Predicción de brotes de COVID-19 a partir de tendencias de búsqueda en Google

Lucía Lemes *Maestría en Ciencia de Datos y Aprendizaje Automático*
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
llemes@cup.edu.uy
Manuel Lara *Maestría en Ciencia de Datos y Aprendizaje Automático*
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
lararojas.mr@gmail.com

Resumen

En el presente trabajo se aborda la migración de datos y consultas relacionadas con las tendencias de búsqueda en Google para predecir brotes epidemiológicos, específicamente el de COVID-19. Se tienen datos de porcentajes de búsqueda en Google Trends, así como datos de casos activos publicados por el GUIAD-COVID-19, separados por departamento y día, en el período comprendido entre el 1 de febrero de 2020 y el 30 de abril de 2022. El objetivo es modelar estos datos en una base de datos no relacional, en este caso, Neo4j AuraDB, y realizar consultas con operaciones medianamente complejas. Entre las consultas a implementar se exploraron algunas donde se utilizaba el valor de correlación cruzada entre las búsquedas y los casos, así como otras donde se buscaban tendencias altas en las búsquedas para predecir brotes. La predicción de eventos epidemiológicos de forma acertada está por fuera del alcance de este trabajo, ya que involucra un análisis más complejo y cuidadoso de los datos seleccionados, así como los modelos elegidos para definir la ocurrencia de un brote. Finalmente, se logró implementar la base de datos de grafo y la mayoría de las consultas deseadas, quedando por delante la tarea de refinación de las consultas y métodos empleados.

I. Introducción

La detección de brotes epidemiológicos para ciertas enfermedades de transmisión directa, es una tarea que insume numerosos recursos; especialmente cuando se realiza un seguimiento de casos y contactos, con el objetivo de encontrar nuevos infectados y anticipar brotes en regiones delimitadas, o para hallar la fuente donde se originó el brote.

Históricamente, se ha utilizado datos estadísticos o poblacionales para obtener información sobre una epidemia en numerosas ocasiones. Entre ellas, se destaca el accionar del médico inglés Jhon Snow que, mediante datos espaciales de muertes por cólera, logró identificar pozos contaminados en Londres, 1854 [1].

Estudios más recientes han buscado relaciones entre el comportamiento de los brotes epidemiológicos con datos de comportamiento, como pueden ser estadísticas de movilidad [2], búsquedas en Google [3]–[5], entre otros. En este trabajo se plantea incorporar datos de búsquedas en Google (tanto en la web como en la sección noticias) junto con recuentos de casos activos de COVID-19 entre abril de 2020 y abril de 2022, segregados por departamento, a una base de datos de grafos.

El objetivo del trabajo fue evaluar la posibilidad de utilizar una base de datos de grafos para representar las búsquedas y registrar los casos activos, a la vez que se realizaban consultas exploratorias sobre ellos. Dichas consultas apuntaron a evaluar si las tendencias de búsqueda se relacionaban con la aparición de casos de COVID-19 o si funcionaban como predictor de un aumento de casos en cierta región o intervalo temporal. Adicionalmente, se planteó explorar la posibilidad de implementar consultas que incluyeran operaciones matemáticas de mayor complejidad que las dadas por el lenguaje de consulta *Cypher*, como podía ser el cálculo del coeficiente de correlación y el ajuste de los coeficientes para una regresión lineal.

II. Desarrollo

Para la ejecución del trabajo se decidió emplear una instancia gratuita de Neo4j AuraDB, un servicio de base de datos de grafo en la nube. La obtención de los datos, carga de la base de datos y consultas se realizaron mayormente en lenguaje Python 3, a partir de los paquetes *pytrend* y *neo4j*. Los jupyter notebook utilizados se pueden acceder mediante el repositorio de GitLab [6] asociado a este trabajo.

II-A. Obtención de datos

Los datos empleados se obtuvieron de dos fuentes independientes:

1. Los casos activos por departamento y día se obtuvieron como datos abiertos en la página web del Grupo Uruguayo Interdisciplinario de Análisis de Datos de COVID-19[1]. Puntualmente, se utilizó el archivo *Cantidad de activos por departamento en formato CSV*.

[1]Disponible en https://guiad-covid.github.io/data/estadisticasuy/estadisticasuy_dpto/

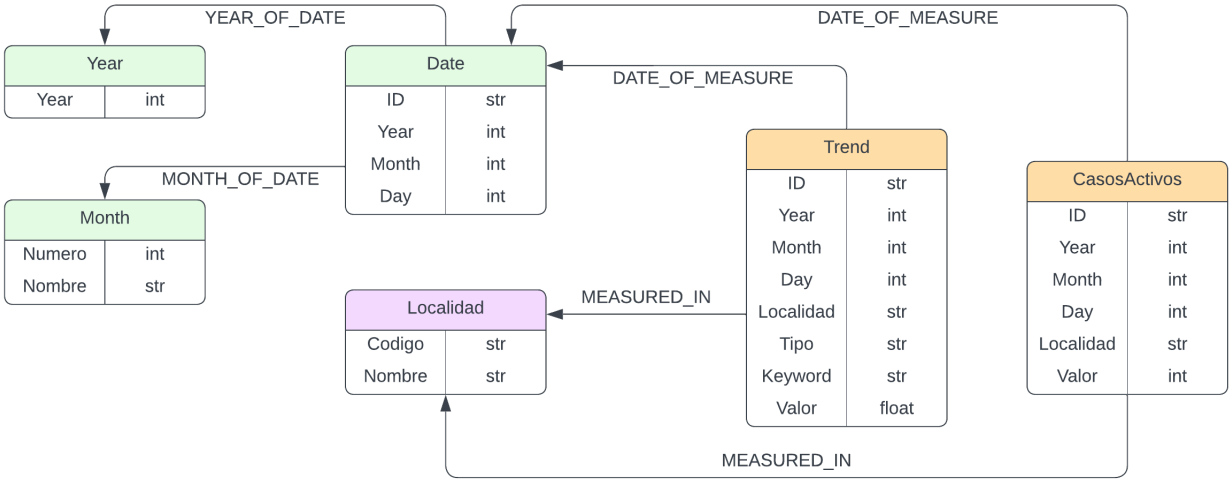

Figura 1: Diagrama de base de datos de grafo, para datos de búsquedas en Google y datos de activos de COVID-19. En verde claro se señalan los nodos pertenecientes al subgrafo temporal, en violeta el perteneciente al subgrafo espacial y en naranja los nodos del subgrafo de datos.

2. En segundo lugar, para los datos de búsquedas web de Google se empleó un script en jupyter notebook que modificaba una función del paquete *pytrend* y permitía descargar las búsquedas diarias de cierta palabra clave en la localidad indicada. Las mismas también podrían obtenerse de la web de *Google Trends*[2].

El script modificado permitía definir una palabra clave, una localización, un intervalo temporal (inicio y fin) y un tipo de búsqueda (dependiendo de si interesaban las búsquedas realizadas en motores de búsqueda en línea o búsquedas específicas de noticias), y devolvía un CSV conteniendo el valor de la búsqueda para cada día del intervalo. En total se seleccionaron seis palabras clave sobre las que se evalúan las búsquedas, la mismas son:

- covid-19
- dolor
- tos
- fiebre
- olor
- gusto

Las regiones de búsquedas se corresponden con cada uno de los departamentos de Uruguay, indicados mediante sus códigos ISO 3166-2.

En total, se obtuvieron los casos activos para cada uno de los 19 departamentos, desde el 29 de abril de 2020 hasta el 17 de abril de 2022. Esto dio como resultado un total de 13642 datos de esta índole. Respecto de los datos de Google Trends, se extrajo información de 19 regiones entre el 01 de febrero de 2020 y el 30 de abril de 2022, para un total de seis palabras claves y dos tipos de búsqueda (*web* y *news*), lo que dio un número máximo total de 186960 datos de búsqueda. Es importante mencionar que para algunas combinaciones de palabras, localidades y tipos no existían datos de búsqueda, por lo que la cantidad final de datos fue menor.

II-B. Diseño de la base de datos de grafos

Dado que los datos obtenidos tenían características espaciales y temporales, se siguió parcialmente los lineamientos de Feng et. al. [7] para el diseño del grafo. Se definieron entonces nodos temporales, que recopilaban información sobre el año, mes y fecha, nodos espaciales, con información sobre las localidades, y nodos de datos, que comprendían aquellos que registraban los valores de búsqueda y casos activos. En la Figura 1 se puede observar el diagrama de la base de datos, incluyendo cada tipo de nodo, sus atributos y relaciones.

Tener una estructura que separe las características espaciales y temporales del resto es beneficiosa desde el punto de vista de la búsqueda, ya que permite obtener los nodos de *Trends* y *CasosActivos* fácilmente a partir de su relación con cierta localidad o fecha.

El diseño de la base de datos estuvo limitado, en parte, por la implementación posterior de la misma. Dado que se empleó una instancia gratuita de Neo4j AuraDB para la carga de datos, se tenía una limitación de almacenamiento de 200000 nodos y 400000 relaciones. Considerando que el número total de datos de búsqueda y casos activos final fue de 171232 y cada uno

[2]Disponible en https://trends.google.es/trends/

tenía al menos dos relaciones (DATE_OF_MEASURE y MEASURED_IN), se esperaba un espacio limitado para expandir estas relaciones. En ese sentido, el nodo *Date* surgió como intermediario entre los nodos *Year* y *Month*, con el objetivo de disminuir la cantidad de relaciones. Asiimismo, se desistió de crear nodos individuales de tipo *Keyword* para facilitar la búsqueda, no sólo por la cantidad de relaciones que introducía, sino también porque la cantidad de palabras clave era reducida (sólo seis). En caso de que se deseara expandir la base de datos a una con más palabras clave, este añadido es algo que podría considerarse (para evitar tener que realizar selecciones sobre atributos cuando se tienen millones de nodos).

II-C. Carga de datos

La carga de los datos se realizó sobre una instancia gratuita de Neo4j AuraDB.

Dado que se tenían los datos de búsquedas y casos activos en CSV individuales, se realizó una transformación previa de los datos a nuevos archivos CSV, uno por cada tipo de nodo, donde cada fila definiera un nuevo nodo de cierto tipo, y las columnas contuvieran los valores de sus atributos.

Para la implementación en jupyter notebook, se utilizó el paquete neo4j junto con la cláusula LOAD CSV. Para cada nodo se definieron atributos que debían ser únicos en valor, garantizando así que no se cargaban nodos idénticos más de una vez. Un ejemplo del código utilizado para el caso de *Trend* puede verse en el Listado 1.

Adicionalmente, en los nodos de etiqueta *Trend* y *CasosActivos*, dado su gran número de ocurrencias, se empleó además el procedimiento CALL { } IN TRANSACTIONS, para realizar la carga de datos en distintos lotes. La consulta que realiza la carga de los nodos *Trend* puede verse en el Listado 2.

Los códigos de carga para los demás nodos están disponibles en GitLab junto con los demás archivos. La base de datos resultante posee 172086 nodos y 344104 relaciones. Una visualización de algunos nodos y relaciones de la base de datos de grafo se pueden observar en la Figura 2.

Listado 1 Consulta en Cypher que fija una restricción de unicidad para el atributo ID en los nodos Trend

```
trend_constraint = """
   CREATE CONSTRAINT FOR (tr:Trend) REQUIRE tr.ID IS UNIQUE
"""

# Crear la sesion del driver
with driver.session() as session:
   # Hacer ID de trends unico
   session.run(trend_constraint).data()
```

Listado 2 Consulta en Cypher que se encarga de cargar los nodos Trend por lote

```
load_trend_csv = """
   LOAD CSV
     WITH HEADERS
     FROM 'https://docs.google.com/spreadsheets/d/e/2PACX-1vSa2nqJC91MyXVdmmud_pBRCUgZ4bH9QenG-1vo_typn0OY9u6PLv0-
          rnx3au52J0UYbLZsNySFEisu/pub?output=csv' AS row
   CALL {
     WITH row
     MERGE (m:Trend {ID: row.ID})
          ON CREATE SET m.Year = toInteger(row.Year),
                        m.Month = toInteger(row.Month),
                        m.Day = toInteger(row.Day),
                        m.Value = toFloat(row.Trend),
                        m.Localidad = row.Localidad,
                        m.Tipo = row.Kind,
                        m.Keyword = row.Keyword
   } IN TRANSACTIONS
   RETURN count(*)
"""

# Crear la sesion del driver
with driver.session() as session:
   # Cargar el archivo CSV
   session.run(load_trend_csv).data()
```

II-D. Consultas

Las consultas que se propuso implementar sobre la base de datos de grafo acompañaban los objetivos planteados antes: evaluar si las palabras clave funcionaban como predictores y utilizarlas para anticipar la existencia de brotes. Sobre ello, se plantearon las siguientes consultas:

1. Calcular la correlación entre los volúmenes de búsqueda web de Google y los casos activos reportados por departamento.
2. Obtener las palabras clave que guardan mayor correlación con los casos activos de COVID-19.
3. Devolver, para períodos delimitados de tiempo qué localizaciones manifestaron incrementos en los volúmenes de búsqueda de las palabras clave válidas.

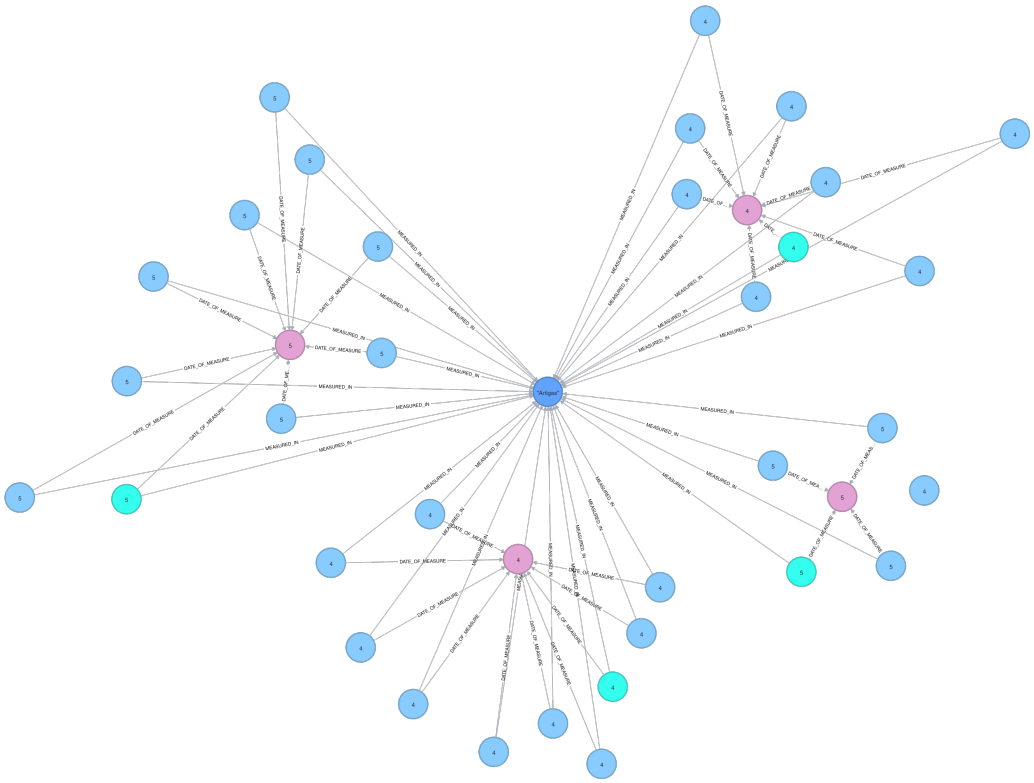

Figura 2: Vista de la base de datos cargada, en la misma se muestra un nodo Localidad (azul al centro), cuatro nodos Date (violeta) y varios nodos Trends (celeste) y CasosActivos (cian).

4. Fijando la localización, ver en qué momento se presentaron aumentos de búsqueda de palabras clave válidas y si estaban asociados a brotes de COVID-19.
5. Explorar las consultas anteriores con combinaciones de más de una palabra clave válida, para evaluar si aumentan el poder de detección.
6. Con las consultas de noticias (*news*), descartar los incrementos de búsquedas web que coincidan con un aumento en búsquedas de noticias (ya que se asumió que fue un aumento de interés en la población lo que aumentó las búsquedas). En caso de ser útil, construir un nuevo valor temporal de búsqueda para cada localización considerando lo anterior y comparar su poder de detección contra los datos de COVID-19.

La primer consulta se implementó de dos formas: una que comparaba los valores a medida que ocurrían (sin desfase) y otra que consideraba un desfase temporal de 7 días entre las búsquedas en google y los casos activos. El motivo del segundo abordaje se basó en la creencia de que las personas tienden a buscar sus síntomas a medida que aparecen, y esto ocurre antes de que vayan al médico por un examen de COVID-19. El texto de la consulta puede verse en los Listados 3 y 4. La correlación cruzada se calculó de forma normalizada de acuerdo a la Ecuación 1, tomando τ como nulo en la primer consulta, y como 7 en la segunda, siendo g las series temporales de valores de búsqueda web en Google.

$$NCC(\tau) = \frac{1}{n\sigma_f\sigma_g}\sum_t f(t)g(t+\tau) \tag{1}$$

Listado 3 Consulta en Cypher que calcula el valor de correlación cruzada normalizado entre los valores de los nodos Trend (de tipo web) y los nodos CasosActivos

```
query1 = """
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(t:Trend {Tipo: 'Web'})-[:MEASURED_IN]->(:Localidad)<-[:MEASURED_IN]-(c:
        CasosActivos)-[:DATE_OF_MEASURE]->(d)
   WITH t.Localidad AS Loc, t.Keyword AS KW, AVG(t.Value) AS avg_t, AVG(c.Value) AS avg_c, stDev(t.Value) AS std_t,
        stDev(c.Value) AS std_c
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(t:Trend {Tipo: 'Web', Keyword: KW})-[:MEASURED_IN]->(:Localidad {Codigo: Loc})
        <-[:MEASURED_IN]-(c:CasosActivos)-[:DATE_OF_MEASURE]->(d)
   WITH Loc, KW, t, c, avg_t, avg_c, std_t, std_c
   RETURN Loc, KW, std_t, std_c, count(t), (sum(t.Value*c.Value)/(std_t*std_c*count(t))) AS NCC
   ORDER BY Loc, KW
"""

# Create the driver session
```

```
with driver.session() as session:
   # Run the Cypher query
   corr = session.run(query1).data()
```

Listado 4 Consulta en Cypher que calcula el valor de correlación cruzada normalizado entre los valores de los nodos Trend (de tipo web) y los nodos CasosActivos. Se consideró un desfase de 7 días

```
query2 = """
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(t:Trend {Tipo: 'Web'})-[:MEASURED_IN]->(:Localidad)<-[:MEASURED_IN]-(c:
        CasosActivos)-[:DATE_OF_MEASURE]->(d)
   WITH t.Localidad AS Loc, t.Keyword AS KW, AVG(t.Value) AS avg_t, AVG(c.Value) AS avg_c, stDev(t.Value) AS std_t,
        stDev(c.Value) AS std_c
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(t:Trend {Tipo: 'Web', Keyword: KW})-[:MEASURED_IN]->(:Localidad {Codigo: Loc})
        <-[:MEASURED_IN]-(c:CasosActivos)-[:DATE_OF_MEASURE]->(:Date {Year: toInteger((date({year: d.Year, month: d.
        Month, day: d.Day})+duration({days: 7})).year), Month: toInteger((date({year: d.Year, month: d.Month, day: d.
        Day})+duration({days: 7})).month), Day: toInteger((date({year: d.Year, month: d.Month, day: d.Day})+duration({
        days: 7})).day)})
   WITH Loc, KW, t, c, avg_t, avg_c, std_t, std_c
   RETURN Loc, KW, std_t, std_c, count(t), (sum(t.Value*c.Value)/(std_t*std_c*count(t))) AS NCC
   ORDER BY Loc, KW
"""

# Create the driver session
with driver.session() as session:
   # Run the Cypher query
   corr2 = session.run(query2).data()
```

La siguiente consulta propuesta se realizó sobre la base de las primeras dos, pero solicitando que la salida comprendiera únicamente aquellas palabras claves con un valor de NCC mayor a 0.25 (elegido de forma arbitraria). El texto de la misma puede observarse en el Listado 5.

Listado 5 Consulta en Cypher que devuelve las palabras claves que presentaron una mayor relación con el incremento de casos activos. Se separan los resultados de acuerdo a la localidad donde ocurrió la coincidencia

```
query2 = """
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(t:Trend {Tipo: 'Web'})-[:MEASURED_IN]->(:Localidad)<-[:MEASURED_IN]-(c:
        CasosActivos)-[:DATE_OF_MEASURE]->(d)
   WITH t.Localidad AS Loc, t.Keyword AS KW, AVG(t.Value) AS avg_t, AVG(c.Value) AS avg_c, stDev(t.Value) AS std_t,
        stDev(c.Value) AS std_c
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(t:Trend {Tipo: 'Web', Keyword: KW})-[:MEASURED_IN]->(:Localidad {Codigo: Loc})
        <-[:MEASURED_IN]-(c:CasosActivos)-[:DATE_OF_MEASURE]->(d)
   WITH Loc, KW, t, c, avg_t, avg_c, std_t, std_c
   WITH Loc, KW, std_t, std_c, count(t) AS N, (sum(t.Value*c.Value)/(std_t*std_c*count(t))) AS NCC
   WHERE NCC > 0.25
   RETURN Loc, KW, NCC
   ORDER BY Loc, KW
"""

# Create the driver session
with driver.session() as session:
   # Run the Cypher query
   validos = session.run(query2).data()
```

La consulta del item 3 consiste en devolver qué localizaciones manifestaron incrementos de búsqueda en un tiempo determinado y ver si esto se relaciona con un aumento de los casos activos en dichos departamentos. El texto de la consulta se muestra en el Listado 6 mientras que los resultados obtenidos se analizan en la Sección III. Para la misma se consideró un período de un mes entre el 15 de octubre y el 15 de noviembre de 2020, calculando cuántos días tuvieron búsquedas no nulas y considerando eso la ”fuerza”de la predicción (se pidió que alcanzara al menos un cuarto de los días totales para ser considerado un brote). Podría cambiarse a cualquier otro período, incluso a un más reducido. En fecha de inicio se incluyó el primer día no negativo de la serie, aunque no se restringió a que los días con búsquedas positivas debieran ser consecutivos.

Listado 6 Consulta en Cypher que devuelve los nombres de las localidades que presentan aumentos en las búsquedas de la palabra clave 'dolor' para un período de un mes

```
query3 = """
   MATCH (y:Year {Year: 2020})<-[:YEAR_OF_DATE]-(d:Date)
   WHERE d.Month IN [10,11]
   MATCH (d)<-[:DATE_OF_MEASURE]-(t: Trend {Tipo:'Web', Keyword:'dolor'})
   WHERE (date({year: y.Year, month: 10, day:15})<date({year: y.Year, month: d.Month, day:d.Day})<date({year: y.Year,
        month: 11, day:15})) AND t.Value>0
   MATCH (t)-[:MEASURED_IN]->(l:Localidad)
   WITH
     min(date({year: y.Year, month: d.Month, day:d.Day})) AS fecha_inicio,
     l.Nombre AS foco,
     count(t.Value) AS fuerza
   WHERE fuerza > 7
   RETURN foco, fecha_inicio, fuerza
```

```
   ORDER BY fecha_inicio
"""

# Create the driver session
with driver.session() as session:
   # Run the Cypher query
   salida3 = session.run(query3).data()
```

La cuarta consulta propuesta pedía fijar la localización (y eventualmente la palabra clave) y devolver las fechas donde podrían haberse presentado brotes en función del aumento de los niveles de búsqueda de cierta palabra clave. Para abordar este problema se consideraron dos consultas: una que compactaba la cantidad de fechas a aquellas que habían registrado búsquedas no negativas en los 10 días siguientes, y otra consulta que calculaba las diferencias entre un día y el siguiente, y se quedaba únicamente con aquellos que registraban un incremento de búsquedas. Cada una de las consultas se muestra en los Listados de código 7 y 8.

Listado 7 Consulta en Cypher que devuelve las fechas donde se registró un número de búsquedas no nulo durante 10 días consecutivos para la palabra clave 'dolor' en el departamento de Canelones

```
query4 = """
   MATCH (t:Trend {Tipo:'Web', Keyword:'dolor'})-[:MEASURED_IN]->(l:Localidad {Codigo: 'UY-CA'})<-[:MEASURED_IN]-(t2:
        Trend {Tipo:t.Tipo, Keyword:t.Keyword})
   WHERE t.Value>0 AND t2.Value>0 AND (date({year: t.Year, month: t.Month, day: t.Day})<date({year: t2.Year, month: t2.
        Month, day: t2.Day}) <= date({year: t.Year, month: t.Month, day: t.Day})+duration({days:10}))
   WITH date({year: t.Year, month: t.Month, day: t.Day}) AS inicio_foco,
      count(t) AS fuerza
   WHERE fuerza > 9
   RETURN inicio_foco, fuerza
   ORDER BY inicio_foco
"""

# Create the driver session
with driver.session() as session:
   # Run the Cypher query
   salida4 = session.run(query4).data()
```

Listado 8 Consulta en Cypher que devuelve las fechas, localidades y palabras clave donde se registró un aumento en el número de búsquedas con respecto al día anterior

```
# Solo los que retornan true (porque son muchos datos y asi se puede ver que existen datos con resultados validos)
query8 = """
   MATCH (loc:Localidad {Codigo: 'UY-MO'})<-[:MEASURED_IN]-(t1:Trend)-[:DATE_OF_MEASURE]->(d1:Date)
   MATCH (loc)<-[:MEASURED_IN]-(t2:Trend)-[:DATE_OF_MEASURE]->(d2:Date)
   WHERE t1.Tipo = 'Web' AND t1.Keyword = t2.Keyword
        AND date({year: d1.Year, month: d1.Month, day: d1.Day}) = date({year: d2.Year, month: d2.Month, day: d2.Day}) -
             duration({days: 1})
        AND t2.Value > t1.Value
        AND t2.Keyword IN ['tos', 'olor', 'gusto', 'dolor', 'fiebre', 'covid-19']
   WITH DISTINCT date({year: t1.Year, month: t1.Month, day: t1.Day}) AS TrendDateAnterior,
               date({year: t2.Year, month: t2.Month, day: t2.Day}) AS TrendDateActual,
               t2.Keyword AS Keyword,
               loc.Nombre as Localidad,
               t1.Value AS TrendDiaAnterior,
               t2.Value AS TrendDiaSiguiente
   ORDER BY TrendDateActual, TrendDateAnterior
   OPTIONAL MATCH (loc)<-[:MEASURED_IN]-(c:CasosActivos)-[:DATE_OF_MEASURE]->(d:Date)
   WHERE date({year: d.Year, month: d.Month, day: d.Day}) >= TrendDateAnterior AND
        date({year: d.Year, month: d.Month, day: d.Day}) < TrendDateActual + duration({days: 2}) // Dos dias de
             tolerancia.
   WITH TrendDateAnterior, TrendDateActual, Keyword, Localidad, TrendDiaAnterior, TrendDiaSiguiente,
         CASE WHEN COUNT(c) > 0 THEN true ELSE false END AS AssociatedWithCovid
   WHERE AssociatedWithCovid = true
   RETURN TrendDateAnterior, TrendDateActual, Keyword, Localidad, TrendDiaAnterior, TrendDiaSiguiente,
        AssociatedWithCovid
"""

# Create the driver session
with driver.session() as session:
   # Run the Cypher query
   salida8 = session.run(query8).data()
```

La propuesta indicada en 5 quedó sin realizar por considerarse que las consultas previas no estaban lo suficientemente resueltas como para continuar añadiendo palabras clave. Por último, para la consigna propuesta en el item 6 del listado se evaluaron dos formas de explorarla, de las cuales se implementó sólo una. La primera forma, que fue finalmente descartada, consistía en crear un nuevo valor de búsquedas que tomara las búsquedas web y le restara las búsquedas de noticias. Esto presentaba inconvenientes delde el punto de vista de la interpretación porque: no tiene sentido obtener valores de búsqueda negativos y además las búsquedas no son comparables entre distintos tipos (web y noticias). El motivo de la imposibilidad de comparar radica en que para construir las métricas de *Trend* se consideran muestras en una población y luego se lo normaliza

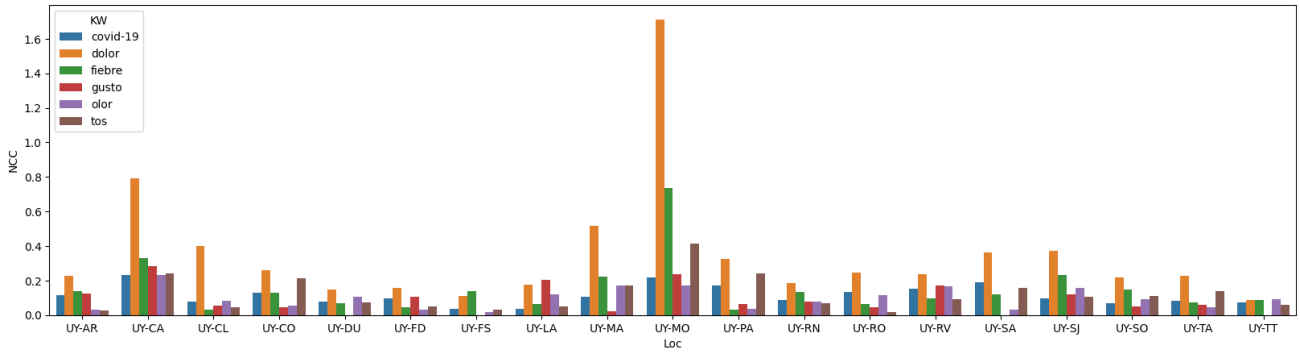

Figura 3: Correlación cruzada normalizada entre las búsquedas web y los casos activos para cada departamento y palabra clave.

a una escala porcentual, perdiéndose idea del volumen que representa cada una de ellas. Se eligió, en cambio, un segundo abordaje que también presenta falencias pero facilita los cálculos (y permite tener al menos un indicador donde se pese las búsquedas de noticias en el resultado final). En éste, se consideraron las búsquedas web y de noticias por separado, calculando un valor de NCC para cada una, y finalmente restándolos para obtener un indicador final. El texto de la consulta puede verse en el Listado 9.

Listado 9 Consulta en Cypher que pondera la relación entre las búsquedas web y los casos activos mediante el uso de las búsquedas de noticias

```
query6 = """
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(tw:Trend {Tipo: 'Web'})-[:MEASURED_IN]->(l:Localidad)<-[:MEASURED_IN]-(c:
        CasosActivos)-[:DATE_OF_MEASURE]->(d),
        (d)<-[:DATE_OF_MEASURE]-(tn:Trend {Tipo: 'News', Keyword: tw.Keyword})-[:MEASURED_IN]->(l)
   WITH l.Codigo AS Loc, tw.Keyword AS KW, stDev(tw.Value) AS std_tw, stDev(c.Value) AS std_c, stDev(tn.Value) AS
        std_tn
   MATCH (d:Date)<-[:DATE_OF_MEASURE]-(tw:Trend {Tipo: 'Web', Keyword: KW})-[:MEASURED_IN]->(l:Localidad {Codigo: Loc})
        <-[:MEASURED_IN]-(c:CasosActivos)-[:DATE_OF_MEASURE]->(:Date {Year: toInteger((date({year: d.Year, month: d.
        Month, day: d.Day})-duration({days: 7})).year), Month: toInteger((date({year: d.Year, month: d.Month, day: d.
        Day})-duration({days: 7})).month), Day: toInteger((date({year: d.Year, month: d.Month, day: d.Day})-duration({
        days: 7})).day)}),
        (d)<-[:DATE_OF_MEASURE]-(tn:Trend {Tipo: 'News', Keyword: KW})-[:MEASURED_IN]->(l)
   WITH Loc, KW, tw, c, tn, std_tw, std_c, std_tn
   RETURN Loc, KW, std_tw, std_c, std_tn, count(tw), count(tn), (sum(tw.Value*c.Value)/(std_tw*std_c*count(tw))) AS
        NCCw, (sum(tn.Value*c.Value)/(std_tn*std_c*count(tn))) AS NCCn, ((sum(tw.Value*c.Value)/(std_tw*std_c*count(tw)
        )) - (sum(tn.Value*c.Value)/(std_tn*std_c*count(tn)))) AS NCCdif
   ORDER BY Loc, KW
"""

# Create the driver session
with driver.session() as session:
   # Run the Cypher query
   corr3 = session.run(query6).data()
```

III. Resultados

Se ejecutaron las consultas alternativas dadas en los Listados 3 y 4 para luego graficar los valore de NCC segregados por departamento y palabra clave. Los resultados dados por cada consulta pueden observarse en las Figuras 3 y 4, mientras que la diferencia relativa entre los resultados de una y otra consulta se recopilan en la Figura 5. Como puede verse en la última figura, no parece observarse una mejora uniforme en los valores de correlación (ya sea para la misma localidad o para la misma palabra clave) por lo que a partir de esta consulta se decidió continuar trabajando sin desfase temporal (por simplicidad).

Algo interesante a notar de estos resultados es que parece observarse una mayor correlación para ciertas palabras clave en las localidades más pobladas. Además, de acuerdo a la consulta del Listado 5, las palabras claves que guardaban mayor correlación con los casos activos reportados fueron ”dolor”, superando el umbral en 8 departamentos, ”fiebre”, con alta correlación tanto para Montevideo como Canelones, ”tos”, con alta correlación para Montevideo así como ”gusto”para Canelones. En las consultas siguiente, se decide utilizar como palabra clave válida para la búsqueda de brotes de COVID-19 a ”dolor”.

La salida de la tercer consulta arrojó siete localidades donde se identificaba un brote de enfermedad a partir de la palabra clave ”dolor.[en] el período comprendido entre el 15 de octubre y el 15 de noviembre. Si se compara estos resultados con los devueltos por la consulta del Listado 10, se puede ver que existían nueve departamentos con al menos cinco casos activos durante siete días en ese intervalo de tiempo. La comparación de las salidas de uno y otro puede verse en la Figura 6, estando a la izquierda lo predicho y al centro y derecha los valores reales (para cinco u once casos activos mínimos). En general, tiene mayor problema anticipando brotes en los departamentos donde la correlación entre las palabras clave y los brotes es baja, o donde se dieron pocos casos.

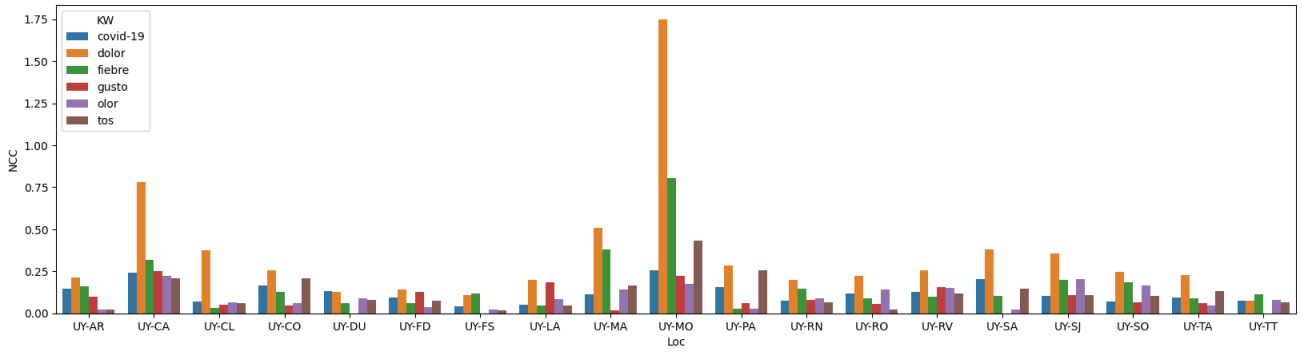

Figura 4: Correlación cruzada normalizada entre las búsquedas web y los casos activos para cada departamento y palabra clave, teniendo en cuenta un desfase de 7 días entre que se realizan las búsquedas y ocurren las infecciones.

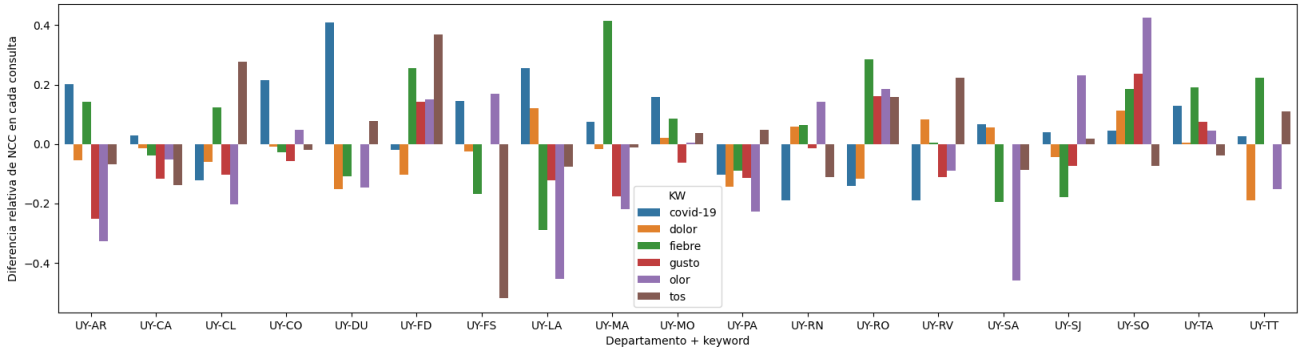

Figura 5: Diferencia relativa entre los valores de correlación cruzada alcanzados en la consulta sin desfase temporal y la consulta con desfase de 7 días. Se muestran los resultados segregados por departamento y palabra clave para facilitar la comparación.

Listado 10 Consulta en Cypher para buscar las localidades que presenten al menos cinco casos activos por siete días entre el 15 de ocutbre de 2020 y el 15 de noviembre del mismo año.

```
query5 = """
   MATCH (y:Year {Year: 2020})<-[:YEAR_OF_DATE]-(d:Date)
   WHERE d.Month IN [10,11]
   MATCH (d)<-[:DATE_OF_MEASURE]-(c: CasosActivos)
   WHERE (date({year: y.Year, month: 10, day:15})<date({year: y.Year, month: d.Month, day:d.Day})<date({year: y.Year,
        month: 11, day:15})) AND c.Value>4
   MATCH (c)-[:MEASURED_IN]->(l:Localidad)
   WITH
     min(date({year: y.Year, month: d.Month, day:d.Day})) AS fecha_inicio,
     l.Nombre AS foco,
     count(c.Value) AS fuerza
   WHERE fuerza > 7
   RETURN foco, fecha_inicio, fuerza
   ORDER BY fecha_inicio
"""

# Create the driver session
with driver.session() as session:
   # Run the Cypher query
   salida5 = session.run(query5).data()
```

Para la cuarta consulta, donde se deseaba fijar una localidad y palabra clave para obtener las fechas de los brotes, el Listado 7 devolvió 29 fechas, muchas de las cuales eran contiguas o cercanas, lo que da indicios de que sería posible refinar más los resultados. Las primeras doce fechas dadas por la consulta se muestran en la Figura 7, a golpe de vista parecería encontrar grandes brotes ocurridos a nivel país, no obstante, para el caso de Canelones, si bien se observan ciertas fluctuaciones en las fechas dadas para el mes 6 y 10 de 2020, son etapas bien tempranas de brotes o se está en el medio de ellos (ya que la curva no vuelve a 0 en 2020). En general, se considera que esta consulta es más compleja que la anterior (al modelar qué es un brote en el tiempo) y necesita más trabajo. Respecto a la salida del Listado 8, el mismo devolvía más de 700 combinaciones de fechas y palabras clave que mostraban un incremento en los datos al mismo tiempo que existían casos activos en esa localidad y fecha. Sería interesante seguir construyendo sobre esta consulta y evaluar en mayor profundidad qué tan significativo es un incremento en la búsqueda con respecto a la fluctuación de casos activos.

Por último, al añadir los datos de búsqueda de noticias y restar las correlaciones cruzadas de ambos tipos, se obtiene la

	foco	fecha_inicio	fuerza
0	Montevideo	2020-10-16	30
1	Canelones	2020-10-16	25
2	Treinta y Tres	2020-10-16	8
3	Colonia	2020-10-17	12
4	Salto	2020-10-18	8
5	Maldonado	2020-10-18	10
6	Tacuarembo	2020-10-26	10

	foco	fecha_inicio	fuerza
0	Rivera	2020-10-16	30
1	Montevideo	2020-10-16	30
2	Canelones	2020-10-16	30
3	Colonia	2020-10-16	20
4	Cerro Largo	2020-10-16	15
5	Tacuarembo	2020-10-18	15
6	Artigas	2020-10-24	11
7	San Jose	2020-10-28	11
8	Maldonado	2020-10-29	10

	foco	fecha_inicio	fuerza
0	Rivera	2020-10-16	30
1	Montevideo	2020-10-16	30
2	Canelones	2020-10-16	30
3	Colonia	2020-10-16	16

Figura 6: Comparación de salidas para la consulta 3. A la izquierda se observa la predicción dada por las búsquedas web, al centro las localidades con casos activos que cumplen los criterios de tener al menos cinco infectados durante siete días y a la derecha las localidades con casos activos y al menos diez infectados en siete días del intervalo.

	inicio_foco	fuerza
0	2020-04-03	10
1	2020-05-31	10
2	2020-06-01	10
3	2020-06-02	10
4	2020-06-03	10
5	2020-06-04	10
6	2020-10-25	10
7	2020-10-26	10
8	2020-10-27	10
9	2021-01-02	10
10	2021-01-03	10
11	2021-01-04	10

Figura 7: Salidas para la consulta 4.1. Puede verse que las fechas encontradas como inicio del foco en algunos casos son cercanas entre sí, porque no logra separar focos extensos temporalmente de varios focos consecutivos (dado que sólo se miran diez días).

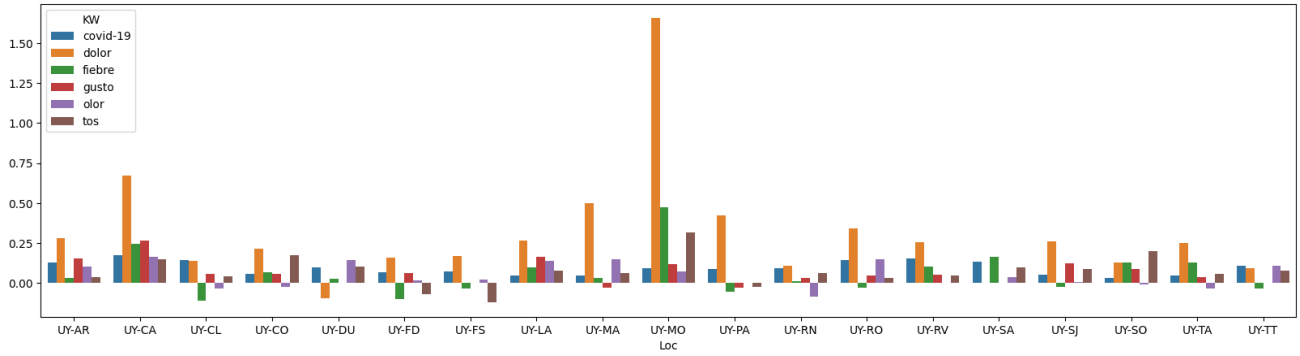

Figura 8: Diferencia de NCC con los casos activos, entre las búsquedas web y las búsquedas de noticias para cada palabra clave y localidad

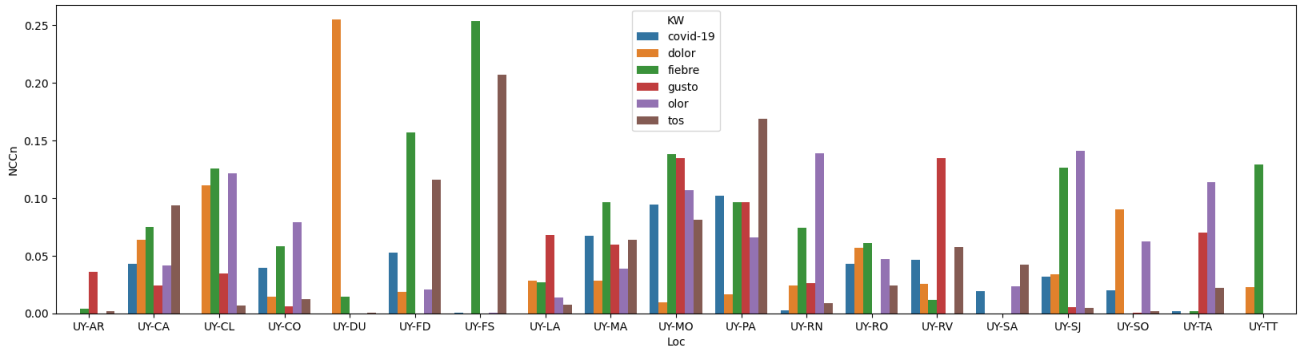

Figura 9: Valor de NCC para cada departamento y palabra clave al evaluar la correlación cruzada entre las búsquedas de noticias y los casos activos.

Figura 8. Como puede verse, de forma similar a lo sucedido para la primer consulta y el desfase, la consideración o no de las búsquedas de noticias no parece arrojar muchas diferencias en la correlación de cada palabra clave, a excepción del término ”covid-19”que parecía estar más relacionado con búsquedas de noticias que con búsquedas de síntomas en la web (algo que se esperaba que ocurriera). Las tendencias de cada palabra clave para la búsqueda de noticias puede observarse en la Figura 9.

IV. Conclusiones y trabajo futuro

Se logró obtener y cargar los datos de *Google Trends* y casos activos de COVID-19 en Uruguay, en una base de datos de grafos. En el proceso se debieron sortear limitaciones de elementos (nodos, relaciones) a causa de estar utilizando una instancia gratuita en Neo4j AuraDB.

El diseño de la base de datos permitió realizar la mayoría de las consultas planificadas, aunque se obtuvieron algunos resultados ambiguos (por ejemplo, al buscar brotes de COVID-19 dentro de una localidad específica).

Como aspecto a tener en cuenta, si bien las palabras claves seleccionadas guardaban relación con síntomas de COVID-19 que cualquier individuo podría querer buscar, éstas búsquedas pueden ser manifestaciones de brotes de otras enfermedades, como por ejemplo la influenza. Adicionalmente, el trabajo de detección y seguimiento de casos y contactos de COVID-19 realizado durante la pandemia podría haber disminuido el volumen de consultas. Por ejemplo, si a los individuos se les informa que tuvieron contacto con un infectado antes de que manifiesten síntomas, es probable que no busquen los mismos en la web en busca de respuestas sobre a qué corresponden sus síntomas.

Sería interesante probar con diseños alternativos que permitan, por ejemplo, conectar los nodos Date entre sí (en orden de ocurrencia), y evaluar si esto sirve de ayuda al momento de realizar la consulta 4. Adicionalmente, sería interesante continuar trabajando en el añadido de nuevas palabras clave, para evaluar predictores armados a partir de combinaciones de ellas.

REFERENCIAS

[1] Jaime Cerda and Gonzalo Valdivia. John snow, la epidemia de cólera y el nacimiento de la epidemiología moderna. *Revista chilena de infectología*, 24(4):331–334, 2007.

[2] Marcelo Fiori, Nicolás Wschebor, Ernesto Mordecki, and Federico Lecumberry. Reporte 12: Relación entre movilidad y tasa de reproducción: Guiad-covid-19, Jun 2021.

[3] Herman Anthony Carneiro and Eleftherios Mylonakis. Google Trends: A Web-Based Tool for Real-Time Surveillance of Disease Outbreaks. *Clinical Infectious Diseases*, 49(10):1557–1564, 11 2009.

[4] Loukas Samaras, Miguel-Angel Sicilia, and Elena García-Barriocanal. Predicting epidemics using search engine data: a comparative study on measles in the largest countries of europe. *BMC Public Health*, 21(1):1–14, 2021.

[5] U Venkatesh and Periyasamy Aravind Gandhi. Prediction of covid-19 outbreaks using google trends in india: A retrospective analysis. *Healthcare informatics research*, 26(3):175–184, 2020.

[6] Repositorio de gitlab: códigos de trabajo final. https://gitlab.fing.edu.uy/lucia.lemes/proyectofinal-bdnr.

[7] Bin Feng, Qing Zhu, Mingwei Liu, Yun Li, Junxiao Zhang, Xiao Fu, Yan Zhou, Maosu Li, Huagui He, and Weijun Yang. An efficient graph-based spatio-temporal indexing method for task-oriented multi-modal scene data organization. *ISPRS International Journal of Geo-Information*, 7(9):371, 2018.