT se busca 1) estudiar, resolver y fortalecer la descripción lingüística a través del uso de la computadora, y 2) desarrollar tecnologías que faciliten su aplicación a los terrenos sociales, culturales, educativos, etc.

A través de estas tecnologías, es posible crear herramientas de procesamiento que permiten utilizar los ordenadores sin renunciar al uso del lenguaje natural como medio de integración y de intercambio de información (Cole *et al.*, 1997; Llisterri y Martí, 2002; Martí, 2001; Uszkoreit, 2002, citados en Llisterri, 2003). Esta interacción ha dado lugar a la generación de herramientas y sistemas basados en reglas y modelos estadísticos desde una perspectiva computacional y al desarrollo de lenguajes de programación orientado a aplicaciones informáticas, al desarrollo web y a la IA, principalmente.

Según la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) (2024), la IA es potencialmente capaz de imitar o incluso superar las capacidades cognitivas humanas. Su finalidad es que estos sistemas puedan reconocer, comprender, interpretar y generar el lenguaje humano en todas sus formas. Para esto, el lenguaje de programación utiliza un conjunto de reglas gramaticales de orden sintáctico y un conjunto de reglas de orden semántico, lo que permite la elaboración de algoritmos computacionales que sirven de enlace para el propósito encomendado (Baciero Fernández, 2020).

Básicamente, el lenguaje de programación está conformado por símbolos, vocabulario, palabras clave y reglas de sintaxis y de sentido que se ejecutan en un entorno, donde se incluyen un editor de texto, un compilador y un depurador. Para poder ejecutar las instrucciones —y, por consiguiente, efectuar la orden—, se utiliza un código binario de base dos (0 y 1). Todas las aportaciones tecnológicas que se han

 https://doi.org/10.46744/bapl.202502.007

desarrollado en el ámbito del NLP se pueden aplicar a cualquier trabajo lingüístico. Una de las contribuciones más relevantes de la lingüística a este campo es la vinculación del léxico de una lengua con la IA, dado que el léxico desempeña un papel fundamental en los procesos de comunicación y comprensión del mundo.

Una de las áreas del léxico que potenciaría el desarrollo de la IA es la disponibilidad léxica, ya que, según Reyes Valdés *et al.* (2021), representa el capital léxico utilizable, en el que se incluye no solo un número finito de palabras, sino también una representación individual y colectiva del mundo social al que pertenece una persona. La finalidad de la disponibilidad léxica es obtener, identificar y organizar las palabras que se pueden activar en la mente del informante a partir de un tema específico. Con la aplicación de la disponibilidad léxica, se espera comprender el estado del léxico de cualquier comunidad de habla y describir cómo está organizado el léxico en la memoria semántica y el lexicón mental. Esto se logra mediante la identificación y cuantificación de las palabras que los hablantes pueden activar de manera más rápida y frecuente para la producción lingüística, lo que revela la estructura interna de su conocimiento léxico (Bartol Hernández, 2010).

Para analizar y comprender la estratificación de la información léxica que se obtiene de una conversación, el binomio Python-disponibilidad léxica puede ser fundamental, ya que permite abordar el estudio científico del lenguaje humano mediante un análisis automatizado de tareas. Estas tareas incluyen desde el análisis del léxico por IDL hasta la generación de redes léxicas que contribuyan a explicar de mejor manera el comportamiento léxico observable en el lexicón mental a través de la aplicación de la lingüística computacional, de la cual forma parte inherente la disponibilidad léxica.

3. Metodología

Para describir el proceso de operacionalización de la fórmula de disponibilidad léxica se ha hecho uso de una metodología cualitativo-descriptiva. En este apartado, se describe detalladamente el análisis realizado. El equipo de trabajo tardó quince meses para desarrollar la operacionalización de la fórmula de disponibilidad léxica.

3.1. Lenguaje de programación

El lenguaje de programación Python se caracteriza por realizar 1) una curva de aprendizaje fácil y dinámica que puede utilizarse con otros lenguajes de programación (Java, C y C++) y que puede funcionar con diferentes sistemas operativos de computadora (Windows, macOs, Linux, Unix, entre otros); 2) una sintaxis simple y efectiva para comunicarse con la computadora; 3) el respaldo tecnológico propio del programa y el apoyo de una comunidad internacional, y 4) el uso de librerías especializadas que acortan el código generado por el usuario.

Para trabajar la fórmula matemática de disponibilidad léxica de López Chávez y Strassburger Frías (2000), se utilizó la herramienta interactiva Jupyter Notebook como ambiente de ejecución de Python. Esta herramienta permite hacer uso de todas las librerías de Python para la IA, en especial para el aprendizaje automático, la generación de librerías de estadística, el procesamiento de palabras, el desarrollo de la ciencia de datos, entre muchas otras tareas. Además, a través del uso de Jupyter Notebook, se generan los prototipos de la aplicación y la posibilidad de compartirlos con los usuarios para hacer pruebas.

Por un lado, Jupyter Notebook presenta la capacidad de soportar múltiples lenguajes de programación a través de *kernels* y la posibilidad de incluir celdas de código y celdas de texto en un mismo documento, lo que facilita el trabajo. Además, las celdas se pueden ejecutar de manera interactiva, por lo cual los resultados se obtienen de manera

 https://doi.org/10.46744/bapl.202502.007

inmediata. Por otro lado, la herramienta Jupyter Notebook se diferencia de otros programas porque comparte de forma rápida los proyectos y los *notebooks* que se pueden guardar en formato JSON, lo cual facilita su distribución y colaboración a través de GitHub o JupyterHub. Finalmente, con esta herramienta, se pueden trabajar de forma directa los *notebooks* desde el navegador sin que se instale un *software* adicional, gracias a los servicios de Google Colab. Todo esto hace que Python sea el lenguaje de programación ideal para modernizar aspectos metodológico-computacionales en los estudios de disponibilidad léxica.

3.2. Grupo de trabajo

Python fue integrado al ámbito de la disponibilidad léxica por un grupo de trabajo de carácter interdisciplinario, específicamente, compuesto por lingüistas, ingenieros y pedagogos pertenecientes a diferentes universidades públicas de San Luis Potosí, México. Cada uno de los integrantes es especialista en su área y ha vinculado sus conocimientos a este trabajo de investigación para potenciar esta área de estudios en el contexto local, nacional e internacional.

3.3. La muestra

La muestra que sirvió como base para la operacionalización de la fórmula se obtuvo de algunos ejemplos tomados de la tesis de maestría *Análisis de disponibilidad léxica de matemáticas en estudiantes de educae ción superior de San Luis Potosí capital* (2024), de Moreno Sánchez. Esta investigación es el primer trabajo en el que se aplicó esta nueva herramienta para el procesamiento de la información y obtención del IDL. En la Figura 1, se presenta la fórmula matemática de disponibilidad léxica utilizada para el análisis de la muestra de esta tesis, que estuvo conformada por 283 participantes del ciclo escolar 2022-2024 de tres instituciones públicas de educación superior pertenecientes al estado de San Luis Potosí (capital) —lo que garantizó una heterogeneidad de los datos—.

Figura 1

Fórmula matemática de disponibilidad léxica

$$D\left(P_j\right) = \sum_{i=1}^{n} e^{-2{,}3 \times \left(\frac{i-1}{n-1}\right)} \times \frac{f_{ji}}{I_1}$$

Donde:
$D(P_j)$: Disponibilidad de la palabra j
n: Máxima posición alcanzada
i: Número de posición de que se trata
j: Índice de la palabra en cuestión
e: Número natural (2.718181818459045)
f_{ji}: Frecuencia absoluta de la palabra j en la posición i
I_1: Número de informantes que participan en la encuesta

Como se observa, la fórmula se centra en a) la frecuencia absoluta con que fue dicha cada palabra en cada posición (f_{ji}); b) la frecuencia absoluta de la palabra, que resulta de sumar las diferentes frecuencias en cada posición; c) el número de participantes en la encuesta (I_1); d) el número de posiciones alcanzadas en la encuesta del centro de interés (n), y e) las posiciones en que fue dicha cada palabra (i). La aplicación del número (e) elevado al exponente que se presenta en la fórmula es el verdadero ponderador. Esto permite combinar sin distorsión la frecuencia y la posición de la emisión de cada palabra, ya que arroja una ponderación limitada entre 1 y 0.1, independientemente del número de participantes de la extensión de los listados producidos por cada informante, el número de sujetos que llegan a cada posición y la frecuencia de aparición del vocablo.

3.4. Procedimiento de aplicación para el análisis en Python

Para llevar a cabo la aplicación de la fórmula de disponibilidad léxica en el programa Python, se recopiló la información proporcionada por los participantes de forma tradicional, mediante un cuestionario de disponibilidad léxica en un tiempo determinado (tres minutos por centro), y se llevaron a cabo los criterios de tratamiento de datos siguiendo el proceso descrito por Samper Padilla (1998): a) estandarización del corpus, respetando, en la medida de lo posible, la versión original de cada una de las listas; b) eliminación de términos repetidos;

 https://doi.org/10.46744/bapl.202502.007

c) corrección ortográfica de cada término hasta lograr la neutralización de variantes flexivas y la elisión de marcas comerciales, y d) asignación de una clave de identificación para el tratamiento de la información del participante. Dicha clave permite registrar 1) el campo con información sociológica básica del informante formado por cinco caracteres sucesivos —cada uno de los cuales representa la codificación de una variante—; 2) un campo de identificación del usuario de tres caracteres, y 3) un campo de identificación del centro de interés de dos caracteres (Moreno Fernández *et al.*, 1995). La clave de identificación resultante fue 0001020116.

Una vez realizadas las adecuaciones de contenido y de estilo, se revisó que los datos estuvieran en archivo de texto (Word) ordenados de la siguiente manera: el centro de interés separado por «##___##» y cada palabra separada por una coma y un espacio. Esto aseguró que los datos fueran procesados de forma rápida y eficiente por el programa. A continuación, se presentan dos ejemplos:

(1) a. *##16## plomero, estudiante, albañil, encuestador, fontanero, psicólogo, matemático, ebanista.*
b. *##14## oso, capibara, puma, mosco, mosca, burro.*

Se descargó el programa Python mediante el acceso al enlace https://www.python.org/downloads/[1]. No obstante, este programa se puede usar en línea a través de la página web https://jupyter.org/try-jupyter/lab/.

Para establecer los tipos de celda se usó Jupyter Notebook: una celda para escribir código y otra para documentar el libro de trabajo. Los datos léxicos se registraron en una hoja de cálculo de Excel con la

1 Se sugiere que se utilice la versión más reciente y se realicen pruebas de funcionamiento.

finalidad de estandarizar el formato de entrada. En cuanto a las columnas, a partir de la segunda se enumeró la producción del informante desde el número 1 hasta *n*, según la distancia establecida para el análisis. En cuanto a las filas, la primera fue designada para los títulos de los datos, mientras que desde la segunda en adelante se relacionaron los vocablos correspondientes a cada participante. Este formato permitió la lectura e interpretación de la información, como se muestra en la Figura 2:

Figura 2
Ejemplo de los formatos registrados en Excel

	A	B	C	D
1	CODIGO	1	2	3
2	2111310401	progresion	razon	primer-termin
3	2111410402	factorizacion	racionalizacion	binomio
4	2111110403	seno	coseno	tangente
5	2111110404	poligono	suma-de-angulo	exterior
6	2111110405	muestreo	suma	media

Una vez codificada la muestra, se usó la librería PANDAS (Python for Data Analysis) para procesar los datos fuente desde las hojas de cálculo en Excel. Esta librería es una extensión de NumPy para manipular y analizar datos. Por medio de NumPy, se ofrecen estructuras de datos y operaciones para manipular tablas numéricas y series temporales conocidos como *dataframes* (o matrices de datos). La funcionalidad de los *dataframes* permitió llevar los datos de la fórmula de disponibilidad léxica al terreno del lenguaje de programación. Para evidenciar lo que se ha mencionado, se presenta la estructura correspondiente:

(2)
```
import pandas as pd
# Leer datos del archivo Excel
```

 https://doi.org/10.46744/bapl.202502.007

```
dfe = pd.read_excel('../Data_Test/M_1.xlsx')
df = pd.DataFrame(dfe)
```

Se etiquetaron en una sola base los datos obtenidos de Excel. Este procedimiento consistió en agrupar y contar el número de palabras totales que se obtuvieron de los cuestionarios de disponibilidad léxica. Los comandos para contar el número de palabras totales que han aparecido en el corpus son los siguientes:

```
(3) # Contar las palabras generadas y las vacías
    df['num_contenido'] = df.iloc[:, 1:].notna().sum(axis=1)
    df['num_vacias'] = df.iloc[:, 1:].isna().sum(axis=1)
```

Con la finalidad de organizar la información léxica de cada centro de interés, se generó un filtrado de la información en NumPy, pues este permite separar los datos por centro de interés. Para esta operación, se utilizó la columna del código para instruirle a Python que solo tomara los datos del centro de interés *X* que se encontrasen en la columna *X* o *Y* cualquiera. Obsérvese el ejemplo siguiente, en el que se le indicó a Python que tomase únicamente los datos del centro de interés 01 (c01):

```
(4) # Filtrar los datos por CODIGO
    c01 = df[df['CODIGO'].astype(str).str.endswith(('01'))]
```

Como resultado se obtuvieron las palabras generales. Luego, mediante la siguiente instrucción, se le indicó al programa que solo contrajera las palabras del c01:

```
(5) palabras_unicas = df.iloc[:, 1:].stack().unique().
```

Para obtener la frecuencia de cada palabra de la lista, se solicitó al programa que buscara todos los términos de la columna 1 y que eliminara los vocablos duplicados. Con base en el siguiente comando, se pudo hacer el cálculo por cada palabra, por el número de repeticiones o por la frecuencia:

(6) def calc_freq(palabra, df):
return df.apply(lambda row: row.isin([palabra]).sum(), axis=1).sum()

Lo anterior permitió reunir los vocablos encontrados de cada centro de interés. Además de la frecuencia de las palabras obtenida, también se necesitó conocer el número de términos de cada columna. Esto fue fundamental dentro de la fórmula matemática de disponibilidad léxica porque, para realizar su cálculo, se requería de un parámetro conocido como frecuencia absoluta de la palabra *j* en la posición *i* (f_{ji}). Para obtenerlo, se dieron las siguientes órdenes en Python:

(7) def ubicacion(palabra, df):
contador = [0] * (df.shape[1] - 1) # Inicializar un vector de ceros
for col in range(1, df.shape[1]):
contador[col - 1] = (df.iloc[:, col] == palabra).sum()
return contador
def encontrar_max_posicion(ubicaciones):
max_indice = ubicaciones.index(max(ubicaciones)) return max_indice + 1

Se calculó la disponibilidad léxica a partir del comando presentado en la Figura 3, y se muestra un ejemplo del procesamiento de la información en la Figura 4:

 https://doi.org/10.46744/bapl.202502.007

Figura 3
Comando utilizado para calcular la disponibilidad léxica

```
def disponibilidad(c,i,n,Fji,I):
  try:
    return math.exp(-c(i-1)/(n-1))*((Fji)/(I))
  except: ZeroDivisionError:
    return math-exp(-c*0)*((Fji)/(I))
```

Figura 4
Procesamiento de la información

Palabra	Frecuencia	Ubicacion	max_posicion	IDLT
seno	5	[5, 0, 0]	1	1.000000
coseno	5	[0, 5, 0]	2	0.100259
tangente	5	[0, 0, 5]	3	0.100259

4. Resultados

Para demostrar su efectividad, se aplicó la fórmula matemática en Python al trabajo de Moreno Sánchez (2024), como se mencionó anteriormente. Para la ejemplificación se consideraron cinco participantes con tres centros de interés de matemáticas. En la Tabla 1 se presenta la organización de la información obtenida del cuestionario de disponibilidad léxica. Para comprender su estructuración se utilizó la siguiente nomenclatura: «Sujeto», «Centro de interés» y «Producción». La primera corresponde al número asignado a cada participante que respondió la encuesta (primeros tres dígitos). La segunda designa el número de centros de interés que se analizan (dos últimos dígitos), el cual se vincula directamente con la cantidad de centros de interés que participan en la investigación. La tercera categoría, «Producción», indica el número total de palabras que el informante proporcionó a partir del

cuestionario de disponibilidad léxica, y la posición señala la aparición de la palabra en el cuestionario. Así, por ejemplo, el código 00101 corresponde al participante 001 asociado al centro de interés 01, quien proporcionó la palabra *adición* en primer lugar en el cuestionario.

Tabla 1

Producción de palabras de cinco participantes del procesamiento

Sujeto-Centro de interés	Producción					
	pt1	pt2	pt3	pt4	pt5	pt6
00101	adición	producto	raíz	división	numera-dor	
00102	exponente	raíz	potencia	literal	variable	
00103	seno	coseno	tangente	cotangente	cosecante	
00201	resta	multiplicación	división	suma-de-fracción		
00202	regla-del-pro-ducto					
00203	función-tri-gonomé-trica	suma	división			
00301	suma	resta	división	fracción		
00302	variable	función	potencia			
00303	tangente	secante	seno			
00401	matemática	calcular	resultado			
00402	multiplica-ción	x	y	fracción	resta	suma
00403	cociente	cateto	tangente			
00501	división	multiplicación	suma	resta		
00502	numero	álgebra-lineal-y-abstracta	fórmula			
00503	fórmula	medida	altura	área		

Nota. pt = Posición de la palabra.

 https://doi.org/10.46744/bapl.202502.007

Después se procedió a calcular el total de palabras. Se realizó una suma de todas las producciones generadas en el centro de interés 01 por los cinco participantes, que dio como resultado veinte palabras. Para el paso siguiente, solamente se calculó cuántas palabras diferentes había, es decir, todas las palabras que no se repetían en el centro de interés 01 de los cinco participantes (trece en total). Del total de palabras diferentes, se formó una matriz de distribución con las posiciones de cada una de ellas, como se observa en la Tabla 2:

Tabla 2

Matriz de posiciones por participante de cada palabra

Palabra	Participantes				
	w1	w2	w3	w4	w5
Adición	1	0	0	0	0
Producto	2	0	0	0	0
Raíz	3	0	0	0	0
División	4	3	3	0	1
Numerador	1	0	0	0	0
Resta	0	1	2	0	4
Multiplicación	0	2	0	0	2
Suma de fracción	0	4	0	0	0
Suma	0	0	1	0	3
Fracción	0	0	0	4	0
Matemática	0	0	0	1	0
Calcular	0	0	0	2	0
Resultado	0	0	0	3	0

Nota. Cuando el valor es cero, significa que el participante no produjo esa palabra. w = Participante.

Luego, se calculó la frecuencia absoluta, la cual muestra la sumatoria de los valores absolutos de aparición que presenta cada palabra en un mismo centro de interés. Los resultados se observan en la Tabla 3:

Tabla 3
Frecuencia absoluta

Palabra	Posición de la palabra					Frecuencia absoluta
	w1	w2	w3	w4	w5	
	Posición (pt)					
Adición	1	0	0	0	0	1
Producto	2	0	0	0	0	1
Raíz	3	0	0	0	0	1
División	4	3	3	0	1	4
Numerador	1	0	0	0	0	1
Resta	0	1	2	0	4	3
Multiplicación	0	2	0	0	2	2
Suma de fracción	0	4	0	0	0	1
Suma	0	0	1	0	3	2
Fracción	0	0	0	4	0	1
Matemática	0	0	0	1	0	1
Calcular	0	0	0	2	0	1
Resultado	0	0	0	3	0	1

Nota. w = Participante; pt = Posición de la palabra.

Por ejemplo, en la Tabla 3 se observa que el participante 001 (w1) mencionó *división* en la posición 4; el participante 002 (w2), en la posición 3; el participante 003 (w3), en la misma posición; el participante 004 (w4) no la produjo, y el participante 005 (w5) la mencionó en la posición 1. Dicha palabra apareció un total de cuatro veces en la matriz, lo que corresponde a su frecuencia absoluta. Para ampliar este análisis, se comparte la distribución de las palabras en la matriz del informante 001 en la Tabla 4:

 https://doi.org/10.46744/bapl.202502.007

Tabla 4

Análisis de posición de las palabras producidas por el primer participante

Palabra	Producción					Frecuencia absoluta
	pt1	pt2	pt3	pt4	pt5	
Adición	1	0	0	0	0	1
Producto	0	1	0	0	0	1
Raíz	0	0	1	0	0	1
División	0	0	0	1	0	1
Numerador	1	0	0	0	0	1

Nota. pt= Posición de la palabra.

Para el cálculo del IDL de López Chávez y Strassburger Frías (2000), la información se organiza en una tabla de este tipo (ver Tabla 4) con el propósito de determinar la frecuencia de las palabras por posición. En ella, como se observa, el informante registró la palabra *adición* en la posición 1, *producto* en la posición 2, *raíz* en la posición 3, *división* en la posición 4 y *numerador* en la posición 1. En esta etapa ya no es relevante identificar al sujeto que las proporcionó, sino registrar cada palabra y la cantidad de veces que aparece en cada posición. Es decir, ahora se contabiliza cuántas palabras se localizaron en cada posición.

La fórmula establece que, para hacer el cálculo del IDL, es necesario sumar los valores obtenidos desde la posición inicial hasta la máxima posición alcanzada por cada palabra, considerando su presencia o ausencia en cada posición. A continuación, se muestra la aplicación de la fórmula para obtener el IDL utilizando la palabra *producto* como ejemplo. Esta palabra no presentó frecuencia en la posición 1 (es decir, frecuencia 0), pero sí en la posición 2 (frecuencia 1), la cual Python consideró para el cálculo del IDL:

$$DP_j = \left(e^{-2.3\left(\frac{1-1}{2-1}\right)}\right)\left(\frac{0}{5}\right) + \left(e^{-2.3\left(\frac{2-1}{2-1}\right)}\right)\left(\frac{1}{5}\right)$$

$$DP_j = \Sigma_{i=1}^{2} 0.10025\ (0) + 0.10026\ (0.2)$$

$$DP_j = \Sigma_{i=1}^{2} 0 + 0.020052$$

$$DP_j = 0.020052$$

La fórmula establece que se debe calcular el IDL para cada posición en la que aparezca la palabra a lo largo de la matriz de respuestas. Así, se obtienen divisiones como 0/5, 1/5, hasta cubrir la totalidad (α) de la matriz. En la aplicación de la formulada dada, la división 0/5 indica que la palabra no se registró en la posición 1, mientras que 1/5 señala que el vocablo apareció una vez en la posición 2. En este caso, el cálculo del IDL se detuvo tras considerar ambas posiciones, dado que no hubo producción de la palabra *producto* en posiciones posteriores.

Para obtener el IDL de cada vocablo, se calcula el valor correspondiente a cada palabra en cada posición, y el cálculo final de cada vocablo se obtiene sumando los valores de todas las posiciones. Cabe destacar que la fórmula considera tanto la posición como la frecuencia de los vocablos en sus cálculos. En la Tabla 5 se presenta el IDL de otros vocablos:

Tabla 5
Obtención del IDL

Vocablo	IDL	Frecuencia	Frecuencia relativa
Razón	0.4000000	2	0.22
Primer-término	0.0200520	1	0.11
Seno	0.0401040	2	0.22
Binomio	0.200000	1	0.11
Tangente	0.0200520	1	0.11
Progresión	0.2000000	1	0.11
Factorización	0.0200520	1	0.11

 https://doi.org/10.46744/bapl.202502.007

5. Conclusiones

En este artículo se ha descrito la operacionalización de la fórmula matemática de disponibilidad léxica en Python, mostrando su funcionalidad paso a paso y resaltando su valor como herramienta para el análisis lingüístico. El objetivo ha sido reforzar los estudios sobre disponibilidad léxica no solo en lo referente a la metodología de recolección y organización de datos, sino también en la interpretación sistemática de los resultados. Asimismo, se ha destacado la importancia de la disponibilidad léxica para comprender la riqueza y la organización del vocabulario en contextos específicos, lo que contribuye a una visión más amplia del comportamiento lingüístico de los hablantes.

El desarrollo de la fórmula matemática en Python representa una herramienta clave, ya que permite automatizar el cálculo del índice de disponibilidad léxica y procesar los datos de manera eficiente, con capacidad de adaptación a distintos tipos de análisis y formatos de información. A diferencia de programas como LexiDisp, DispoLex y DispoCen, que están diseñados para funciones concretas y ofrecen poca flexibilidad para personalizar procedimientos o integrar nuevos tipos de análisis, Python permite una implementación más versátil. Su compatibilidad con librerías especializadas de estadística, procesamiento de lenguaje natural e inteligencia artificial facilita la expansión de los estudios de disponibilidad léxica a contextos multidisciplinarios y a proyectos más complejos.

Por estas razones, Python se considera una herramienta idónea para la modernización de los estudios de disponibilidad léxica, no solo agilizando el trabajo analítico, sino también ampliando significativamente las posibilidades tecnológicas de la lingüística del corpus en la actualidad.

Referencias bibliográficas

Alfonseca, M. (2000). La máquina de Turing. En A. Martinón (Ed.), *Las matemáticas del siglo XX: una mirada en 101 artículos* (pp. 165-168). Universidad de La Laguna; Sociedad Canaria Isaac Newton de Profesores de Matemáticas; Nivola.

Alvar Ezquerra, M. (2005). La frecuencia léxica y su utilidad en la enseñanza del español como lengua extranjera. En M.ª A. Castillo Carballo, O. Cruz Moya, J. M. García Platero y J. P. Mora Gutiérrez (Coords.), *Las gramáticas y los diccionarios en la enseñanza del español como segunda lengua: deseo y realidad* (pp. 19-33). Universidad de Sevilla.

Ávila, R. (1988). Lengua hablada y estrato social: un acercamiento lexicoestadístico. *Nueva Revista de Filología Hispánica*, *36*(1), 131-148. https://doi.org/10.24201/nrfh.v36i1.668

Ávila Muñoz, A. M. (2014). Patrones sociolingüísticos de la riqueza léxica. Estudio basado en una propuesta original para el cálculo del índice de la densidad léxica virtual de los hablantes. *LEA: Lingüística Española Actual*, *36*(2), 171-194.

Ávila Muñoz, A. M., Sánchez Sáez, J. M.ª, y Odishelidze, N. (2021). *DispoCen*. Mucho más que un programa para el cálculo de la disponibilidad léxica. *ELUA. Estudios de Lingüística*, (35), 9-36. https://doi.org/10.14198/ELUA2021.35.1

Baciero Fernández, J. I. (2020). *Elaboración de un modelo de reconocimiento de entidades nominales (NER) para su uso en aplicaciones de procesamiento del lenguaje natural (NLP)* [Trabajo de fin de grado, Universidad Politécnica de Madrid]. Archivo Digital UPM. https://oa.upm.es/62858/

 https://doi.org/10.46744/bapl.202502.007

Bartol Hernández, J. A. (2010). Disponibilidad léxica y selección del vocabulario. En R. M.ª Castañer Martín y V. Lagüéns Gracia (Eds.), *De moneda nunca usada: estudios dedicados a José M.ª Enguita Utrilla* (pp. 85-107). Instituto Fernando el Católico.

Bartol Hernández, J. A., y Hernández Muñoz, N. (2004, 3-7 de mayo). *Dispolex: banco de datos de la disponibilidad léxica* [Presentación de la ponencia]. VI Congreso de Lingüística General, Santiago de Compostela, España.

Capsada Blanch, R., y Torruella Casañas, J. (2017). Métodos para medir la riqueza léxica de los textos. Revisión y propuesta. *Verba. Anuario Galego de Filoloxía*, *44*, 347-408. https://doi.org/10.15304/verba.44.3155

Davies, M. (2005). Vocabulary Range and Text Coverage: Insights from the Forthcoming *Routledge Frequency Dictionary of Spanish* [Extensión del vocabulario y alcance textual: perspectivas del próximo *Diccionario de frecuencia del español de Routledge*]. En D. Eddington (Ed.), *Selected Proceedings of the 7th Hispanic Linguistics Symposium* (pp. 106-115). Cascadilla Proceedings Project.

García Marcos, F. (1997). *Estudios de disponibilidad léxica*. GRUSTA.

Gómez Molina, J. R. (2004). La subcompetencia léxico-semántica. En J. Sánchez Lobato e I. Santos Gargallo (Dirs.), *Vademécum para la formación de profesores. Enseñar español como segunda lengua (L2) / lengua extranjera (LE)* (pp. 491-510). SGEL.

Hernández, T., y Ferrer i Cancho, R. (2019). *Lingüística cuantitativa: la estadística de las palabras*. Emse Edapp.

Llisterri, J. (2003). Lingüística y tecnologías del lenguaje. *LynX. Panorámica de Estudios Lingüísticos*, (2), 9-71.

López Chávez, J., y Strassburger Frías, C. (2000). El diseño de una fórmula matemática para obtener un índice de disponibilidad léxica confiable. *Anuario de Letras*, *38*, 227-251.

López Morales, H. (2011). Los índices de «riqueza léxica» y la enseñanza de lenguas. En J. de Santiago Guervós, H. Bongaerts, J. J. Sánchez Iglesias, M. Seseña Gómez (Eds.), *Del texto a la lengua: la aplicación de los textos a la enseñanza-aprendizaje del español L2-LE* (Vol. 1, pp. 15-28). Asociación para la Enseñanza del Español como Lengua Extranjera.

Moreno Fernández, F., Enrique Moreno Fernández, J., y García de las Heras, A. (1995). Cálculo de disponibilidad léxica. El programa LexiDisp. *Lingüística*, (7), 243-250.

Moreno Sánchez, A. (2024). *Análisis de disponibilidad léxica de matemáticas en estudiantes de educación superior de San Luis Potosí capital* [Tesis de maestría]. Universidad Autónoma de San Luis Potosí.

Muñoz-Basols, J., Fuertes Gutiérrez, M., y Cerezo, L. (Eds.). (2024). *La enseñanza del español mediada por tecnología: de la justicia social a la inteligencia artificial (IA)*. Routledge. https://doi.org/10.4324/9781003146391

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. (2024). *Guía para el uso de IA generativa en educación e investigación*.

Pierrel, J.-M., y Romary, L. (2000). Dialogue homme-machine [Diálogo entre el hombre y la máquina]. En J.-M. Pierrel (Dir.), *Ingénierie des langues* (pp. 331-349). Hermes.

 https://doi.org/10.46744/bapl.202502.007

Rasinger, S. M. (2019). *La investigación cuantitativa en lingüística: una introducción* (1.ª ed.). Akal. (Obra original publicada en 2008)

Reyes Valdés, D., Reyes Valdés, J. R., Flores Treviño, M.ª E., y Ojeda Castañeda, R. B. (2021). Una propuesta de herramientas informáticas para el tratamiento estadístico del índice de disponibilidad léxica en estudios correlacionales de educación y movilidad social. *Forma y Función*, *34*(1). https://doi.org/10.15446/fyf.v34n1.80581

Romero-Pérez, I., Alarcón-Vásquez, Y., y García-Jiménez, R. (2018). Lexicometría: enfoque aplicado a la redefinición de conceptos e identificación de unidades temáticas. *Biblios*, (71), 68-80. https://doi.org/10.5195/biblios.2018.466

Samper Padilla, J. A. (1998). Criterios de edición del léxico disponible: sugerencias. *Lingüística*, (10), 311-333.

Shannon, C. E., y Weaver, W. (1949). *The Mathematical Theory of Communication* [Una teoría matemática de la comunicación]. The University of Illinois Press.

Turing, A. (1948). Intelligent Machinery [Maquinaria inteligente]. En B. Meltzer y D. Michie (Eds.), *Machine Intelligence 5* (pp. 3-23). Edinburgh University Press.