

ANÁLISIS SEMÁNTICO LATENTE PARA LA DETECCIÓN DE NOTICIAS FALSAS SOBRE COVID-19 UTILIZANDO COMPUTACIÓN HETEROGÉNEA

Bryam David Vega Moreno¹, Gabriel Alejandro León Paredes²,
David Andrés Morales Rivera³


¹ Universidad Politécnica Salesiana - Grupo de Investigación GIHP4C

E-mail: bvegam1@est.ups.edu.ec

 <https://orcid.org/0000-0003-3122-5846>


² Universidad Politécnica Salesiana - Grupo de Investigación GIHP4C

E-mail: gleon@ups.edu.ec

 <https://orcid.org/0000-0002-4361-8965>

³ Universidad Politécnica Salesiana - Grupo de Investigación GIHP4C

E-mail: dmoralesr1@est.ups.edu.ec

 <https://orcid.org/0000-0002-4963-8991>

Fecha de recepción:
18/08/2020

Fecha de aceptación:
18/09/2020

RESUMEN

La detección de noticias falsas hoy en día es un gran reto para los sistemas de predicción debido a la gran cantidad de información que se tiene actualmente, en especial, en fuentes de información como las redes sociales, blogs o sitios de web. En adición, la capacidad de procesamiento que se requiere para analizar grandes cantidades de datos es muy grande por lo que el tiempo de ejecución tiende a ser alto. En este artículo se propone un sistema de aprendizaje utilizando paradigmas de procesamiento en paralelo a nivel de CPU y GPU usando el dataset COVID-19 Open Research Dataset Challenge (CORD-19) para un primer enfoque a la detección de noticias falsas sobre COVID-19. El sistema de predicción está basado en técnicas de

procesamiento de lenguaje natural utilizando como modelo de entrenamiento el análisis semántico latente o LSA, por sus siglas en inglés. También, se utilizan técnicas de multiprocesamiento a nivel de CPU para el preprocesamiento de texto, obtención de términos o palabras claves, obtención de matriz término por documento, normalización de valores utilizando TF-IDF y obtención de la similitud de coseno, mientras que para la parte de reducción de la dimensionalidad utilizando la descomposición de valores singulares o SVD, por sus siglas en inglés se ha utilizado la arquitectura de CUDA para el procesamiento a nivel de la GPU.

Palabras Clave: *análisis semántico latente, computación heterogénea, Covid-19, procesamiento de lenguaje natural*

ABSTRACT

Detecting false news today is a great challenge for prediction systems due to the large

amount of information currently available, especially, in information sources such as social networks, blogs or websites. In addition, the processing capacity required to analyze large

amounts of data is very large so, the execution time tends to be high. This paper proposes a learning system using parallel processing paradigms at the CPU and GPU level using the COVID-19 Open Research Dataset Challenge (CORD-19) for a first approach to detecting false news about COVID-19. The prediction system is based on natural language processing techniques using latent semantic analysis or LSA as a training model. Also, CPU-level multi-processing techniques are used for text pre-processing,

terms or keywords extraction, term-by-document matrix extraction, value normalization using TF-IDF and cosine similarity extraction, while the dimensionality reduction phase using the Singular Value Decomposition (SVD) is based on the CUDA architecture for GPU-level processing.

Keywords: *latent semantic analysis, heterogeneous computing, Covid-19, natural language processing*

Forma sugerida de citar: Vega Moreno, B., León Paredes, G. & Morales Rivera, D. (2021). Análisis semántico latente para la detección de noticias falsas sobre COVID-19 utilizando computación heterogénea. Convergence Tech Revista Científica. 5(1), 18-29. <https://doi.org/10.53592/convtech.v5iV.14>

INTRODUCCIÓN

Hoy en día el uso de redes sociales y sitios web para encontrar información ha ido creciendo exponencialmente a lo largo del tiempo. Sin embargo, existe una gran cantidad de información falsa hacia ciertos temas de interés (Hlaing and Kham, 2020). Son tal los casos de desinformación, que con la llegada del virus denominado COVID-19 se ha apreciado una gran cantidad de noticias erróneas o falsas que han circulado en las redes sociales, sitios web, blogs, entre otros medios de comunicación.

Para solucionar dicho problema de desinformación se han utilizado conjuntos de algoritmos de procesamiento de lenguaje natural y aprendizaje de máquina que permiten predecir cuándo una noticia es falsa o real, como por ejemplo, las máquinas de vectores de soporte, clasificador de bayes, regresión logística, entre otros (León-Paredes et al., 2017). Asimismo, dentro del área de procesamiento de lenguaje natural encontramos un método estadístico que nos permite encontrar la similitud entre textos para la recuperación de información relevante, dicho método es denominado como Análisis Semántico Latente o LSA, por sus siglas en inglés (Landauer and Dumais, 1997).

Entonces, LSA es un método que permite indexar y recuperar información automáticamente de un conjunto de objetos mediante la técnica de descomposición de valores singulares (SVD, por sus siglas

en inglés). Además, LSA mejora el manejo de palabras polisémicas asumiendo que existe alguna estructura semántica subyacente latente en los datos (León-Paredes et al., 2017). Finalmente, LSA, desde sus principios ha sido considerada como una nueva teoría general de adquisición de similitudes y representación del conocimiento, que es útil para simular el aprendizaje de vocabulario y otros fenómenos psicolingüísticos (Landauer and Dumais, 1997).

No obstante, LSA tiene una complejidad de $O(n^2r^3)$, donde n es el valor más pequeño entre el número de documentos y el número de términos, y r es el número de valores singulares (Zhang et al., 2008). Por lo tanto, LSA toma una cantidad considerable de tiempo para indexar y calcular el espacio semántico cuando se tiene un gran conjunto de datos.

Por consiguiente, en este artículo se propone utilizar técnicas de procesamiento en paralelo a nivel de la CPU y GPU para la creación de un sistema heterogéneo que permita detectar presuntas noticias falsas sobre COVID-19 utilizando el método

LSA. Para la parte de CPU utilizamos técnicas de multiprocesamiento la cual permite el uso de varios núcleos del procesador y la memoria RAM para mejorar los tiempos de ejecución. Para el caso de GPU utilizamos la arquitectura CUDA, la cual nos permitiría trabajar el método de reducción de dimensionalidad usando los cientos de miles de núcleos CUDA disponibles en las tarjetas gráficas NVIDIA.

Asimismo, el sistema abarca la ejecución completa del método LSA, es decir, la creación de la matriz término-documento para describir la frecuencia de los términos para una colección de documentos, la ejecución del TF-IDF para la normalización de la matriz término-documento, la implementación del SVD para reducir la dimensionalidad de la matriz TF-IDF y por último, la obtención de la similitud de coseno el cual nos permite obtener una matriz de similitud entre los documentos y las noticias de tal manera que se pueda predecir si la información de una noticia es presuntamente falsa o real.

Este documento está organizado de la siguiente manera. En la sección 2 describimos trabajos relacionados sobre el análisis semántico latente. Luego, en la sección 3 se presenta a detalle la propuesta del sistema de predicción desarrollado por los autores. En la sección 4, se presentan los experimentos y resultados encontrados.

Finalmente, en la sección 5 se concluye con la importancia del desarrollo del sistema propuesto en este artículo científico.

TRABAJOS RELACIONADOS

Se tomaron en cuenta trabajos relacionados con LSA en diferentes aplicaciones, con esto podemos entender el alcance que podemos obtener usando este método. En el trabajo denominado "Latent Semantic Analysis: An Approach to Understand Semantic of Text" podemos encontrar aspectos básicos de LSA y los procesos que se deben aplicar para obtener las similitudes entre los términos de los documentos.

Los autores plantean como objetivo del método LSA la extracción de la relación semántica entre los términos de los documentos analizando la correlación con diferentes valores del espacio semántico truncado. El procedimiento que emplean para lograr su cometido es crear una matriz de documentos por términos, obtener las ocurrencias de los términos en los distintos documentos. Una vez que se consigue realizar estos procesos se aplica una reducción de la dimensión de la matriz término por documento a través del método SVD. Para lo cual, truncan las matrices resultantes del SVD a un

factor k , que representa el número de dimensiones deseadas. La SVD descompone la matriz original en tres matrices, que, al multiplicarse juntas, producen la matriz original del documento término. Pero para la optimización del espacio, se requiere calcular sólo la dimensión K más significativa (Kherwa and Bansal, 2017).

En la investigación "Application Research on Latent Semantic Analysis for Information Retrieval" se propone realizar el procedimiento de LSA aplicando clústeres k -means y similitud del coseno. Este método consiste en agrupar los documentos que previamente ya han sido pre procesados, utilizando la agrupación k -means basada en el análisis de la semántica latente, gracias a esto se ha obtenido un punto central dentro cada agrupación para posteriormente calcular la similitud entre el vector de consulta y los puntos centrales de cada agrupación para la recuperación. Al realizar este proceso se consigue que simplemente se calcule la similitud entre los vectores de consulta y los centros de agrupación calculados, con esto se consigue acortar el tiempo de recuperación de la información sin perder la precisión y eficacia de los resultados y reduciendo los tiempos de ejecución. (Wenli, 2016).

Si bien los métodos propuestos en estos trabajos funcionan correctamente y cumplen su cometido, con el pasar del tiempo ha venido surgiendo en conjunto con el aumento de información uno de los desafíos más grandes dentro de esta rama, la optimización y mayor eficiencia en el procesamiento. Es por eso que, en la investigación "Parallel Latent Semantic Analysis using a Graphics Processing Unit" se realiza una comparación entre algoritmos LSA con implementación en CPU y GPU. El resultado de estas comparaciones demuestra que el proceso utilizando en GPU es 5 o 6 veces más rápido que con CPU, ya que es sabido que la GPU puede resolver problemas altamente paralelos mucho más rápido que el CPU. Por lo tanto, con esta investigación nos damos cuenta de que un sistema desplegable que utilice una GPU para acelerar los procesos de LSA a gran escala sería una opción mucho más eficaz (en términos de relación costo/rendimiento) que utilizar un clúster de computadoras (Cavanagh et al., 2009).

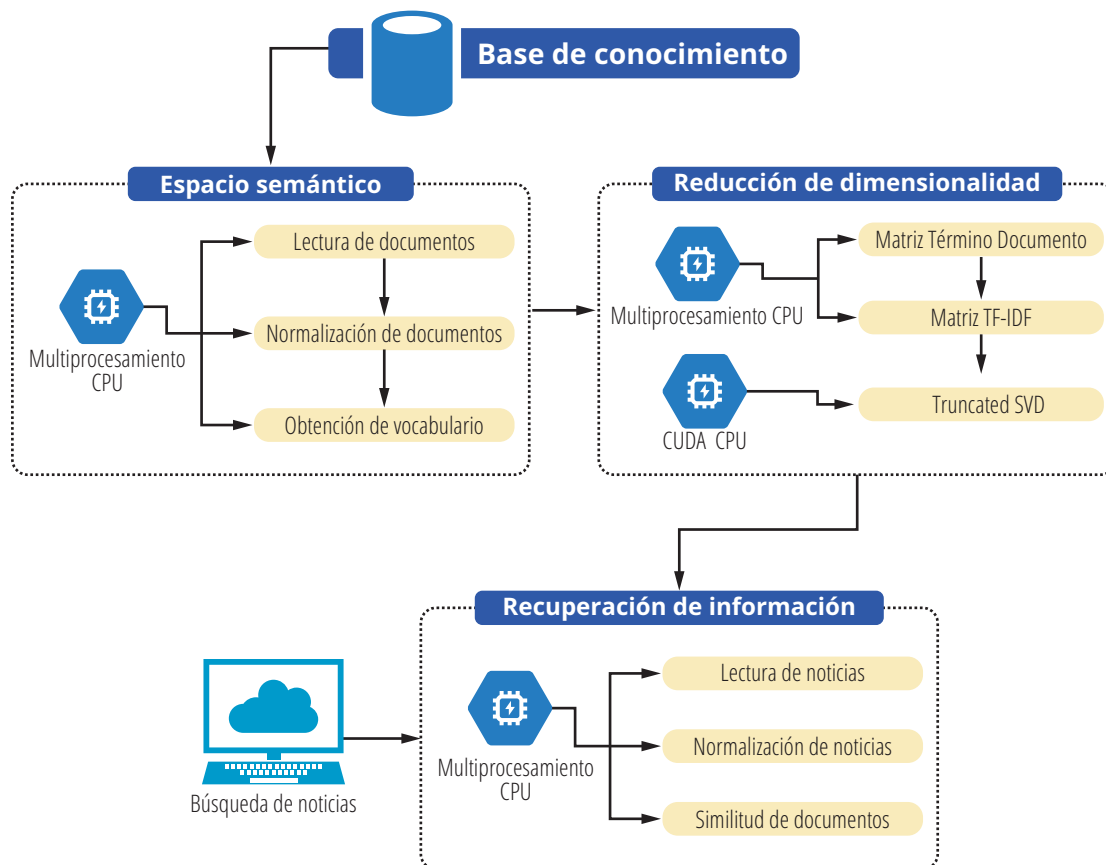
De igual manera en el trabajo "A Heterogeneous System Based on Latent Semantic Analysis Using GPU and Multi-CPU" se utilizan varias técnicas para reducir los costos computacionales y tiempos de ejecución. En este

trabajo se usan varias técnicas de programación que ayudarán al rendimiento de este proceso propuesto, primero se utiliza programación en paralelo dentro de la CPU para poder generar las matrices término-documento. Luego se emplean técnicas de programación dentro de la GPU usando CUDA para calcular la reducción de dimensionalidad de las matrices obtenidas anteriormente (León-Paredes et al., 2017). Los autores han encontrado una aceleración de 8x comparada con la versión en CPU del método LSA usando grandes conjuntos de datos.

Por lo tanto, el presente trabajo de investigación está basado en la investigación denominada "A Heterogeneous System Based on Latent Semantic Analysis Using GPU and Multi-CPU" antes descrita para implementar un sistema de detección de presuntas noticias falsas relacionadas al COVID-19.

SISTEMA PROPUESTO PARA LA PRESUNTA

Figura 1» Fases del sistema de predicción propuesto: (1) Espacio Semántico, (2) Reducción de Dimensionalidad, (3) Recuperación de información



DETECCIÓN DE NOTICIAS FALSAS SOBRE COVID-19

El sistema propuesto está compuesto por tres fases, (1) la construcción del espacio semántico en el cual usamos la arquitectura multi-CPU para la lectura de documentos, normalización de documentos y obtención del vocabulario, (2) la reducción de dimensionalidad en donde utilizamos arquitectura multi-CPU para armar la matriz término por documento y matriz normalizada TF-IDF, mientras que para el método de SVD utilizamos la arquitectura GPU, y (3) la recuperación de información donde utilizamos la arquitectura multi-CPU para lectura de noticias, normalización de noticias y obtención de similitud de los documentos como se puede observar en la Figura. 1. A continuación, detallamos cada una de las fases implementadas en el sistema propuesto.

Espacio Semántico

En primer lugar, se procede a construir un espacio semántico, para ello se debe tener una base de conocimiento que por lo general se almacena en un disco duro. Por tal cuestión, se necesita leer los documentos de texto almacenados en un dispositivo de hardware externo. A medida que va creciendo el número de documentos el costo computacional va aumentando, por lo que utilizamos técnicas de multiprocesamiento a nivel de CPU para leer una lista de documentos. Esta lista es normalizada, al evitar palabras comunes conocidas como stopwords por ejemplo, "the", "for", "and", entre otras. Asimismo, eliminamos caracteres especiales como - "?/[.]". También, se lemantizan ciertas palabras que se escriben diferente, pero tienen el mismo significado, este proceso da un sinónimo general a las diferentes palabras. Por último, procedemos a obtener la frecuencia de las palabras de cada documento en el conjunto de documentos de tal manera que como resultado obtenemos un vocabulario de palabras normalizadas y que tienen una mayor frecuencia en el conjunto de documentos. En este punto se debe destacar, que la base de conocimiento con el que ha sido entrenado nuestro sistema se encuentra en el idioma inglés, dicha base la discutiremos en la siguiente sección.

Reducción de Dimensionalidad

Con el vocabulario de palabras obtenidas en la fase anterior se procede a construir la matriz término-documento, la cual nos permite obtener la frecuencia que se repite una palabra dentro del vocabulario y dentro del conjunto de documentos de la base de conocimientos. Con la matriz término-documento obtenida, procedemos a normalizar los valores de la matriz aplicando el método TF-IDF, esto nos permite obtener la relevancia de las palabras obtenidas en el espacio semántico dentro del conjunto de documentos de tal manera que tengamos una matriz con sus valores normalizados. La fórmula para calcular la matriz TF-IDF está dada por:

$$TF = \frac{t \in d}{T \in d} \quad 1$$

Donde t es la cantidad de veces que un término aparece en el documento, y T la cantidad de palabras totales en el documento.

$$IDF = \log\left(\frac{D}{nT}\right) \quad 2$$

Donde D es el número total de documentos, y nT la cantidad de veces que aparece el término en todos los documentos. Finalmente, el TF-IDF se calcula con el siguiente producto:

$$TF - IDF = TF \times IDF \quad 3$$

La Matriz TF-IDF se obtiene utilizando técnica de multiprocesamiento, en donde procesamos de manera paralela tanto la obtención del IDF como la multiplicación entre TF e IDF, con ello logramos que los procesos para obtener la matriz TF IDF se hagan en paralelo y el costo computacional sea bajo. Una vez obtenida la matriz TF-IDF, procedemos a realizar una reducción de dimensionalidad, para ello utilizamos la técnica de descomposición de valores singulares. SVD permite reducir la dimensionalidad de una matriz basándose en sus primeros k componentes. Para calcular el SVD tenemos la siguiente ecuación:

$$A_{[w,d]} = T_{[w,n]} \times S[n,n] \times D^T_{[n,d]} \quad 4$$

Dónde $A_{w,d}$ es la matriz término-documento; T es una matriz ortogonal $[w \times n]$ cuyos valores representan los vectores singulares izquierdos de A; D es una matriz ortogonal $[d \times n]$ cuyos valores representan los vectores singulares derechos de A; y S es una matriz diagonal $[n \times n]$ que contiene los valores singulares de A en orden descendente. Se debe tener en cuenta que n es el valor más pequeño entre el número total de palabras.

Para procesar el método SVD se ha utilizado procesamiento en paralelo a nivel de la GPU, para este caso utilizamos la biblioteca Rapids¹ que contiene un conjunto de algoritmos que pueden ser ejecutados a nivel de GPU, utilizando CUDA como arquitectura base.

Recuperación de Información

Tras finalizar el truncamiento de la matriz término-documento se procede a comparar noticias tanto falsas como reales con el espacio semántico construido en las fases anteriores a partir de la base de conocimiento. Para esto las noticias son extraídas de varias páginas relacionadas a la medicina, bibliotecas digitales, entre otras, de tal manera que se obtiene un conjunto de noticias tanto reales como falsas. Una vez encontradas las noticias se procede a normalizarlas y a insertarlas en el espacio semántico, para ello se uti-

¹ Más información sobre la biblioteca Rapids en <https://rapids.ai/>

lizan técnicas de multiprocesamiento a nivel de CPU para la lectura y normalización de noticias. Dichos procesos constan en remover stopwords, signos de puntuación y lematización de palabras, a su vez se aplica los métodos de término por documento, matriz TF-IDF a nivel de CPU, y SVD a nivel de GPU debido a que se debe insertar dichas noticias en el espacio semántico de la base de conocimiento antes procesado. Luego, se procede a obtener la similitud entre las noticias extraídas y los documentos de la base de conocimiento. Para obtener la similitud entre dos documentos se aplica la similitud de coseno que es una medida de similitud entre dos vectores midiendo el coseno del ángulo entre ellos. El coseno de 0 es 1, y menor que 1 para cualquier otro ángulo; el valor más bajo del coseno es -1. El coseno del ángulo entre dos vectores determina así si dos vectores apuntan aproximadamente en la misma dirección (Soyusiawaty and Zakaria, 2018), su fórmula está dada por:

$$\text{Cos}(\theta) = \frac{a \cdot b}{\|a\| \|b\|} \quad 5$$

Donde, a hace referencia a un documento en la base de conocimiento y b hace referencia a una noticia extraída y desplegada en el espacio semántico. Con esto podemos obtener la similitud que existe

entre todos los documentos de la base de conocimientos y cada una de las noticias extraídas.

Experimentos y Resultados

Para realizar los experimentos utilizamos el dataset denominado COVID-19 Open Research Dataset Challenge (CORD-19) el cual contiene artículos científicos en el idioma inglés relacionados al COVID-19. El dataset al momento de realizar nuestros experimentos contiene 172.252 documentos de los cuales solamente 65.910 contienen el abstract o resumen que es el texto que vamos a utilizar para los experimentos. En la Tabla 1, podemos apreciar los parámetros principales para nuestro sistema de predicción donde el número de cores de CPU hace referencia a la cantidad de procesadores que utilizamos para aplicar la técnica de multiprocesamiento, k componentes SVD es el número de componentes que utilizamos para probar el método de descomposición de valores singulares, número de documentos hace referencia a la cantidad de documentos que utilizamos de la base de conocimiento para realizar las pruebas de nuestro sistema de predicción y por último frecuencia de palabras es el rango de frecuencia para filtrar las palabras obtenidas por la normalización de documentos y así obtener el vocabulario.

Tabla 1: Principales parámetros utilizados para el entrenamiento del sistema de predicción propuesto

Número de cores CPU	k componentes SVD	Número de Documentos	Frecuencia de palabras
32	75, 150, 300	5000, 6000, 7000	5

Para probar nuestro sistema de predicción realizamos varios experimentos utilizando los parámetros de la Tabla 1, dichos experimentos se realizaron para las tres fases de nuestro sistema. Para cada uno de los procesos que tiene las fases del sistema se midió el tiempo de ejecución tanto a nivel de CPU como a nivel de GPU.

De la misma manera se mide el nivel de predicción de nuestro sistema referente a las noticias utilizando distinta cantidad de número de documentos y k componentes para la reducción de dimensionalidad, de la misma manera se midió el tiempo de ejecución de la predicción.

Tabla 2: Tiempo de ejecución de la fase para obtener el espacio semántico utilizando Multi-CPU

Número de documentos	Lectura de documentos (segundos)	Normalización de documentos (segundos)	Obtención vocabulario (segundos)
5000	0.59	6.15	0.21
6000	0.61	7.62	0.25
7000	0.70	7.84	0.28

Como se puede apreciar en la Tabla 2 los tiempos de todos los procesos de la primera fase están dentro de un tiempo de procesamiento adecuado, debido a la técnica de multiprocesamiento a nivel de CPU. Podemos darnos cuenta de que el proceso más demorado ha sido la normalización de documentos y esto es debido a que se debe realizar varios métodos para limpieza y lematización de palabras, sin embargo, el tiempo sigue siendo corto y el costo computacional no es alto.

La segunda fase de reducción de dimensionalidad se realizó con el vocabulario obtenido en la primera fase. El

proceso de la segunda fase consta de obtener la matriz término documento de tal manera que obtengamos la frecuencia de las palabras del vocabulario para cada documento de la base de conocimiento. Una vez realizado esto se procede a normalizar la matriz término documentos utilizando la matriz TFIDF, esta nos permite obtener la relevancia de las palabras dentro del conjunto de documentos, de tal manera se consigue a la vez normalizar la matriz. Cada proceso mencionado fue realizado mediante técnica de multi-CPU para mejorar los tiempos de ejecución. En la Tabla 3 podemos apreciar el tiempo que tomó la ejecución de cada proceso.

Tabla 3: Tiempo de ejecución de la fase de reducción de la dimensión utilizando Multi-CPU

Número de documentos	Número de palabras	Tamaño de la matriz	Matriz Término-Docmento (segundos)	Matriz TF-IDF (segundos)
5000	10,804	5,420,000	85.73	816.44
6000	11,949	71,694,000	116.60	1097.51
7000	13,131	91,917,000	146.21	1047.91

Como se aprecia en la Tabla 3 el proceso que más tiempo tomó en ejecutarse fue la obtención de la matriz TF-IDF debido a que el proceso de normalización requiere realizar un conjunto de operaciones matemáticas para lograr obtener dicha matriz. Un inconveniente de esta matriz es la gran cantidad de información que tiene, como podemos apreciar en la Tabla 3, el tamaño de la matriz es muy grande, ya que tiene un tamaño de 91,917,000 por lo que el costo computacional aumenta.

Luego se utiliza la técnica de descomposición de valores singulares (SVD) para reducir el tamaño de la matriz dependiendo del número de componentes que utilice-

mos. El valor de k depende del tamaño de la matriz, para este caso utilizamos entre 75 y 300 k componentes. Como resultado obtenemos una matriz truncada a sus primeras k dimensiones en donde cuyas filas son documentos y columnas son los términos.

Para la técnica de SVD hemos planteado utilizar tres k componentes como se aprecia en la Tabla 1, a su vez utilizamos el algoritmo Jacobi (An and Wang, 2016) el cual es mucho más rápido que el algoritmo full. El proceso de SVD se realizó con procesamiento a nivel de GPU, utilizando la arquitectura CUDA. En la Tabla 4 podemos visualizar los tiempos de ejecución para cada k componente.

Tabla 4: Tiempo de ejecución para la descomposición de valores singulares utilizando CUDA

Número de documentos	$k=75$	Obtención SVD(segundos) $k=150$	$k=300$
5000	88.63	89.61	88.85
6000	112.37	111.51	112.32
7000	140.03	139.90	140.14

Como podemos apreciar en la Tabla 4, el tiempo más grande se obtiene al momento de procesar 300 componentes con 7000 documentos, esto debido al tamaño de la matriz, ya que como se puede apreciar en la Tabla 3, el tamaño de la matriz es muy grande ya que tiene un tamaño de 91,917,000. Sin embargo, podemos apreciar que

gracias al procesamiento en paralelo a nivel de GPU los tiempos son apropiados para gran cantidad de información que está siendo procesada.

Por último, contamos con la fase de recuperación de información, donde se realiza el proceso de lectura de no-

ticias, normalización del texto y obtención de vocabulario de la misma manera como lo hemos mencionado en las tres fases de entrenamiento del sistema. Así mismo procedemos a utilizar los métodos para la obtención de la matriz término documento y matriz TF-IDF, todos estos procesos están siendo ejecutados en Multi-CPU.

Después de ejecutar dichos procesos procedemos a ejecutar el método de SVD a nivel de GPU de tal manera que obtengamos una matriz reducida para que sea insertada en el espacio semántico y poder ejecutar la similitud de documentos. Para la parte de similitud utilizamos la

técnica del coseno que mediante su fórmula se obtiene una similitud entre la noticia y el conjunto de documentos que se tiene. Para este caso, se ha determinado que un valor 0.15 en adelante implica que la noticia tiene una gran similitud con la base de conocimientos y una noticia que está por debajo de dicho valor se podría presumir que es una noticia que no tiene relación con bases o hechos científicos. Para ejecutar todo este proceso utilizamos técnicas de Multi-CPU para optimizar los procesos y mejorar los tiempos. En la Tabla 5 podemos apreciar los resultados de lo antes mencionado.

Tabla 5: Tiempo de ejecución de la fase de recuperación de información utilizando Multi-CPU y 15 noticias

Número de documentos	Lectura de Noticias (segundos)	Normalización de Noticias (segundos)	Obtención Similitud (segundos)		
			k=75	k=150	k=300
5000	8.83	43.62	10.12	10.65	11.09
6000	8.83	43.62	12.93	13.71	15.97
7000	8.83	43.62	12.83	14.28	16.49

Como podemos apreciar en la Tabla 5, los tiempos en los procesos de lectura y normalización de noticias son los mismos indiferentemente al número de documentos, esto debido a que la longitud de lectura y normalización de noticias es la misma para el número de documentos. Lo que podemos notar claramente es la diferencia de tiempos que existe al realizar la similitud de Coseno.

Para la similitud del coseno utilizamos quince noticias las cuales están divididas en cinco noticias falsas,

cinco noticias reales y cinco noticias relacionadas a los documentos de la base de conocimientos. Al final cada una de esas noticias se utilizan para obtener la similitud de documentos, con ello obtenemos la similitud de todas las noticias frente a todos los componentes de la matriz reducida de la base de conocimientos. Con ello obtenemos la media de cada una de las noticias de tal manera que obtengamos un valor de 0 a 1 para detectar si la noticia es falsa o verdadera. A continuación, en la Tabla 6 presentamos información relevante de las noticias.

Tabla 6: Noticias obtenidas para experimentos de predicción

Título de la Noticia	URL	Tipo de Noticia
Prediction and Evolution of B Cell Epitopes of Surface Protein in SARS-CoV-2	https://www.biorxiv.org/ /10.1101/ 2020.04.03.022723v1.full	Artículo
Potentially highly potent drugs for 2019-nCoV	https://www.biorxiv.org/ content/10.1101/ 2020.02.05.936013v1.full	Artículo
The SARS-CoV-2 receptor-binding domain elicits a potent neutralizing response without antibody-dependent enhancement	https://www.biorxiv.org/ content/10.1101/ 2020.04.10.036418v1.full	Artículo
Mechanistic modeling of the SARS-CoV-2 disease map	https://www.biorxiv.org/content/10. 1101/2020.04.12.025577v1.full	Artículo
Significance of hydrophobic and charged sequence similarities in sodium-bile acid cotransporter and vitamin D-binding protein macrophage activating factor	https://www.biorxiv.org/ content/10.1101/ 2020.03.03.975524v1.full	Artículo

F.D.A. Clears First Home Saliva Test for Coronavirus	https://www.nytimes.com/2020/05/08/health/fda-coronavirus-spit-test.html?action=click&module=Top%20Stories&pgtype=Homepage	Real
UK scientists condemn 'Stalinist' attempt to censor Covid-19 advice	https://www.theguardian.com/world/2020/may/08/revealed-uk-scientists-fury-over-attempt-to-censor-covid-19-advice	Real
China is promoting coronavirus treatments based on unproven traditional medicines	https://www.nature.com/articles/d41586-020-01284-x	Real
How Remdesivir, New Hope for Covid-19 Patients, Was Resurrected	https://www.nytimes.com/2020/05/01/health/coronavirus-remdesivir.html?searchResultPosition=10	Real
Coronavirus treatments: what drugs might work against COVID-19?	https://theconversation.com/coronavirus-treatments-what-drugs-might-work-covid-19-135352	Real
Here's What a Pandemic Would Look Like With Zombies	https://futurism.com/pandemic-coronavirus-zombies	Falso
COVID-19 Apocalypse: Coronavirus pandemic to stay for years even with vaccine, highest single-day 1,281 cases in Venezuela, record 103 new cases in South Korea	https://www.dimsumdaily.hk/covid-19-apocalypse-coronavirus-pandemic-to-stay-for-years-even-with-vaccine-highest-single-day-1281-cases-in-venezuela-record-103-new-cases-in-south-korea/	Falso
Conspiratorial Corona: Hoaxes and Conspiracy Theories in the Balkans	https://balkaninsight.com/2020/07/07/conspiratorial-corona-hoaxes-and-conspiracy-theories-in-the-balkans/	Falso
what is the truth behind the 5g coronavirus conspiracy theory culture clash	https://www.euronews.com/2020/05/15/what-is-the-truth-behind-the-5g-coronavirus-conspiracy-theory-culture-clash	Falso
The Post-Coronavirus World May Be The End Of Globalization	https://www.forbes.com/sites/kenrapoza/2020/04/03/the-post-coronavirus-world-may-be-the-end-of-globalization/#329758d37e66	Falso

Por último, en la Figura 2, Figura 3 y Figura 4 presentamos los resultados de las predicciones obtenidas por el sistema propuesto en donde se puede apreciar las simili-

tudes obtenidas por cada uno de los k componentes que se ejecutaron en los experimentos arriba mencionados.

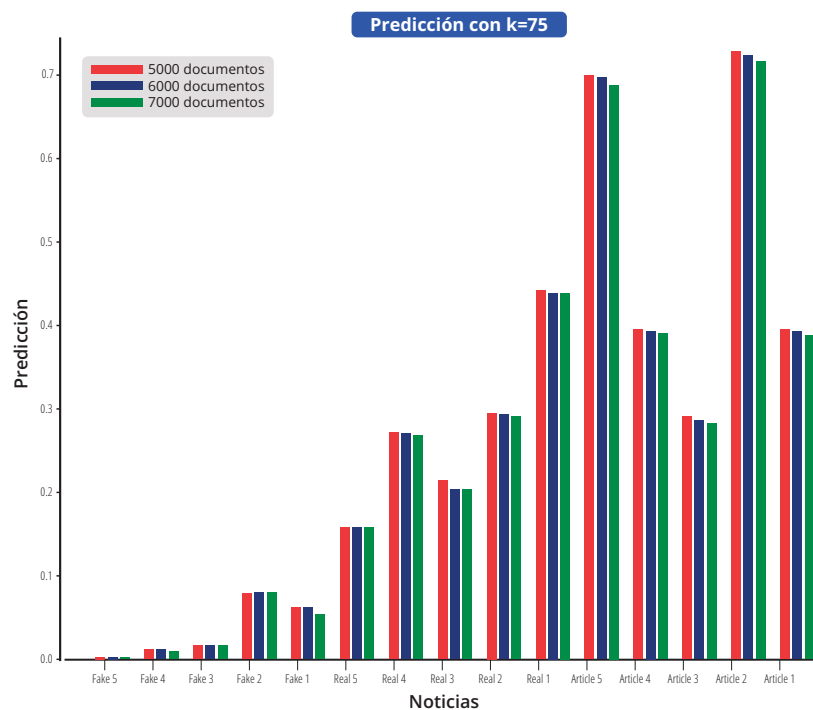
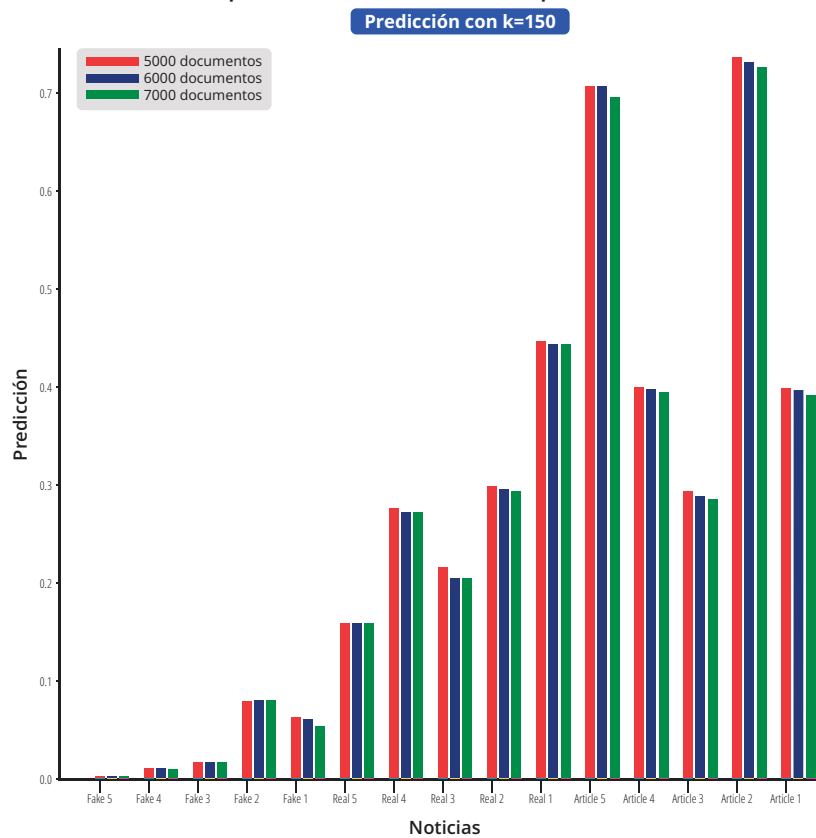
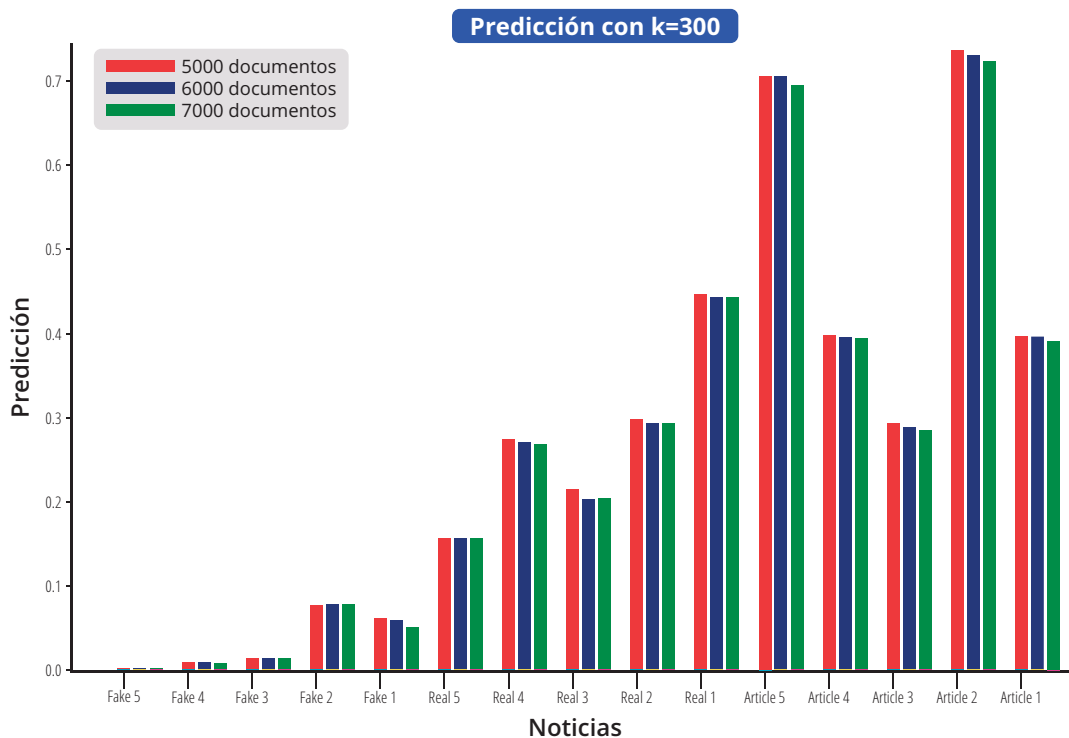
Figura 2» Resultados del sistema de predicción con $k = 75$ componentes

Figura 3» Resultados del sistema de predicción con $k = 150$ componentes


Figura 4» Resultados del sistema de predicción con $k = 300$ componentes

Como se puede apreciar en los resultados las similitudes para los tres k componentes son similares y a su vez coinciden con la baja similitud de noticias falsas lo cual comprueba la precisión de nuestro sistema, ya que al no ser una noticia real o relacionado a la base de conocimiento la similitud debe ser baja. Por otro lado, como podemos apreciar las noticias reales tienen una similitud mayor. A su vez, también podemos darnos cuenta que la última división de noticias obtenidas de bibliotecas digitales y las cuales explican con bases y hechos científicos el virus Covid-19, tienen la similitud más alta puesto que sus temas están relacionados con la base de conocimientos debido a que en ambas hablan de hechos científicos.

CONCLUSIONES

El sistema de predicción permite detectar presuntas noticias falsas utilizando Procesamiento de Lenguaje Natural, para ello se utilizó la técnica LSA implementada en tres fases las cuales son: espacio semántico, reducción de dimensionalidad y recuperación de información. Para las fases mencionadas se utilizaron técnicas de procesa-

miento paralelo a nivel de CPU y GPU. Los resultados obtenidos por el sistema de predicción son buenos puesto que la similitud de noticias reales y noticias científicas superan el rango de predicción de 0.15 donde su valor mínimo es 0.16 y su máximo es 0.73. Por otro lado, también podemos afirmar dichos resultados gracias a que las noticias falsas no superan el rango de predicción teniendo como mínimo un valor de 0.01 y como máximo un valor de 0.079.

Además, el sistema de predicción presenta tiempos apropiados para cada una de las fases del sistema. Para la fase de espacio semántico tenemos un tiempo promedio de 8.03 segundos, por otro lado, en la fase de reducción de dimensionalidad tenemos un tiempo promedio de 1337.2 segundos y para la fase de recolección de información tenemos un promedio de 12.32 segundos para las 15 noticias. Dicha investigación se presenta como una línea base para futuras investigaciones de tal manera que se puedan aplicar otras técnicas como redes neuronales las cuales nos permiten obtener mejores predicciones, así mismo combinar otras técnicas para que el sistema propuesto pueda generar mejores resultados.

BIBLIOGRAFÍA

- An, J. and D. Wang (2016). Efficient one-sided jacobi svd computation on amd gpu using opencl. In 2016 IEEE 13th International Conference on Signal Processing (ICSP), pp. 491-495.
- Cavanagh, J. M., T. E. Potok, and X. Cui (2009). Parallel latent semantic analysis using a graphics processing unit. In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO '09, New York, NY, USA, pp. 2505-2510. Association for Computing Machinery.
- Hlaing, M. M. M. and N. S. M. Kham (2020). Defining news authenticity on social media using machine learning approach. In 2020 IEEE Conference on Computer Applications (ICCA), pp. 1-6.
- Kherwa, P. and P. Bansal (2017, Sep.). Latent semantic analysis: An approach to understand semantic of text. In 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), pp. 870-874.
- Landauer, T. K. and S. T. Dumais (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review* (2), 211.
- León-Paredes, G. A., L. I. Barbosa-Santillán, and J. J. Sánchez-Escobar (2017). A Heterogeneous System Based on Latent Semantic Analysis Using GPU and Multi-CPU. *Scientific Programming* 2017, 8131390.
- Soyusiawaty, D. and Y. Zakaria (2018). Book data content similarity detector with cosine similarity (case study on digilib.uad.ac.id). In 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), pp. 1-6.
- Wenli, C. (2016). Application research on latent semantic analysis for information retrieval. In 2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp. 118-121.
- Zhang, W., T. Yoshida, and X. Tang (2008). TFIDF, LSI and multi-word in information retrieval and text categorization. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 108-113.