Análisis de texto con n-grams. Relaciones entre discursos presidenciales y cotizaciones de divisas

Jonathan Ruben Loscalzo
Estudiante Licenciatura en Sistemas, Facultad de Informática
Universidad Nacional de La Plata.
50 y 120 – La Plata, Argentina.
jonathan.r.loscalzo@gmail.com

Javier Díaz
Director
Alejandra Osorio y Ana Paola Amadeo
Tutoras
Laboratorio de Investigación en Nuevas Tecnologías Informáticas.
Facultad de Informática. Universidad Nacional de La Plata.
50 y 120 – La Plata, Argentina.
{jdiaz, aosorio}@unlp.edu.ar,
pamadeo@linti.unlp.edu.ar

Resumen. El presente artículo describe el desarrollo realizado como trabajo final de en una materia de análisis de datos, a partir de la consigna del uso de datos abiertos y su integración con distintas fuentes de datos para la toma de decisiones. A partir de la hipótesis sobre que los discursos presidenciales afectan el precio de la divisa estadounidense, este trabajo presenta los datos utilizados, la metodología y las herramientas utilizadas y los resultados obtenidos tomando los discursos presidenciales disponibles en el portal de la casa Rosada y las variaciones del dólar disponibles en el portal BankerSalgo[3]. Se presenta el desarrollo para la obtención de los datos, el procesamiento, la API para ofrecer los datos y el portal de navegación de la solución.

1 Introducción

Durante el año 2018, la materia optativa de la Facultad de Informática de la Universidad Nacional de La Plata relacionada análisis de datos, Tecnologías aplicadas a Business Intelligence, a cargo de Paola Amadeo y Alejandra Osorio, coordinada por Javier Díaz en el Laboratorio de Nuevas Tecnologías Informáticas LINTI, planteó como trabajo final para la aprobación de la materia, una solución que permitiera integrar datos abiertos a nivel nacional, en cualquier jurisdicción, con datos provenientes de otras fuentes de información a fin de visualizar, en forma conjunta, los datos obtenidos. Esta visualización debería permitir a un usuario final realizar distintas consultas para facilitar la

toma de decisiones. Los discursos presidenciales disponibles en el portal de la Casa Rosada [1], junto con un informe del sitio web de Chequeado como ejemplo y material disponible con licencia CC [2] fueron una de las opciones disponibles para trabajar. A partir de esta consigna, surgió que en distintos medios de comunicación nacionales hacían referencia a que, durante el gobierno de Mauricio Macri, el precio de la divisa dólar entraba en alza cada vez que él mismo realizaba un discurso en una presentación en público o en cadena nacional. Resulto interesante entonces visualizar de una manera descriptiva, los días de los distintos discursos y los precios del dólar histórico desde el comienzo de su gobierno. También observar las relaciones entre los mismos analizando, de cada uno de ellos, las ocurrencias de las palabras.

En la siguiente sección se describe la metodología utilizada y los principales problemas y soluciones. En la sección 3 se describen las herramientas, la mayoría de ellas de código abierto, empleadas y finalmente en la sección 4 se presentan imágenes y gráficos con los principales resultados obtenidos.

2 Implementación de la solución

2.1 Desarrollo

La recopilación de la información fue una de las principales dificultades para la realización del trabajo. Los precios históricos del dólar no se encontraban ofrecidos por muchos sitios a través de una API de monedas [3], y los discursos sólo se encontraban en la página oficial de Casa Rosada. A partir de allí, se realizó un preprocesamiento para obtener distintas estadísticas: cantidad de palabras, ngrams; por último, una web API[4] para ofrecer los datos, y distintas visualizaciones. Los ngrams[5] son una subsecuencia dentro de una secuencia más general, en nuestro caso serían los ngrams de 5, 6 y 7 palabras para cada discurso.

En el caso de la cotización del dólar, se realizó un script en Javascript para obtener iterativamente todas las cotizaciones diarias desde el inicio del mandato el día 12 de diciembre del 2015 hasta el 1 de diciembre de 2018, día por día. La persistencia se realizó en una base de datos MongoDB en la nube [6].

Para la obtención de los discursos, se realizó la extracción de información desde el sitio oficial de Casa Rosada. Para realizar esta tarea se utilizó una técnica conocida como Scraping. Usualmente, estas técnicas simulan la navegación de un ser humano, y van recolectando información, transformándola para su posterior almacenamiento y procesamiento. Esta simulación se puede realizar con distintas herramientas y librerías que tienen este propósito, como pueden ser scrapy (Python), rvest (R lang), etc.

Esta tarea fue realizada en dos etapas: obtención de los links de cada discurso, y posteriormente la obtención de cada cuerpo de discurso. El primer paso consistió en conocer como indexaba los discursos la página oficial. Se analizaron las URL's que generaba el sitio mientras se navegaban las páginas. Se pudo observar el siguiente patrón:

Página	URL

1	https://www.casarosada.gob.ar/informacion/discursos
2	https://www.casarosada.gob.ar/informacion/discursos?start=40
3	https://www.casarosada.gob.ar/informacion/discursos?start=80

El patrón encontrado fue evidente, se mostraban 40 discursos por página, y se indexaba en la misma cantidad. Una parte del script generado fue la siguiente:

```
// Función que retorna los links de una pág.
const getLinks = async (leaf = 1) => {
    let skip = (leaf - 1) * 40; // le indica la pág.
    const browser = await puppeteer.launch();
    const page = await browser.newPage();
    await page.goto(`${url}${skip}`); //navega la pág
    const links = await page.evaluate((sel) => {
        let htmlCollection = document
          .getElementsByClassName(sel);
        return Array
          .from(htmlCollection)
          .map(a => a.getAttribute("href"));
    }, box); // Obtiene la url
    await browser.close();
    return links; // retorna todos los links de la página
};
…
// función que itera por todas las páginas desde 1 al 13
async function all_links() {
    let links = []
    await asyncForEach(range(1, 13), async (key) => {
        let items = await getLinks(key) // Func anterior
        links = links.concat(items)
    })
    return links; // acumula todos los links
}
```

Esta tarea fue realizada con una herramienta prácticamente nueva, desarrollada por Google, llamada Puppeteer[7]. Posterior a la obtención de los links de cada discurso, se realizó el scraping de cada uno, en este caso eran 505 discursos. En este volumen, la performance se vio degradada para realizar la tarea completa (abrir una pestaña, obtener el discurso, cerrar la pestaña, etc.); además el script no era resiliente en caso de falla: demoraba mucho tiempo y si fallaba se debía comenzar todo el proceso nuevamente, sin posibilidades de continuar desde el punto de falla o detectar el problema. La idea era que el script sea performante y resiliente al mismo tiempo, por esto se utilizó Puppeteer con un adicional para realizar procesamiento de manera clusterizada. Esto permitió disminuir el tiempo de la solución de aproximadamente 1 hora a 15 minutos. Además, se agregó un logging para verificar en particular que discurso fallaba y continuar con la ejecución.

Posterior a la extracción de información, fue trabajada para obtener valor de la misma y poder realizar algún tipo de visualización y estadística. Se analizó la posibilidad de utilizar una librería particular de Python llamada NLTK, pero nos decantamos por la solución distribuida. Para ello, se utilizó un framework de procesamiento distribuido conocido como Spark, es su versión para el lenguaje Python (PySpark)[8]. Se realizaron transformaciones sobre los datos de los discursos para obtener los ngrams y la contabilización de palabras.

Esta tarea, fue la que demandó más tiempo y recursos para su realización, debido a que el procesamiento distribuido se realizaba sólo en una máquina, simulando dos clústeres donde distribuir dicho procesamiento. Para ello, se utilizó Docker[9] como administrador de los clústeres, la coordinación entre ellos se provee por la herramienta.

Lo que se realizó con esta herramienta muy utilizada en el ambiente de Big Data, fue generar un contabilizador de palabras y un contabilizador de ngramas para cada discurso. Suele ocurrir que cierto tipo de tareas de Big Data sean de reducción y transformación de datos, pero en este caso fue aumento de los mismos. En la siguiente tabla se puede observar la cantidad de información inicial, y la cantidad de información generada por el procesamiento:

Table 1. Información almacenada en base de datos

Colección	Cantidad	Tamaño
Exchanges	1,128	171.28 KB
wordcounts_totals	18,613	2.57 MB
ngrams_5_counts	503,507	69.41 MB
ngrams_6_counts	519,192	71.59 MB
ngrams_7_counts	524,855	72.57 MB
Speeches	505	5.07 MB
speeches_stats	505	9.64 MB

Es interesante observar cómo a partir de solo 5mb de información de los discursos, que se encuentran en la colección “Speeches” generamos alrededor de 210 mb de información de los ngrams. Las otras contabilizaciones que tuvimos en cuenta fueron por palabras, en total y por cada discurso. La realización de dicha tarea conlleva la utilización de “Stop words” o “palabras vacías” [10]: no debemos contabilizar palabras que no tienen valor semántico y más bien poseen solo valor sintáctico: “algún, cierto, cual, cualquier, osea, esta”, etc.

```
# código pyspark para inicializar stop-words
stopWords = (StopWordsRemover
  .loadDefaultStopWords("spanish"))

# utilizamos el stop words con todas las palabras
remover = StopWordsRemover(
  inputCol="raw_words",
  outputCol="words",
  stopWords=stopWords)
```

```
# tenemos que usar remover, transformamos
tokenized_by_word = remover.transform(tokenized_by_word)
```

Posterior a este aumento de información, se desarrolló un servicio web codificado en Nodejs para disponibilizar los datos almacenados. El despliegue se realizó en la plataforma Heroku [11], lo que nos permite tener dentro de una cuota gratuita nuestro servicio productivo[14]. En el mismo se realizaron las consultas presentadas en la tabla 2[12]:

Table 2. Recursos disponibilizados por la API

URI	Descripción
/speeches/dollar	Devuelve las cotizaciones del dólar dentro de un rango de fechas.
/speeches/:id	Devuelve el discurso por su identificador: discurso completo, wordcounts.
/stats/word-counts	Devuelve la contabilización de palabras general de todos los discursos
/stats/ngrams	Devuelve los ngrams de 5, 6 y 7 palabras con su contabilidad.
/stats	Estadísticas generales

Para terminar, la visualización se realizó con una web de tipo SPA (es una aplicación web que cabe en una sola página), desarrollada con la librería React. Para las visualizaciones en particular utilizamos Highcharts y Recharts.

En su mayoría, las gráficas son nubes de palabras y para los discursos se provee un formulario para realizar la búsqueda de los datos y poder visualizarlos, al estilo línea de tiempo[12].

Evaluamos otro tipo de solución en donde se podrían implementar las mismas consultas sobre la base de datos por medio de Jupyter Notebooks[13] junto a los gráficos de las librerías que poseen el lenguaje Python, como por ejemplo Matplotlib o Seaborn. Esta solución sería mucho más flexible para un usuario con conocimientos de series temporales y programación; pudiendo realizar consultas y un análisis exploratorio de la información de una manera más flexible y veloz. La inclinación por la idea de la visualización de manera Web, fue para que sea más amigable a usuarios inexpertos en tecnologías, como puede ser los altos mandos de alguna compañía donde solo les interesaría poder realizar búsquedas sobre los indicadores que definan y sacar sus propias conclusiones.

2.2 Herramientas utilizadas

2.2.1 Web Visualización.

Para la web de visualización se utilizaron las siguientes herramientas:

- React y Redux: para el desarrollo del portal Web.
- Highcharts y Recharts: para realizar las gráficas y visualizaciones.
- Axios: para las consultas a la API de datos.

2.2.2 API de datos

Para la API desarrollada se utilizaron las siguientes herramientas:

- Nodejs y Express.js: para realizar la API
- Mongoose: como interfaz para realizar las consultas a la base de datos.
- Lodash para utilidades generales y Moment para utilidades de fechas.

2.2.3 Scraping a Casa Rosada

Para el script de scraping se utilizaron las siguientes herramientas:

- Puppeter y Puppeteer-Cluster: para realizar el scraping
- Nodejs: para scripting
- Winston: para realizar el trace de logs.

2.2.4 Cotizaciones del dólar

Para realizar el script donde se obtuvo los precios del dólar se utilizó un stack semejante al utilizado la API de datos: nodejs, mongoose, axios.

2.2.5 Transformación a Ngrams

Para procesar los datos de los discursos, se utilizó una solución distribuida [16] con las siguientes herramientas:

- Python: como lenguaje de scripting
- Spark y PySpark: para la generación de ngrams y la contabilización de palabras de todos los discursos. Esta herramienta es muy utilizada en el mundo del Big Data.
- Docker: como administrador de los contenedores de Spark, se poseen dos nodos donde uno corresponde a un spark-master y otro a un spark-slave. La distribución de carga se da automáticamente entre los nodos.
- mongo-spark-connector: para realizar la conexión a la base de datos.

2.2.5 Obtención de datos para este trabajo

Las distintas tablas que se muestran en la sección de resultados obtenidos, se pueden observar consultas de variaciones del dólar y contadores de palabras. Para realizar esta tarea se utilizaron las siguientes herramientas.

- Jupyter Notebook**:** para realizar los scripts y obtener los resultados inmediatos.
- Python3**:** como lenguaje de scripting dentro de las notebooks.
- PyMongo: para obtener los datos de la base de datos
- Pandas: como librería soporte de DataFrames.

2.3 Resultados Obtenidos

Como resultado, podemos visualizar en la siguiente tabla, algunas de las palabras que más fueron mencionadas dentro de todos los discursos:

Table 3. Alguna de las palabras más mencionadas

Palabra	Cantidad
Argentina	2521
Argentinos	1839
Trabajo	1621
Mundo	1505
ustedes	1378
Juntos	1029
Verdad	909
Compromiso	458

Tiene sentido que la palabra "argentina" sea la más mencionada de todos los discursos, con un promedio de 5 menciones por discurso. Las otras palabras, suelen ser las más importantes que el gobierno siempre intenta destacar, como una política a seguir: "trabajo", "mundo", "juntos", "verdad", "compromiso".

En la web donde están las visualizaciones, se utilizó una nube de palabras donde la importancia es directamente proporcional al tamaño dentro de la nube, así como su posición con respecto al centro del gráfico.

Con respecto a los ngramas, podemos destacar varios según su longitud que, colocándolos en contexto, parecen ser apropiados por donde se fundó el poder del gobierno y las propuestas que presentaron para su candidatura.

Table 4. Algunos ngrams más destacados

Ngrams	Cantidad
la ciudad de buenos aires	132
y eso es lo que	96
lo que somos capaces de hacer	51
el plan de infraestructura más importante[18]	26
de un día para el otro	56

Podemos destacar que "la ciudad de buenos aires" (132) es una de las frases más mencionadas, ya que el gobierno gestó su camino hacia la presidencia desde esta ciudad. Luego, el Plan Belgrano y otras obras fueron los proyectos sobre los que se hizo mención en el "plan de infraestructura más importante" (26), donde fue mencionado en muchos discursos desde el comienzo, pero se fue atenuando de a poco su aparición. También, otra frase muy mencionada fue como se iban a resolver los problemas del país, sobre el camino a seguir, pues no podían ser "de un día para el otro" (56) y debería ser algo más bien gradual, donde las "soluciones mágicas" no existen. Dando apoyo a los ciudadanos, "lo que somos capaces de hacer" (51) se encuentra entre las más nombradas. Otra frase, que quizás no tiene mucho sentido y es más bien un conector de oraciones: "y es lo que" (96). Este ngram, quizás no tiene un sentido tan figurativo

como los demás, pero hace referencia a un modismo que posee el presidente para explayarse al momento de conectar ideas. En la figura 1,2 y 3 podemos observar los ngrams que se pueden encontrar en el portal desarrollado para el trabajo.

Fig. 1. Ngrams de 5 palabras

Fig. 2. Ngrams de 6 palabras

Fig. 3. Ngrams de 7 palabras

En la comparación de los precios de la cotización del dólar, frente a los discursos que fueron dados por el mandatario, podemos hallar la correspondencia que hay entre los discursos en los días que tuvo mayor variación el precio de la divisa. Para ello, se realizó una búsqueda para encontrar los días que tienen mayor variación positiva y negativa, posteriormente se buscaron los discursos que se encuentran dentro de esos periodos. Para obtener la variación, se realizó el remuestreo de los datos como una serie temporal, donde se computaba el día actual y el día anterior, y se realizaba la sustracción para obtener cuánto varió el precio. Posterior a este hallazgo, se buscó si unos días anteriores y posteriores a esa variación se encontraba un discurso.

La que mayor variación positiva tuvo, fueron las cotizaciones del 28 de Agosto y 29 de Agosto de 2018, donde fue de $ 4,73. En este rango, encontramos un discurso el día 29 de agosto, donde las palabras más mencionadas fueron las que se presentan en la figura 4:

Fig. 4. Palabras más mencionadas del discurso del 29 de Agosto de 2018

Otra variación importante fue la del 25 al 26 de Septiembre de 2018 donde el cambio fue de $1,31. Podemos encontrar el día 26, un discurso en la Asamblea de Naciones Unidas, en dicho discurso, las palabras más mencionadas fueron las que se observan en la figura 5.

Fig. 5. Palabras más mencionadas del discurso del 26 de Septiembre de 2018

Una variación con tendencia a la baja, fue la variación del día 19 y 20 de Septiembre de 2018, donde en ambas fechas hubo inauguraciones de obras, así como también el día 14 de Septiembre hubo otra inauguración. Por lo que podemos ver, aquel mes fue de grandes altibajos para la divisa, ya que también se encuentra dentro de las mayores variaciones positivas, como se puede observar en la figura 6[15].

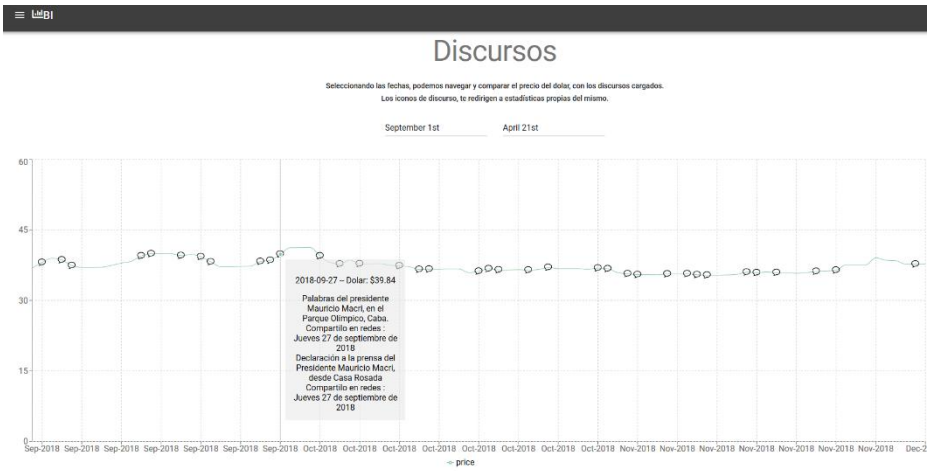

Fig. 6. Tendencia del dólar para el mes de Septiembre de 2018; imagen obtenida desde la web de visualización desarrollada para el trabajo.

La disponibilidad de datos abiertos a través de APIs, la disponibilidad de herramientas que facilitan la integración de distintas fuentes de datos, la ejecución en la nube y definición dinámica de clústers para el procesamiento de Big Data facilitan el procesamiento de los datos para obtener información consistente, en menor tiempo y con un gran volumen de datos. El uso de herramientas de código abierto para la construcción de distintas visualizaciones de acuerdo al usuario final, facilita el análisis integrado y, en este caso, verificar la validez de una hipótesis dada por los medios de comunicación más importantes del país.

3 Referencias

1. Discursos Presidenciales: Página Oficial Casa Rosada. https://www.casarosada.gob.ar/informacion/discursos
2. Chequeado. Artículo relacionado a la investigación, con el cual se inspiró este trabajo. https://chequeado.com/el-explicador/que-tienen-en-comun-mauricio-y-nestor/
3. BankerSalgo: Currency & Cryptocurrency rates API. https://bankersalgo.com/
   Nota: Actualmente el sitio se encuentra retirado. Como solución alternativa para posibles continuaciones del trabajo se optó por recuperar los datos desde el sitio de estadísticas del Banco Central: https://estadisticasbcra.com/api/documentacion
4. Portal de Navegación de visualizaciones y estadísticas: http://bi-jaiio.surge.sh
5. N-grama. Definición. https://es.wikipedia.org/wiki/N-grama
6. M-Lab. Servicio MongoDB en la nube. https://mlab.com/ y próximamente https://www.mongodb.com/cloud/atlas
7. Puppeteer. Herramienta utilizada para scraping. https://github.com/GoogleChrome/puppeteer
8. Spark. Herramienta para procesamiento distribuido. https://spark.apache.org/
9. Spark Container. Imagen de Docker para utilizar spark. https://hub.docker.com/r/gettyimages/spark/

10. Características de Spark para utilizar StopWords: https://spark.apache.org/docs/latest/ml-features.html#stopwordsremover
11. Plataforma como Servicio donde se desplegó la API: http://heroku.com
12. Web de visualización. http://bi-jaiio.surge.sh/
13. Utilizamos de todas maneras Jupyter para realizar algunas consultas la base de datos y poder agregarlas a este trabajo. Los archivos en formato IPython nbreader se encuentran disponibles en la cuenta de github del autor. Se pondrá a disposición en la versión camera-ready y se agrega link en archivo - carátula.
14. Servicio para consumir los datos. https://bi-tp-final.herokuapp.com/
15. En la web de visualización, se pueden observar los discursos presionando click sobre los iconos dentro del gráfico. http://bi-jaiio.surge.sh/Dashboard
16. Código desarrollado en Spark. Disponibles en la cuenta de github del autor. Se pondrá a disposición en la versión camera-ready y se agrega link en archivo – carátula.
17. Chequeado. Frase “plan de infraestructura más importante de la historia”. Encontramos que el sitio Chequeado también encontró esta referencia como importante. https://chequeado.com/ultimas-noticias/macri-vamos-a-poner-en-marcha-el-plan-de-infraestructura-mas-importante-de-la-historia-2018/