Instituto Politécnico Nacional

Centro de Investigación en Computación

"Sistema de visualización de la información de tópicos más importantes generados en medios sociales"

T E S I S

PARA OBTENER EL GRADO DE:

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

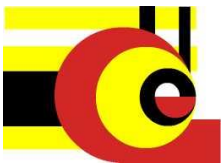
P R E S E N T A

EL ING. MAURICIO IVÁN GUERRERO HERNÁNDEZ

DIRECTORES DE TESIS:

Dr. Adolfo Guzmán Arenas

Dr. Gilberto Lorenzo Martínez Luna



México, D.F. Agosto 2015

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 11:00 horas del día 17 del mes de junio de 2015 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Sistema de visualización de la información de tópicos más importantes generados en medios sociales"

Presentada por el alumno:

GUERRERO	**HERNÁNDEZ**	**MAURICIO IVÁN**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

A	1	3	0	2	5	6

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Adolfo Guzmán Arenas

Dr. Gilberto Lorenzo Martínez Luna

Dr. Grigori Sidorov

Dr. Francisco Hiram Calvo Castro

Dr. Jesús Manuel Olivares Ceja

M. en C. Sandra Dinora Orantes Jiménez

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN COMPUTACION
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 23 del mes de Junio del año 2015, el (la) que suscribe Mauricio Iván Guerrero Hernández alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro A130256, adscrito al Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del Dr. Adolfo Guzmán Arenas y el Dr. Gilberto Lorenzo Martínez Luna y cede los derechos del trabajo titulado "Sistema de visualización de la información de tópicos más importantes generados en medios sociales", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección morris0287@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Mauricio Iván Guerrero Hernández

Nombre y firma

Resumen

Las publicaciones en los medios de información y los medios sociales son una forma de ver el interés de los publicadores y por lo tanto su contenido y es posible que a partir de ellas generar modelos del interés de las publicaciones. Estos medios también se destacan por la rapidez en que las publicaciones se llevan a cabo y que se relacionan con tópicos de interés y que podrían ser una medida de la importancia de los tópicos, los cuales podrían ser parte del modelo.

Las características de las publicaciones ya sean noticias o tweets, pueden variar en cuanto a los tópicos de interés que aparecen y desaparecen, así como el número de ellos y el número de publicaciones, dependiendo del rango de tiempo coleccionado (día, semana, quincena y mes entre otros). Dado que puede haber tópicos que no se conocen, que dependen del número de las publicaciones y del rango del tiempo, para identificarlos se requiere de un algoritmo de clasificación no supervisado.

Detectar los tópicos más importantes en medios sociales es de utilidad dado que de manera casi inmediata se puede conocer el surgimiento de problemáticas en diferentes campos de la ciencia, tecnología, dispersión de enfermedades o algún problema existente en el país. El conocimiento de estos tópicos puede ayudar a evitar seguir utilizando tecnología con problemas, productos que pueden ser nocivos al ser humano, realizar o evitar actividades que puedan afectar la economía de un país, entre otros.

En este trabajo se da una propuesta de solución de detectar los tópicos más importantes e informar de su permanencia en el tiempo, para lo cual se utiliza una versión ya implementada del algoritmo no supervisado llamado Latent Dirichlet Allocation (LDA), que con modificaciones permite identificar un número determinado de tópicos a partir de un rango de días o un período de tiempo en una colección de publicaciones preparadas con el Procesamiento de Lenguaje Natural.

Las colecciones de publicaciones se obtienen de suscripciones RSS de secciones de periódicos, así como de tweets que son publicados por periodistas destacados por su índice de seguidores o número de publicaciones, información que es almacenada en bases de datos, pero antes es tratada con procesos del área de Minería de Datos (extracción, transformación y carga).

En el trabajo se define el concepto de tópico más importante, basado en el mayor número de apariciones de un tópico en diferentes noticias de la colección. También se da una definición de similitud entre tópicos de diferentes períodos que ayuda a revisar su aparición de un tópico a lo largo de varios períodos de tiempo. Definiciones útiles para identificar y dar a conocer los tópicos a los usuarios que acceden a estos medios y dar un seguimiento de ellos, si son de su interés.

También se construye un prototipo de software que utiliza las anteriores definiciones, el cual incluye una interfaz de usuario para alimentar los parámetros de entrada como: número de periodos, número de tópicos, tipo de noticia, fecha y tipo de periodo. Parámetros que usa el modelo LDA y que genera un numero de tópicos requeridos para cada uno de varios periodos deseados a analizar con una duración determinada días y que tengan una similitud s en el rango [0, S] con $s < 1$.

El prototipo, bajo el concepto de visualización, permite observar la relación entre los tópicos que son: el número de palabras que coinciden en dos tópicos, cuales palabras y la posición de ellas en dos tópicos y un valor asignado de similitud. Lo que origina la similitud entre los tópicos de diferentes periodos y medir en cuántos periodos se mantiene la importancia de un tópico. Así también se desarrolla una nube de palabras que ayuda a determinar la importancia de un tópico mediante las palabras con mayor tamaño y la visualización de similitud con respecto a palabras que existan en tópicos de diferentes periodos

Clasificación ACM:

CCS → computación centrada en Humanos → Visualización → Visualización aplicada a dominios → Visualización de la Información
CCS → Metodologías de la computación → Aprendizaje automático → Paradigmas de aprendizaje → Aprendizaje no supervisado → Modelado de Tópicos

Abstract

The publications in the media and social media is a way to see the interest of the publishers and therefore it's content and may generate them from models interest publications. These media are also highlighted by the speed with which events are published that relate to topics of interest and could be a measure of the importance of the topics, which could be part of the model.

The characteristics of publications either news or tweets, may vary as to the topics of interest which appear and disappear, and the number of them and the number of publications, collected depending on the range of time (day, week, fortnight and month etc.). Since there may be topics that are not known, depending on the number of publications and the range of time required to identify an unsupervised classification algorithm.

Detect the most important topics in social media is useful because almost immediately you can know the emergence of problems in different fields of science, technology, spread of disease or an existing problem in the country. The knowledge of these topics can help prevent problems continue to use technology, products that can be harmful to humans, take or avoid activities that may affect the economy of a country, among others.

In this paper it is given a proposal for a solution to detect the most important topics and report on their permanence in time, for what you use an already deployed version of the model is not supervised called Latent Dirichlet Allocation (LDA), which allows identified a number of topics from a range of days and a period of time in a collection of publications prepared with the Natural Language Processing.

The collections of publications obtained from RSS subscriptions sections of newspapers, as well as tweets are published by journalists highlighted by its index of followers or number of publications, information that is stored in databases, but before it is treated with processes area data Mining (extraction, transformation and loading).

At work, the most important topic concept is defined, based on the highest number of occurrences of a news topic in different collection. It also gives a definition of similarity between different periods topical helps examine its appearance of a topical over various time

periods. Definitions useful to identify and publicize the topics to users who access the media and keep track of interest on the topic.

Also built a software prototype that uses the above definitions, which includes a user interface to feed the input parameters such as number of periods, number of topics, news type, date and type of period. LDA parameters using model and generating topical K required for each of the P periods of duration T days and having a similarity s in the range [0, S].

The prototype, under the concept of visualization shows the relationship among the topics on the number of words that match in two topics, determining which words and what position are found in both topical and an assigned value of similarity. What causes the similarity between the topics of different periods and measure how many periods the importance of a topic is maintained. So a word cloud that helps determine the importance of a topic and the measure of similarity for words that exist in different periods topics is also developed

Classification ACM:

CCS → Human-centered computing → Visualization → Visualization application domains → Information visualization

CCS → Computing methodologies → Machine learning → Learning paradigms → Unsupervised learning → Topic modeling

Agradecimientos

Primero que nada doy gracias a Dios por mantenerme con salud y darme la fortaleza y sabiduría que de él provienen para que día a día pudiera cumplir con mis deberes para así lograr un objetivo más en la vida y recordar que el amor que me tiene es único e inigualable.

Al Dr. Gilberto Lorenzo Martínez Luna por darme la oportunidad de desarrollar mi tema de tesis a su cargo y por el apoyo que me dio durante mi estancia en el centro de investigación.

Al Dr. Adolfo Guzmán Arenas por las sesiones de aprendizaje cooperativo, con las cuales se lograron obtener varias ideas que lograron ayudar a la conclusión del trabajo de tesis.

A mis padres por darme su apoyo en los intentos por entrar a la maestría, así como su apoyo al darme casa y sustento. También más que nada, su apoyo moral en los problemas personales que tuve durante mi estancia en la maestría y que en la salud y enfermedad siempre he contado con ellos los quiero.

A mis hermanos que están al pendiente de mi progreso y que siempre están dándome palabras de aliento para terminar y superarme aún más.

A mi novia Mónica Lorena, quien es una mujer estupenda que Dios puso en mi camino para estar a mi lado y que me enseña cosas de la vida de las cuales no tengo conocimiento, ella es apoyo importante porque ora por mí y siempre camina a mi lado ya que nunca me ha dejado caer TE AMO.

A mis amigos de la generación: Erika, Caro, Felipe, Emma, Miguel, Eder, Luis Berthet, Armando, José Luis por esos momentos de risas que tuvimos desde los primeros días, los cuales ayudaron a no sentirnos presionados (locos) y lograr una sana convivencia en la maestría porque siempre fueron chistes, apodos o risas incontenibles Gracias amigos por todo viva el A13.

INDICE

Glosario. 12
Capítulo 1: Introducción. 14
1.1 Planteamiento del problema. 14
1.2 Objetivo general. 15
1.3 Objetivos particulares. 15
1.4 Justificación. 16
Capítulo 2: Estado del arte. 17
2.1. Trabajos actuales sobre tópicos en Twitter. 17
2.1.1 Tendencias en Twitter. 17
2.1.2 *"Twitter Trending Topic Classification".* 18
2.1.3 *"Developing a Hierarchical Multi-Label Classifier for Twitter Trending Topics".* 19
2.1.4 *"Beyond Trending Topics: Real-World Event Identification on Twitter".* 20
2.1.5 *"TwitterStand: News in tweets".* 21
2.1.5 *"Multi-Modal Multi-Correlation Person-Centric News Retrieval".* 21
2.1.6 "Comparing Twitter and Traditional Media using Topic Models". 22
Capítulo 3: Marco teórico. 25
3.1 Conceptos generales. 25
3.1.1 Recuperación de la información (IR) 25
3.1.2 Procesamiento del lenguaje natural (PLN). 25
3.1.3 Minería de Datos. 27
3.1.4 Visualización de la Información. 30
3.2 Métodos de clasificación. 30
3.2.1 Naive Bayes (NB). 30
3.2.2 K-Nearest Neighbors (K-NN). 31
3.2.3 Term frequency · Inverse document frequency (TF·IDF). 32
3.2.4 Matriz término documento (MTD). 34
3.2.5 Latent Dirichlet Allocation (LDA). 35
Capítulo 4: Metodología e Implementación. 37
4.1 Desarrollo del sistema de visualización de tópicos más importantes generados en medios sociales (VITOIM). 37
4.1.1 Arquitectura del sistema VITOIM. 37
4.2 Recolección de Datos. 40
4.2.1 Recolección de Noticias 40
4.2.2 Transformación de las noticias Firebird a MySQL. 42

4.2.3 Colección de reporteros mexicanos en Twitter. 43
4.2.4 Programa que recolecta tweets mediante periodistas que se consideran importantes. 44
4.2.5 Creación de un diccionario de todas la noticias en forma Término Frecuencia. 45
4.2.6 Modelo LDA y su implementación. 46
4.2.7 Programa para determina tópicos importantes en las noticias durante un periodo de tiempo. 52
4.3 Modelo Formal de comparación entre tópicos 56
4.3.1 Número de comparaciones entre los Tópicos por periodos. 56
4.3.2 Modelo de similitud y distancia entre tópicos. 57
4.4 Gráficas de noticias. 60
4.4.1 Gráficas generadas del número de noticias en un tópico de cada periodo. 60
4.4.1 Gráficas generadas de similitud entre tópicos. 64
4.4.2 Gráfica de nube de palabras. 70
4.4.3 Gráfica de nube de palabras por tópicos. 71
Capítulo 5: Pruebas y Resultados. 75
5.1 Rendimiento del modelo LDA en cuanto a tópicos conocidos. 75
5.2 Comportamiento de memoria durante la utilización del programa (E) que utiliza el modelo LDA... 80
Capítulo 6: Aportación computacional y Conclusiones. 84
6.1 Aportación. 84
6.2 Conclusión 85
Capítulo 7: Trabajo futuro. 86
7.1 Trabajo futuro 86
Referencias 87
Apéndice. 88

Figuras.

Figura 1. Tendencias adquiridas directo de Twitter personal. *18*
Figura 2. "Twitter Trending Topics Classification" [9]. *20*
Figura 3. Minería de conocimiento. *27*
Figura 4. La minería de datos como un paso en el proceso de descubrimiento de conocimiento. *29*
Figura 5. Ejemplo de matriz término documento. *34*
Figura 6. Modelo de representación gráfica de LDA. *36*
Figura 7. Arquitectura del sistema de visualización de tópicos más importantes generados en medios sociales (VITOIM). *39*
Figura 8. Recolección de noticias RSS. *41*
Figura 9. Programa (A) extracción, transformación y carga de noticias Firebird a MySQL. *42*
Figura 10. Periodistas mexicanos más influyentes en Twitter Dic. 2014 ..... *43*
Figura 11. Parte parcial de la arquitectura del sistema VITOIM la cual muestra la utilización de un programa (C) desarrollado para la captura de tweets sobre periodismo. *44*
Figura 12. Parte parcial de la arquitectura del sistema VITOIM que muestra el desarrollo del programa (B) y el programa (D), los cuales generan tres archivos de salida para que el sistema VITOIM pueda procesar la salida mediante LDA. *45*
Figura 13. Ejemplo del archivo con formato "archivo.tokens" donde se muestra el diccionario de palabras. *48*
Figura 14. Ejemplo del archivo .ldac que contiene del índice invertido para la implementación del modelo LDA *48*
Figura 15. Ejemplo del archivo con formato "archivo.titles" donde se muestra el id de la noticia y su contenido. *49*
Figura 16. Ejemplo del archivo de salida del programa (E) que implementa el modelo LDA" generando Palabras_por_Tópico, Noticias_por_Tópico, y Cantidad_de_Noticias_por_Tópico, al lado izquierdo de los (:) siempre se localiza un identificador de un tópico y del lado derecho de los (:) se encuentra la información asignada a dicho tópico. *50*
Figura 17. Ventana inicial del sistema VITOIM, la cual ejecuta primero el programa (D) que genera los tres archivos de salida que utilizara el programa (E) mediante los cuales implementa el modelo LDA. *53*
Figura 18. Visualización de los tópicos no etiquetados por periodos de tiempo (semana, quincena, meses). *53*
Figura 19. Palabras que representan al tópico 2, tópico 1 y tópico 5 en los periodos continuos periodo 0, periodo 1 y periodo 2. ..... *54*
Figura 20. El botón Gráfica generara tanto las gráficas sobre el número de noticias en los tópicos por periodo, como la visualización por nube de palabras *55*
Figura 21. Botón que genera la gráfica de continuidad entre tópicos de diferentes periodos. *55*
Figura 22. Botón que muestra las palabras que son similares en los dos tópicos de diferentes periodos. *55*

Figura 23. Resultado de LDA sobre el número de noticias en las que aparecen los tópicos del periodo 0. 60
Figura 24. Gráfica que determina la importancia de los tópicos generados para el periodo 0 mediante el número de noticias en las que se encuentran en cada tópico. 61
Figura 25. Gráfica que determina la importancia de los tópicos generados para el periodo 1 mediante el número de noticias en las que se encuentran en cada tópico. 61
Figura 26. Gráfica que determina la importancia de los tópicos generados para el periodo 2 mediante el número de noticias en las que se encuentran en cada tópico. 62
Figura 27. Gráfica que determina la importancia de los tópicos generados para el periodo mediante el número de noticias en las que se encuentran en cada tópico. 62
Figura 28. Gráfica que determina la importancia de los tópicos generados para el periodo 4 mediante el número de noticias en las que se encuentran en cada tópico. 63
Figura 29. Número de noticias del tópico 9 del periodo 1, tópico 1 del periodo 2, tópico 6 del periodo 3 similares al tópico inicial 0 del periodo 0, con valor de similitud <= 0.85.....65
Figura 30. Número de noticias del tópico 2 del periodo 1, tópico 8 del periodo 2, tópico 9 del periodo 3 similares al tópico inicial 6 del periodo 0, con valor de similitud <= 0.85.....66
Figura 31. Número de noticias del tópico 1 del periodo 3, tópico 2 del periodo 4 similares al tópico inicial 5 del periodo 2, con valor de similitud <= 0.85 67
Figura 32. Número de noticias del tópico 2 del periodo 4 similar al tópico inicial 0 del periodo 3, con valor de similitud <= 0.85.....68
Figura 33. Lista de palabras entre tópicos con similitud <= 0.85, tópico 0 del periodo 0 con el tópico 9 del periodo 1.....69
Figura 34. Visualización de la importancia mediante nube de palabras para un periodo P. donde se pude determinar que en ese lapso de tiempo se mantubo la palabra "cantante" y "mexico" como las mas representativas de los tópicos del periodo P, mientras que las palabras "casa", "pais", "latina" y "mal" son las menos importantes.....71
Figura 35. Resultados de las palabras representativas de los tópicos sobre todas las noticias con 4 tópicos de un periodo.....76
Figura 36. Comparativo en % de error en LDA, entre el número de noticias en la base de datos y el número de noticias generado por LDA con 4 tópicos.....78
Figura 37. Comparativo en % de error en LDA, entre el número de noticias en la base de datos y el número de noticias generado por LDA con 3 tópicos.....79
Figura 38. Recursos de memoria RAM sin utilizar el programa E.....81
Figura 39. Recursos de memoria RAM utilizados por parte del programa E. 81
Figura 40. Uso de memoria RAM al implementar el índice invertido dentro del programa E. 82
Figura 41. Liberación de memoria al término de la implementación del programa (E), liberando el espacio.......83

Tablas.

Tabla 1. Comparación entre el estado del arte y el sistema de la información de tópicos más importantes generados en medios sociales. *23*
Tabla 2. Valores de similitud del tópico 9 del periodo 1, tópico 1 del periodo 2, tópico 6 del periodo 3 con respecto al tópico inicial 0 del periodo 0. ..... *65*
Tabla 3. Valores de similitud del tópico 2 del periodo 1, tópico 8 del periodo 2, tópico 9 del periodo 3 con respecto al tópico inicial 6 del periodo 0. ..... *67*
Tabla 4. Valores de similitud del tópico 1 del periodo 3, tópico 2 del periodo 4 con respecto al tópico inicial 5 del periodo 2, ... *68*
Tabla 5. Valores de similitud del tópico 2 del periodo 4 con respecto al tópico inicial 0 del periodo 3 .*68*
Tabla 6. Total de noticias por sección según la base de datos. *75*
Tabla 7. Resultado de LDA para 4 tópicos *76*
Tabla 8. Temas etiquetados manualmente que indican la el contenido de cada tópico *76*
Tabla 9. Generalización de Noticias en la base de datos y LDA *77*
Tabla 10. Prueba 2 de LDA con tres tópicos *78*
Tabla 11. Generalización de Noticias en la base de datos y LDA con 3 tópicos *79*

Glosario.

Twitter.

Es un servicio de microblogging que permite enviar mensajes de texto plano de corta longitud, con un máximo de 140 caracteres, llamados tweets, que se muestran en la página principal del usuario. Los usuarios pueden suscribirse a los tweets de otros usuarios a esto se le llama "seguir" y a los usuarios abonados se les llama "seguidores" o "followers"[1].

Noticia.

(Del lat. notitĭa).

1. f. Noción, conocimiento. 2. f. Contenido de una comunicación antes desconocida. Dar noticia de un acuerdo. 3. f. Hecho divulgado. Noticia triste. 4. f. p. us. Divulgación de una doctrina.[2].

RSS.

Es una forma muy sencilla con la cual se pueden recibir, directamente en un ordenador o en una página web online (a través de un lector RSS) información actualizada sobre las páginas web favoritas, sin necesidad de que se tenga que visitarla una a una. Esta información se actualiza automáticamente, sin que se realice otro proceso manual[3].

Noticias RSS.

Las noticias RSS se refieren el contenido, artículos o noticias que publican las páginas web y que se distribuyen a los lectores a través de los canales RSS. Esto incluiría desde las noticias de un periódico online favorito hasta los artículos del blog de tu mejor amigo. La palabra original en inglés es News RSS, por lo que muchas páginas se referirán a las noticias por su nombre en inglés News RSS[4].

---

[1] Twitter. https://twitter.com/

[2] Noticia, http://lema.rae.es/drae/?val=Noticia.

[3] RSS, http://www.rss.nom.es/

[4] RSS, http://www.rss.nom.es/

Stop words.

Es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. son filtradas antes o después del procesamiento de datos en lenguaje natural (texto)[5].

Tópico o tema del inglés (topic).

Un tema de discusión o conversación.

Lingüística: Una palabra o frase en una oración, por lo general suministrá información a partir de un discurso anterior o conocimiento compartido[6].

Componentes léxicos (tokens).

Unidad mínima de información que significa algo a la hora de compilar; concepto de palabra; las fases de un lenguaje constan de cadenas de componentes léxicos[7].

[5] Stop wods, http://es.wikipedia.org/wiki/Palabra_vac%C3%ADa

[6] **Tópico o tema** http://es.thefreedictionary.com/tema

[7] Componentes léxicos (tokens). http://10380054.galeon.com/u5.htm

Capítulo 1: Introducción.

En este capítulo se describe el problema a resolver, el alcance que se pretende lograr al término de los capítulos siguientes, así como los objetivos a cumplir con el resultado esperado del trabajo de tesis.

1.1 Planteamiento del problema.

La dinámica de las noticias publicadas en los medios de información y los medios sociales (RSS, Facebook, Twitter[1], LinkedIn, Google+, Flicker, YouTube etc.) es una forma de ver el comportamiento de la sociedad, en ella los usuarios de ambos medios se informan y pueden realizar una comunicación, mediante la cual se llegan a distintos destinatarios ya sean amigos, conocidos o personas de interés con las que desean tener un contacto más cercano o simplemente por accidente se conoce la publicación.

La relación de la información en los medios es utilizando *Twitter* como red social, dado que se pueden hacer comentarios en ambos sentidos (a partir de las noticias o de los tweets), la importancia de la relación puede considerarse en principio ya sea por la frecuencia de las noticias que aparecen en los tópicos y de igual forma por los tweets. Los tópicos que se generan de las noticias o tweets, se pueden mantener a través del tiempo, demostrando así que ellos tienen otra forma de importancia, la cual es soportada por la cantidad de usuarios que publican información (noticias y tweets) sobre temas (tópicos) que ellos consideran importantes durante un periodo de tiempo. Con ello se observa que esto es una forma en la cual los tweets están relacionados con las noticias generadas en medios de información.

Otra forma de medir tópicos importantes es que muchos tweets son publicados por uno o más seguidores (*followers*) que hablan sobre el mismo tópico en cuestión, señalando la dificultad que se tiene al momento de poder interpretar los 140 caracteres con los que cuenta un comentario en la línea de tiempo de Twitter.

El presente trabajo de tesis es una alternativa a la detección de los tópicos más importantes de interés para los usuarios tanto de medios informativos como de redes sociales, aplicando ramas de la Ciencias de la Computación como:

- ✓ Bases de datos de noticias.
- ✓ Procesos ETL de Minería de Datos [1] .
- ✓ Normalización de texto, Procesamiento del Lenguaje Natural [2].
- ✓ Visualización de la Información [3].

Lo cual se describirá en la siguiente sección.

1.2 Objetivo general.

Construir una herramienta que permita visualizar los tópicos más importantes que existen en medios de información y sociales, junto con la duración en que prevalecen en estos medios.

1.3 Objetivos particulares.

- Obtener una colección de documentos (noticias) para aplicarles técnicas de procesamiento del lenguaje natural y separar los términos significativos.
- Aplicar a los términos significativos un modelo de clasificación para generar los k-tópicos.
- Definir los tópicos más importantes, junto con un modelo de similitud para hallar tópicos similares.
- Diseñar el software que permita capturar los parámetros para hallar los tópicos más importantes durante periodos de tiempo establecidos por el usuario y un valor de similitud de interés.
- Diseñar una visualización que permita observar el comportamiento de los tópicos más importantes de la colección adquirida y procesada, así como, porqué ciertos tópicos de diferentes periodos son similares.

1.4 Justificación.

Detectar los tópicos más importantes en medios sociales es de gran utilidad debido a su velocidad de propagación (sin validar su verosimilitud), y que de manera casi inmediata se da a conocer el surgimiento de problemáticas en diferentes campos de la ciencia, tecnología, dispersión de enfermedades o algún problema existente en el país, además pueden ser detectados en medios informativos. El conocimiento de estos tópicos puede ayudar a evitar seguir utilizando tecnología con problemas, productos que pueden ser nocivos al ser humano, realizar o evitar actividades que puedan afectar la economía de un país, entre otros. Por este motivo surge la idea de localizar los tópicos importantes y mantener informadas a las personas que estén interesadas en ellos. Basta recordar problemas tecnológicos o de otra índole, por ejemplo:

- Fallas en automóviles con frenos dañados [4].
- iPhones que se doblan [5].
- Posibles casos de Ébola en Norteamérica [6].
- Influenza H1N1 [7].
- Delitos frecuentes en algunas áreas del Distrito Federal y Estado de México, operativos en Tláhuac-Chalco [8].

Para dar una definición de cómo identificar los tópicos más importantes, es utilizando un nivel de importancia definido (cantidad de noticias por tópico), una posible forma de visualizar los tópicos y su composición (es mediante las palabras que generaron el tópico, las cuales son escritas tanto por los usuarios de Twitter como los generadores de las noticias). La visualización debe ayudar a determinar fácilmente lo anterior mencionado. El ejercicio iniciara con noticias como: entretenimiento, política, ciencia, tecnología u otro tema en particular.

Utilizando Twitter como un impacto en medios sociales (periódicos, noticieros, etc.), se da un soporte más amplio a los tópicos que son generados por las palabras en dichos medios, por lo cual muchos de los usuarios publican comentarios con relación a dicha noticia, pero aun así no siempre se da continuidad a las noticias y por dicha razón un tópico interesante se llega a perder en el paso del tiempo o hasta la llegada de una noticia sobre él mismo tema de interés.

Capítulo 2: Estado del arte.

En este capítulo se describen brevemente algunos trabajos relacionados a las tendencias en Twitter y como se clasifican, para así determinar los temas sobre los que se hablan más en la red social y noticias.

2.1. Trabajos actuales sobre tópicos en Twitter.

Los trabajos que se investigaron están relacionados a la clasificación de algunos conjuntos de tweets obtenidos previamente y mediante clasificadores probabilísticos como los siguientes: Naive Bayes(NB), K-Nearest Neighbour (K-NN) , Term frequency · Inverse document frequency (TF·IDF) y LanguageModel (LM) [9], se logra determinar el tipo de tópicos al que pertenecen mediante algunos temas que se tenían clasificados como importantes como pueden ser: *art & design, books, charity & deals, fashion, food & drink, health, humor, music, politics, religion, holidays & dates, science, sports, technology, business, tv & movies, other news, and other* [10]. Después de este procesamiento se logra realizar una gráfica de visualización de las estadísticas sobre los tópicos.

2.1.1 Tendencias en Twitter.

Las tendencias son links que se muestran en la página de Twitter y se localizan en la parte izquierda de la página de inicio de cada usuario.

Las tendencias se determinan mediante un algoritmo el cual es propiedad de Twitter y son personalizadas para cada usuario mediante las cuentas que sigue y la ubicación en la que se encuentran. Este algoritmo identifica temas que son inmediatamente populares, más allá de temas que han sido populares por un largo periodo de tiempo o comunes diariamente, para así ayudar a descubrir el contenido más actual y emergente dentro de los temas en discusión de más relevantes para los usuarios.

Se pueden alterar las tendencias dependiendo de la ubicación que el usuario desee obtener o donde se localice actualmente[8].

Las tendencias que aparecen en las cuentas de usuario son 10 links importantes que indican personas, empresas, celebridades o sucesos que se generan por medio de un #hashtag o algún comentario referente a ellos (ver Figura 1).

Figura 1. Tendencias adquiridas directo de Twitter personal.

2.1.2 *"Twitter Trending Topic Classification".*

En Kathy Lee et al. [10], se comentan que en Junio de 2011, los usuarios publican 200 millones de tweets por día. Los clasifican en 18 categorías ya generadas. Cuenta con dos enfoques: "Bolsa de Palabras" la cual usa clasificadores de texto y "Clasificación basada en red" que enumera los mejores usuarios que se relacionan a un tema seleccionado.

Mediante la "Bolsa de palabras" los tópicos son clasificados mediante un clasificador de pesos TF•IDF y algoritmos de clasificación Naive Bayes, Logistic Regression, k-Nearest Neighbor, C5.0 y.

$$P(c \mid d) \;\; \alpha \;\; P(c) \prod_{1 \le K \le n_d} P(t_k \mid c)$$

Donde $P(c \mid d)$ es la probabilidad de un documento d iniciando en la clase c.

$P(c)$ es la probabilidad *apriori* de la ocurrencia de un documento en la clase c.

[8] Sobre las tendencias en Twitter, https://support.twitter.com/articles/349215-sobre-las-tendencias#location

$P(t_k \mid c)$ es la probabilidad condicional de que el término t_k ocurre en un documento de la clase *C*.

Un documento *d* en nuestro caso es la tendencia o relación que tienen los tweets en cada tema de tendencia. Para su desarrollo se utilizó la herramienta WEKA[9] y su gran cantidad de algoritmos, mediante los cuales se logró realizar la clasificación con el enfoque de bolsa de palabras.

2.1.3 *"Developing a Hierarchical Multi-Label Classifier for Twitter Trending Topics".*

En Jinan Fiaidhi et al. [9], enseñan que mediante la herramienta LingPipe[10] se puede clasificar con precisión mediante algunos de los algoritmos con los que cuenta.

El trabajo se realizó coleccionando una cantidad considerable de tweets mediante el Twitter Streaming API[11] y después utilizando 12 clases (política, educación, salud, mercadotecnia, música, noticias y medios, deportes y recreación, cómputo y tecnología, comida, familia y otros) se logró obtener la clasificación de los tweets en una de las clases mencionadas. Utiliza una máquina de aprendizaje mediante sinónimos que mejora la clasificación de nuevos documentos. El clasificador jerárquico de multietiqueta se realiza con el API de LingPipe el cual cuenta con diversos clasificadores como Naive Bayes, LanguageModel, K Nearest Neighbor, TF·IDF.

Los resultados del artículo muestran una gráfica de pastel en la cual se determinan los pesos sobre las diferentes clases en las cuales se clasificaron los tweets (ver Figura 2).

[9] WEKA, http://www.cs.waikato.ac.nz/ml/weka/

[10] LingPipe, http://alias-i.com/lingpipe/

[11] Twitter Streaming API, https://dev.twitter.com/docs/streaming-apis

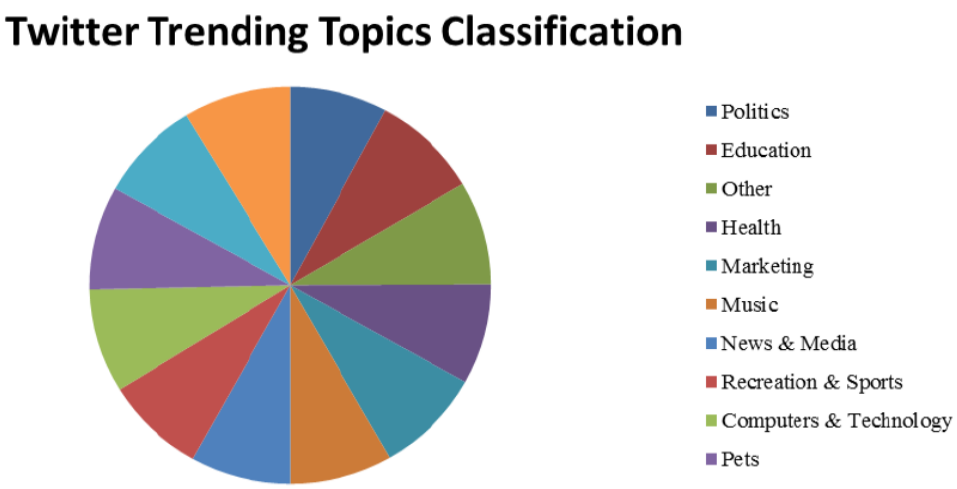


Figura 2. "Twitter Trending Topics Classification" [9].

2.1.4 *"Beyond Trending Topics: Real-World Event Identification on Twitter".*

En Hila Becker et al. [11], se analizan las corrientes de los mensajes en Twitter basándose en la estadística agregada de grupos de mensajes sobre temas similares del mundo real.

La clasificación de los tweets es asumida por un algoritmo incremental de clúster, el cual calcula similitud entre todos los mensajes M_i y los clúster C_j, está dada por.

$$\sigma(M_i, C_j)$$

Los mensajes son representados mediante pesos de TF·IDF y usan la similitud del coseno para relacionarlos a cada clase existente. Mientras que la obtención de los tweets fue hecha utilizando el Twitter API durante el periodo de febrero de 2010.

Realizaron pruebas exploratorias en las cuales se utilizó un clasificador de tipo máquina de soporte vectorial, pero al final se decidió usar un enfoque diferente, el cual es el clasificador de Naive Bayes que se usó para generar un gran clúster y luego mediante el uso de TF·IDF poder distinguir entre tweets que son eventos y los que no lo son.

2.1.5 *"TwitterStand: News in tweets"*.

En Sankaranarayana et al. [12], se obtienen los tweets relacionados, dando como resultado un sistema de tópicos de noticias el cual no cuenta con reporteros, sino con usuarios que publican tweets relevantes siendo ellos herramientas que el sistema ocupa como sus ojos y oídos para así conocer de las noticias que surgen mediante los tweets que generan.

La visualización que se obtiene es geográfica ya que la localización de los usuarios es relacionada con las noticias que publican mediante tweets que describen hechos que suceden en cierta localidad específica.

Aquí realizan un clasificador Naive Bayes el cual es aplicado al cuerpo del tweet, determinando si es noticia o basura para la colección final; el algoritmo que se realizó fue mediante un agrupamiento de tipo clúster al cual se le asigna un grupo de palabras claves, mientras que los términos de cada tweet son ponderados mediante un clasificador TF·IDF, recolectando un promedio del tiempo en el que fueron publicados y la similitud se mide bajo el coseno.

2.1.5 *"Multi-Modal Multi-Correlation Person-Centric News Retrieval".*

En Zechao Li et al. [13], se describe la estructura de una noticia, se clasifican las noticias por la similitud de las imágenes que publican los periódicos; por lo tanto se puede conocer si dos imágenes de diferentes noticias tienen relación cercana a la misma persona en la que se describe en el texto de la publicación.

Además de la utilización de imágenes, se describe el uso de clasificación textual la cual utiliza diferentes secciones de la noticia como son: título, resumen y el cuerpo de la noticia mediante las cuales se puede determinar la similitud entre las diferentes noticias. Ellos identifican que TF·IDF es todavía la técnica dominante para representar un documento y la similitud del coseno es la métrica de similitud más utilizada generalmente para la medición sobre la matriz de noticias.

2.1.6 "Comparing Twitter and Traditional Media using Topic Models".

En Wayne Xin Zhao et al. [14], se realiza un estudio de dos distintos medios de información, noticias y tweets. En el artículo se utiliza el modelo de tópicos Latent Dirichlet Allocation (LDA) para encontrar empíricamente una relación entre los tópicos de una colección de noticias ya existentes sobre *The New York Times* (NYT) y Twitter. Por la dificultad que plantea analizar la poca información de un tweet, los autores modificaron el modelo original LDA para poder utilizar una variante que utiliza una matriz más reducida de información. El modelo es utilizado para encontrar los 100 tópicos más comunes en Twitter y después de ser encontrados son etiquetados manualmente para entender los temas a los que pertenecen, mediante estos tópicos son comparados empíricamente con los tópicos existentes encontrados por el modelo LDA original que le es aplicado a las noticias del NYT utilizando la medida *JS-divergence* para determinar la semejanza entre ambos tópicos.

En el resultado muestra los temas semejantes entre Noticia y Twitter además de una tabla que muestra las palabras que lo representan y una gráfica de los temas con mayor impacto en los dos medios.

El comparativo entre todos los artículos que se encuentran en el estado del arte se puede observar en una descripción resumida dentro de la cual se muestran ventajas y desventajas (ver Tabla 1).

Tabla 1. Comparación entre el estado del arte y el sistema de la información de tópicos más importantes generados en medios sociales.

Articulo	Clasificador y Estrategias	Programa de Clasificación	Enfoque	Similitud	Ventajas	Desventajas
"Twitter Trending Topic Classification"	Naive Bayes, Logistic Regression, k-Nearest Neighbor, C5.0 y TF•IDF	WEKA	Bolsa de palabras	-	Programa de clasificación supervisado existente	Solo clasifica
"Developing a Hierarchical Multi-Label Classifier for Twitter Trending Topics"	Naive Bayes, LanguageModel, K Nearest Neighbor, TF·IDF	LingPipe	Bolsa de palabras	-	Programa de clasificación supervisado existente	Solo clasifica
"Beyond Trending Topics: Real-World Event Identification on Twitter"	TF•IDF y Naive Bayes	WEKA	Algoritmo incremental de clúster	Coseno	Programa de clasificación supervisado Existente y pesos TF	Solo clasifica
"TwitterStand: News in tweets".	TF•IDF y Naive Bayes, PLN, Part-Of-Speech, Named-Entity Recognition	-	Agrupamiento de clúster	Coseno	Realiza una clasificación y geolocalización de noticias.	Necesario GPS para la localización
"Multi-Modal Multi-Correlation Person-Centric News Retrieval".	TF•IDF	-	Probabilistic Matrix Factorization	Coseno	Clasificación por pesos TF y reconocimiento facial	Clasificación principal por reconocimiento facial no textual

"Comparing Twitter and Traditional Media using Topic Models"	LDA, Twitter LDA	-	Modelo tópicos	JS-divergence	Uso de dos clasificadores no supervisados, una variante especial para Twitter y detección de similitud entre noticias y tópicos	Clasificación con respecto a toda la colección y similitud empírica entre noticias y tópicos
"Sistema de visualización de la información de tópicos más importantes generados en medios sociales"	LDA, TF•IDF	Propio	Modelo y similitud entre tópicos	Distancia entre Palabras por posicionamiento (Propio)	Uso de un clasificador no supervisado durante periodos de tiempo establecidos. Modelo de similitud de periodos por posición de palabras, para determinar su comportamiento.	Uso de recursos y tiempo de resultados.

Capítulo 3: Marco teórico.

En este capítulo se describen detalladamente los conceptos, algoritmos y herramientas que se pueden utilizar para el mejor entendimiento y desarrollo del trabajo de tesis.

3.1 Conceptos generales.

El trabajo de tesis se sustenta por medio de los conceptos que se muestran a continuación, en los cuales podemos observar el beneficio de la recuperación de la información que en nuestro caso son noticias, su manejo textual mediante algunos métodos del procesamiento del lenguaje natural y su clasificación, con estos pasos se podrá determinar el impacto que tiene una noticia en Twitter.

3.1.1 Recuperación de la información (IR)

Según Christopher D. Manning, et al. [15], en su libro, la recuperación de la información (IR) es encontrar el material (generalmente documentos) de naturaleza no estructurada (generalmente texto), que satisface una necesidad de información desde grandes colecciones de documentos (generalmente almacenados en los ordenadores).

3.1.2 Procesamiento del lenguaje natural (PLN).

El recurso más importante que posee la raza humana es el conocimiento, que es lo que se valida de la información. En la época actual de información, el manejo eficiente de este conocimiento depende el uso de todos los demás recursos como son los naturales, industriales y humanos.

Durante toda la historia de humanidad el conocimiento, en su mayor parte se comunica, se guarda y se maneja en la forma de lenguaje natural –griego, latín, inglés, español, etc. La época actual no es ninguna excepción: el conocimiento sigue existiendo y creándose en la forma de documentos, libros, artículos, aunque éstos se guardan en forma electrónica, o sea digital. El gran avance es que en esta forma, las computadoras ya pueden ser una ayuda enorme en el procesamiento de este conocimiento.

Sin embargo, lo que es conocimiento para nosotros –los seres humanos– no lo es para las computadoras. Son los archivos, unas secuencias de caracteres, y nada más. Una computadora puede copiar tal archivo, respaldarlo, transmitirlo, borrarlo –como un burócrata que pasa los papeles a otro burócrata sin leerlos. Pero no puede buscar las respuestas a las preguntas en este texto, hacer las inferencias lógicas sobre su contenido, generalizar y resumirlo –es decir, hacer todo lo que las personas normalmente hacemos con el texto. Porque no lo puede entender.

Para combatir esta situación, se dedica mucho esfuerzo, sobre todo en los países más desarrollados del mundo, al desarrollo de la ciencia que se encarga de habilitar a las computadoras a entender el texto. Esta ciencia, en función del enfoque práctico versus teórico, del grado en el cual se espera lograr la comprensión y de otros aspectos tiene varios nombres: procesamiento de lenguaje natural, procesamiento de texto, tecnologías de lenguaje, lingüística computacional. En todo caso, se trata de procesar el texto por su sentido y no como un archivo binario.

El esquema general de la mayoría de los sistemas y métodos que involucran el procesamiento de lenguaje es el siguiente:

- Primero, el texto no se procesa directamente sino se transforma en una representación formal que preserva sus características relevantes para la tarea o el método específico (por ejemplo, un conjunto de cadenas de letras, una tabla de base de datos, un conjunto de predicados lógicos, etc.).
- Luego, el programa principal manipula esta representación, transformándola según la tarea, buscando en ella las subestructuras necesarias, etc.
- Finalmente, si es necesario, los cambios hechos a la representación formal (o la respuesta generada en esta forma) se transforman en el Lenguaje Natural.

Entre las tareas principales del procesamiento de Lenguaje Natural se puede mencionar:

- Recuperación de Información
- Interfaces en Lenguaje Natural
- Traducción automática.[12]

3.1.3 Minería de Datos.

La Minería de Datos se refiere a la extracción de conocimiento al hacer "minería" de grandes cantidades de datos. La Minería de Datos debería haber sido nombrado de manera más apropiada "Minería de Datos de Conocimiento", que es por desgracia un poco largo. "La Minería de Conocimiento", es un término más corto, puede no reflejar el énfasis en la minería de grandes cantidades de datos. Sin embargo, la minería es un término para caracterizar el proceso de hallar un pequeño conjunto de pepitas preciosas de una gran cantidad de materia prima (ver Figura 3).

Figura 3. Minería de conocimiento [1].

Muchas personas tratan la Minería de Datos como sinónimo de otro término utilizado popularmente, “Descubrimiento de Conocimiento de los datos o KDD”. Alternativamente,

[12] ¿Qué es Procesamiento del Lenguaje Natural?, Asociación Mexicana para el Procesamiento del Lenguaje Natural, http://www.cicling.org/ampln/NLP.htm

otros ven la Minería de Datos simplemente como un paso esencial en el proceso de descubrimiento de conocimiento. Descubrimiento de conocimiento como un proceso se representa en la Figura 4 y se compone de una secuencia iterativa de los pasos siguientes:

1. Limpieza de datos (eliminar el ruido y los datos inconsistentes).
2. Integración de datos (múltiples fuentes de datos se pueden combinar).
3. Selección de datos (los datos pertinentes a la tarea de análisis se recuperan de la base de datos).
4. Transformación de datos (los datos se transforman o consolidan en formas apropiadas para la minería mediante la realización de operaciones de resumen o de agregación).
5. Minería de Datos (un proceso esencial donde se aplican técnicas de Inteligencia Artificial para extraer patrones de datos).
6. Evaluación del patrones (identificar los patrones verdaderamente interesantes que representan el conocimiento basado en algunas medidas de Intereses).
7. Presentación del conocimiento (utilizan técnicas de Visualización de la Información y representación del conocimiento extraído para el usuario) [1].

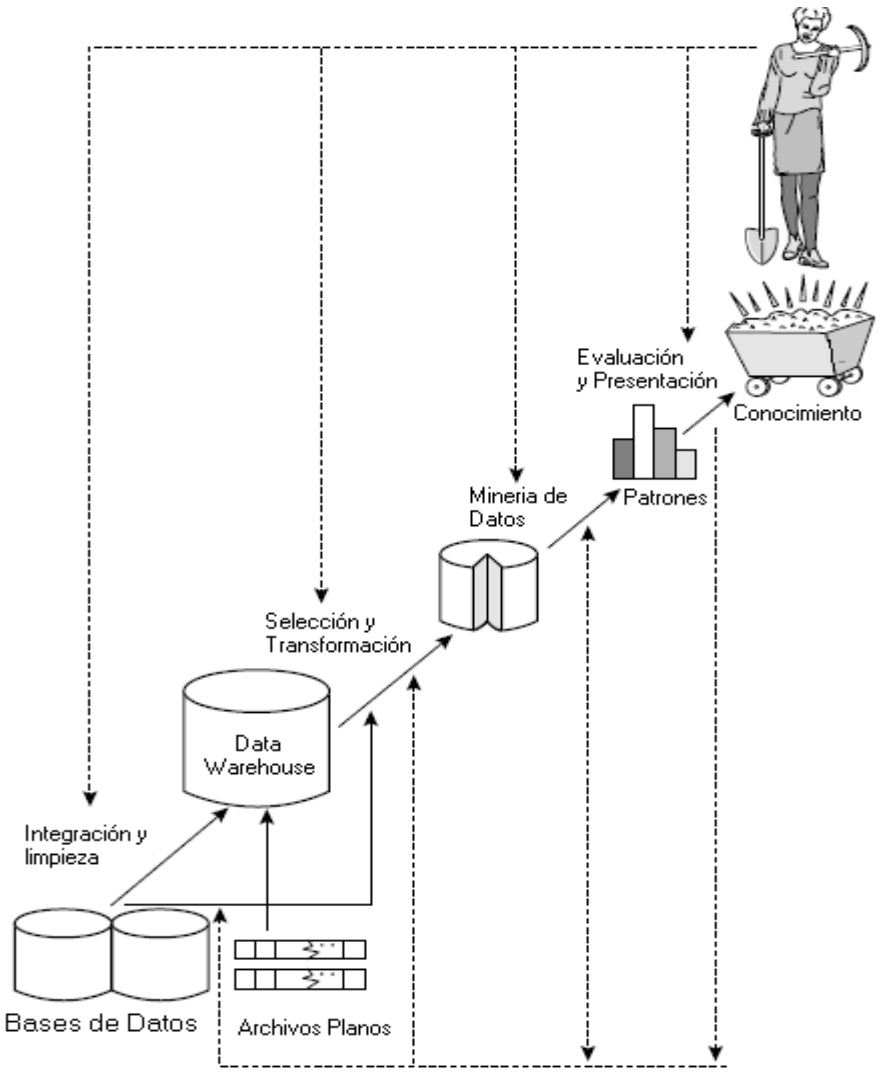


Figura 4. La minería de datos como un paso en el proceso de descubrimiento de conocimiento [1].

3.1.4 Visualización de la Información.

La Visualización es un método de computación que transforma los datos numéricos y categóricos a símbolos geométricos y permite a los investigadores observar con la vista sus simulaciones y cálculos, ofreciendo un método para ver lo que no se encuentra al alcance de otra representación de sus resultados, enriquece el proceso del descubrimiento científico fomentando ideas profundas e inesperadas.

La Visualización es el proceso de transformación de los datos, la información y el conocimiento en presentaciones gráficas para apoyar tareas tales como el análisis de datos, la explicación de la información, la predicción de tendencias, detección de patrones, detección de ritmo, y así sucesivamente. En este sentido, la Visualización es el proceso de cristalización de una imagen mental, o un proceso de valor añadido de la comunicación entre los usuarios y los datos. La Visualización es también un método de análisis de datos que supera a los métodos numéricos, estadísticos, donde las relaciones se mantienen durante el análisis visual de datos [3].

3.2 Métodos de clasificación.

En esta sección se hablará de algunos algoritmos que ayudan a la clasificar los textos mediante métodos probabilísticos, como son: Naive Bayes (NB), K-Nearest Neighbour (K-NN), Term frequency·Inverse document frequency (TF·IDF) y algunos metodologías del PLN como (MTD).

3.2.1 Naive Bayes (NB).

Un clasificador de Naive Bayes asume que la presencia o ausencia de una característica particular, no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable.

Por ejemplo, una fruta puede ser considerada como una manzana si es roja, redonda, y con un diámetro de 7 cm.

Un clasificador de Naive Bayes considera que cada una de estas características contribuye de manera independiente a la probabilidad de que un objeto sea clasificado dentro de una clase, independientemente de la presencia o ausencia de las otras características.

Para otros modelos de probabilidad, los clasificadores de Naive Bayes se pueden entrenar de manera muy eficiente en un entorno de aprendizaje supervisado. Una ventaja del clasificador de Naive Bayes es que sólo se requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros (las medias y las varianzas de las variables) necesarias para la clasificación. Como las variables independientes se asumen, sólo es necesario determinar las varianzas de las variables de cada clase y no toda la matriz de covarianza[13]

Usando el teorema de Bayes se escribe:

$$P(C \mid F_1,.....,F_n) = \frac{P(c)\, p(F_1,.....,F_n \mid C)}{P(F_1,.....,F_n)}$$

C, con un pequeño número de resultados (o clases). Esta variable está condicionada por varias variables independientes desde F_1 a F_n.

3.2.2 K-Nearest Neighbors (K-NN).

Es un método de clasificación supervisada (Aprendizaje, estimación basada en un conjunto de entrenamiento y prototipos) que sirve para estimar la función de densidad $F(x \mid C_j)$ de las predictoras x por cada clase C_j.

Éste es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad "a posteriori" de que un elemento x pertenezca a la clase C_j a partir de la información proporcionada por el conjunto de prototipos. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras.

En el reconocimiento de patrones, el algoritmo K-NN es usado como método de clasificación de objetos (elementos) basado en un entrenamiento mediante ejemplos cercanos en el

[13] Naive Bayes, http://es.wikipedia.org/wiki/Clasificador_bayesiano_ingenuo

espacio de los elementos. K-NN es un tipo de "*Lazy Learning*", donde la función se aproxima solo localmente y todo el cómputo es diferido a la clasificación[14].

3.2.3 Term frequency · Inverse document frequency (TF·IDF).

TF·IDF (del inglés *Term frequency · Inverse document frequency*), frecuencia de término · frecuencia inversa de documento (o sea, la frecuencia de ocurrencia del término en la colección de documentos), es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección. Esta medida se utiliza a menudo como un factor de ponderación en la recuperación de información y la minería de texto. El valor TF·IDF aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero es compensada por la frecuencia de la palabra en la colección de documentos, lo que permite manejar el hecho de que algunas palabras son generalmente más comunes que otras. Variaciones del esquema de peso TF·IDF son empleadas frecuentemente por los motores de búsqueda como herramienta fundamental para medir la relevancia de un documento dada una consulta del usuario, estableciendo así una ordenación o *ranking* de los mismos. TF·IDF puede utilizarse exitosamente para el filtrado de las denominadas *stop words* (palabras que suelen usarse en casi todos los documentos), en diferentes campos como la clasificación y resumen de texto.

Una de las funciones de *ranking* más sencillas se calcula como la suma de los valores TF·IDF de cada término de la consulta[15].

TF·IDF es el producto de dos medidas, frecuencia de término y frecuencia inversa de documento. Existen varias maneras de determinar el valor de ambas. En el caso de la frecuencia de término *TF (t, d),* la opción más sencilla es usar la *frecuencia bruta* del término *t* en un documento *d*, o sea, el número de veces que el término *t* ocurre en un documento *d*.

[14] K-NN, http://es.wikipedia.org/wiki/Knn

[15] Tf-Idf, http://es.wikipedia.org/wiki/Tf-idf

Si denotamos la frecuencia bruta de *t* por *f* (*t*,*d*), entonces el esquema *TF* simple es *tf* (*t*, *d*) = *f* (*t*,*d*).

$$TF(t,d) = \frac{f(t,d)}{\max\{f(w,d) : w \in d\}}$$

La frecuencia inversa de documento es una medida en la que se determina si el término es común o no, en la colección de documentos. Se obtiene dividiendo el número total de documentos por el número de documentos que contienen el término, y se toma el logaritmo de ese cociente:

$$IDF(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Donde:

- $|D|$: cardinalidad de D, o número de documentos d en la colección.
- $|\{d \in D : t \in d\}|$: número de documentos donde aparece el término t. Si el término no está en la colección se producirá una división-por-cero. Por lo tanto, es común ajustar esta fórmula a $1 + |\{d \in D : t \in d|$.

Matemáticamente, la base de la función logaritmo no es importante y constituye un factor constante en el resultado final.

Luego, TF·IDF se calcula como:

$$TF \cdot IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

3.2.4 Matriz término documento (MTD).

Entonces, nuestros objetos que estamos considerando son documentos (textos). Tenemos palabras (términos) como características de esos documentos, cada palabra tiene un valor TF·IDF en cada documento. Eso quiere decir que a cada documento le corresponde un vector de valores TF·IDF de las palabras.

También podemos representar toda esa información como una matriz. Esa matriz se llama "matriz término-documento". Se presenta un ejemplo en la Ilustración Figura 5.

	Doc_1	Doc_2	Doc_3	Doc_4...
$Palabra_1$	0.3	0	0	0.02
$Palabra_2$	0.7	0.01	0	0.5
$Palabra_3$	0	0	0	0
$Palabra_4$	0	0	2.2	0
$Palabra_5$	1.2	0	0	0
...				

Figura 5. Ejemplo de matriz término documento [2].

Los documentos Doc1...DocN representan los documentos en nuestra colección, las palabras Palabra1...PalabraM representan todas las palabras que se encuentran en esos documentos ordenadas de alguna manera. Los valores en el cuadro corresponden a los supuestos valores TF·IDF en la colección.

Nótese que típicamente el cuadro es muy disperso, es decir tiene muchos ceros. Se recomienda utilizar el índice invertido para su representación, el cual se utilizó en este trabajo (ver sección 4.1.5). El índice invertido consiste en conversión del cuadro en una lista. Por ejemplo, la lista invertida para la ilustración anterior sería:

(Doc1, Palabra1, 0.3), (Doc1, Palabra2, 0.7), (Doc1, Palabra5, 1.2), (Doc2, Palabra2, 0.01), (Doc3, Palabra4, 2.2), (Doc4, Palabra1, 0.02), (Doc4, Palabra2, 0.5).

En este caso no es necesario mantener en la lista ningún elemento para los valores iguales a cero. La matriz inicial y el índice invertido contienen la misma información. Es un procedimiento estándar que siempre debe aplicarse a las matrices dispersas [2].

3.2.5 Latent Dirichlet Allocation (LDA).

Es un modelo probabilístico generativo de un corpus. La idea básica es que los documentos se representan como mezclas aleatorias sobre temas latentes, donde cada tema se caracteriza por una distribución de palabras.
Formalmente se describen los siguientes términos.

Una palabra es la unidad básica de datos discretos que representan a un documento, dichas palabras son elementos de un vocabulario indexado por $\{1,...,V\}$, el vocabulario contiene las palabras que se encuentran en todos los documentos, nosotros representamos palabras usando vectores de unidades base que tienen un solo elemento igual a uno y todos los demás componentes iguales a cero. Así, utilizando superíndices para denotar componentes, de v_{th} palabra en el vocabulario es representada por un V-vector w de tal manera que $w^v=1$ y $w^u=0$ para $u \neq v$.

Un documento es una secuencia de *N* palabras denotadas por $W = (w_1, w_2, ..., w_N)$, donde w_n es el n_{th} palabra en la secuencia.

Un corpus es una colección de M documentos denotados por $D = (W_1, W_2, ..., W_M)$.
Se desea encontrar un modelo probabilístico de un corpus que no sólo asigna alta probabilidad a los miembros del corpus, pero también asigna alta probabilidad a otros documentos "similares" [16].

El modelo LDA es representado en un modelo grafico de probabilidad en la Figura 6. Como la figura deja claro, hay tres niveles a la representación LDA. Los parámetros α y β son parámetros de sobre el corpus, asume ser muestreado una vez en el proceso de generación de un corpus. Las variables θ_d son variables sobre el documento {noticias o tweets}, muestra una vez por documento. Por último, las variables Z_{dn} y W_{dn} son variables a nivel de palabra y se toman muestras de una vez por cada palabra en cada documento.

α es el parámetro de Dirichlet, previo sobre las distribuciones de tópicos por documento.

β es el parámetro de Dirichlet, previo a la distribución de palabra por tópico.

θ_d es la distribución del tópico por documentos d.

Z_{dn} es el tópico asignado a la palabra n_{th} del documento d.

W_{dn} es la palabra específica en un documento d.

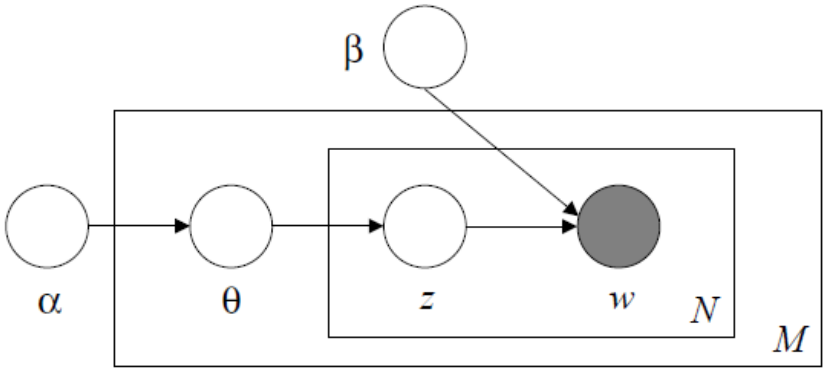


Figura 6. Modelo de representación gráfica de LDA [16].

Las cajas son "platos" mediante los cuales se trata de modelar en una imagen el funcionamiento de LDA y representan:

M el número de documentos en la colección.

N las palabras que se encuentran en cada tópico z y su combinación:

- $\{Z_{dn}, W_{dn}\}$ representan las palabras y tópicos que pertenecen a un documento d.

Capítulo 4: Metodología e Implementación.

En este capítulo se describen la metodología e implementación del sistema para la visualización de tópicos más importantes en medios sociales.

4.1 Desarrollo del sistema de visualización de tópicos más importantes generados en medios sociales (VITOIM).

4.1.1 Arquitectura del sistema VITOIM.

La arquitectura del sistema VITOIM para visualizar tópicos se puede ver en la Figura 7.

La arquitectura se construyó en diferentes etapas:

En los recuadros morados con doble línea punteada donde se puede ver:

A: Una aplicación programada en C# que realiza la etapas de extracción, transformación y carga de noticias de un gestor de base de datos a otro para su mejor utilización.

B: Un programa que realiza una conexión a las bases de datos para obtener como salida, una colección con la información de noticias o tweets.

C: Un programa en PHP que recolecta tweets de periodistas importantes.

D: Programa que utiliza la salida del programa B. a la cual se le aplican técnicas de PLN como es TF y con estas técnicas realizar una MTD en forma de índice invertido, su salida son tres archivos para la ejecución del modelo LDA.

E: Un programa desarrollado en Python que mediante la salida del programa D. utiliza el modelo (LDA) generando un archivo de salida con los contenidos de: Palabras_por_Tópico, Noticias_por_Tópico y Cantidad_de_Noticias_por_Tópico.

F: Un programa que contiene una interfaz gráfica que se programó en java con la intención de poder facilitar su implementación y hacer más amigable la interacción con el usuario y que utiliza la salida del programa E. para así identificar tópicos importantes y visualizarlos, determinando la similitud entre tópicos de distintos periodos.

Los recuadros en rojo con líneas de puntos y guiones, muestran las diversas colecciones:

- EA: noticias recibidas por FeedReader RSS.
- EB: noticias recibidas por TinyTiny RSS.
- EC: Colección de periodistas.

Los cilindros representan los gestores de bases de daros:

- BD: FireBird.
- BD: MySQL.

Los recuadros naranjas punteadas con recuadros internos punteados, muestran los archivos de salida generados por el programa D y E del sistema VITOIM, los cuales serán de utilidad para la ejecución de otra etapa del mismo sistema

- SA: Archivo donde se encuentra el cuerpo de la noticia.
- SB: Archivo donde se encuentran las palabras del diccionario.
- SC: Archivo donde se encuentra la MTD, con forma de índice invertido.
- SD: Archivo donde se encuentra la salida del programa E.

Los cuadros verdes con rombos muestran las tres visualizaciones sencillas que ayudan a reconocer los tópicos más importantes, así como su comportamiento con respecto a otros periodos y su similitud.

- VA: Visualización de tópicos por su importancia debido al número de noticias o tweets.
- VB: Visualización de tópicos semejantes en periodos diferentes.
- VC: Visualización de la semejanza de tópicos por sus palabras repetidas en ambos periodos.
- VD: Visualización por nube de palabras en tópicos, periodos y similares.

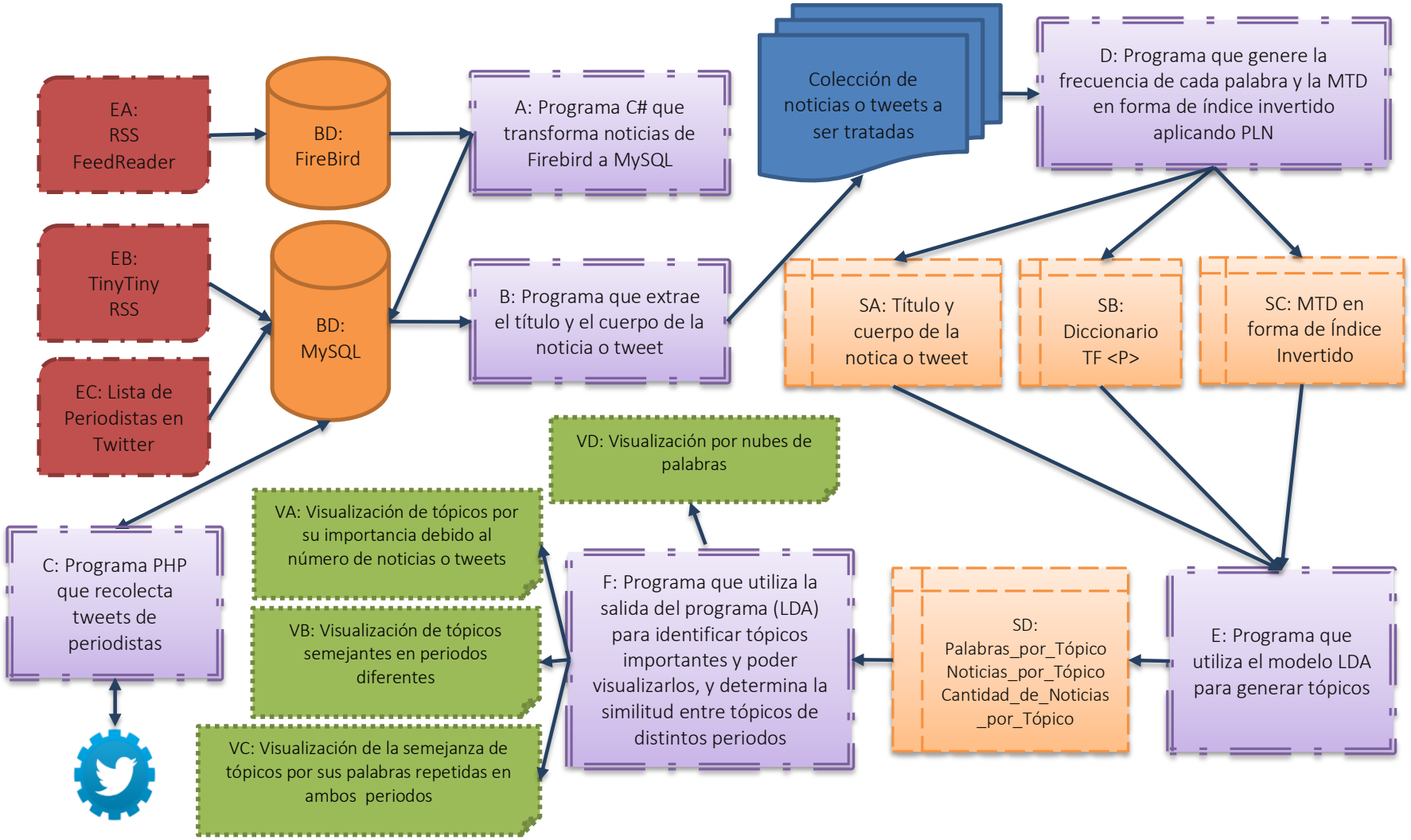


Figura 7. Arquitectura del sistema de visualización de tópicos más importantes generados en medios sociales (VITOIM).

4.2 Recolección de Datos.

Para la recolección de los datos se ocuparon algoritmos existentes y se desarrollaron programas para uso en la web, los cuales facilitaron la obtención de la colección de noticias y tweets, que son dos principales fuentes de información. Con estas fuentes se llevó a cabo el trabajo de tesis, donde la primera parte, consiste en limpiar los datos para su posterior tratamiento en el análisis de tópicos más importantes.

4.2.1 Recolección de Noticias

Para la recolección de noticias se revisó la página web del periódico El Universal[16] donde se seleccionaron cuatro listas de noticias RSS de diferentes secciones (Espectáculos, Computación-Tecnología, Ciencia y México), al final se logró tener una colección de aproximadamente 22,000 noticias mediante dos gestores de RSS que ayudaron a la recolección, los cuales son los siguientes:

1. EA: FeedReader[17] es un programa el cual recibe noticias RSS de forma automática, se recibieron aproximadamente 50 noticias diariamente; su ventaja fue, que no se necesita intervención humana para lograr recibir las noticias.

   Su desventaja es que la base de datos está en un sistema gestor de bases de datos conocido como FIREBIRD®, el cual fue un desafío al principio ya que no resultó ser transparente la forma en que se almacenaban las noticias.

   FeedReader al ser un sistema externo, no mostraba la localización de la base de datos y se pensó que probablemente todos los datos obtenidos hasta ese entonces tendrían que ser procesados manualmente, por lo cual se buscó en ese momento una herramienta que se encargara de almacenar de forma más simple y transparente todas las noticias que se necesitaban, para ello se pensó en utilizar otro gestor de RSS.

[16] El universal RSS, http://www.eluniversal.com.mx/disenio/servicios/EU_rss.htm

[17] FeddReader, http://feedreader.com/

2. EB: TinyTiny RSS[18], es un gestor de noticias RSS el cual funciona casi de la misma forma que FeedReader, su diferencia principal radica en que su almacenamiento es más sencillo utilizando el sistema gestor de base de datos MySQL[19], el cual es de código abierto y más popular, con su documentación, una menor curva de aprendizaje y con un buen número de herramientas de utilidad.

   Su desventaja radica en que la recolección de noticias es de forma manual, ya que se tiene que actualizar ejecutando el ambiente web que ocupa mediante PHP y actualizar los enlaces de cada una de las secciones del periódico, por lo tanto se tiene que tener un orden de actualización de al menos 2 veces al día y así no perder ninguna noticia enviada a través de los RSS de la página El Universal Figura 8.

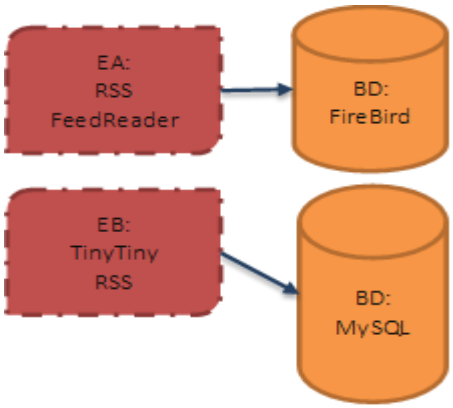


Figura 8. Recolección de noticias RSS.

Dadas las ventajas y desventajas que ofrecían ambos gestores RSS y para evitar la pérdida de noticias ya recolectadas por FeedReader y TinyTiny, se seleccionó FeedReader como principal gestor RSS, ya que después de una actualización de este programa se encontró la localización de la base de datos y mediante un esquema (id, titulo, cuerpo, link y fecha creación) parecido al de TinyTiny, se realizó una transformación de esquemas entre Firebird y MySQL.

[18] TinyTiny RSS, https://tt-rss.org/redmine/projects/tt-rss

[19] MySQL, https://www.mysql.com/

4.2.2 Transformación de las noticias Firebird a MySQL.

Para resolver la problemática que se suscitó al utilizar FeedReader, se desarrolló el programa (A) que llevó a realizar una extracción, transformación y carga de datos [1], para así poder manejar de manera sencilla las noticias recibidas en los meses de noviembre 2013 a septiembre 2014, dicho tratamiento de datos llevó a instalar el sistema Firebird server como un gestor de base de datos, el cual proporciona una librería(DLL) que es utilizada mediante Visual Studio .NET para poder extraer la base de datos de noticias y transformar los campos "ID, TITLE, BODY, DATE", y cargarlos a un esquema parecido a TinyTiny con los mismos atributos originales dentro de base de datos en MySQL.

La extracción, transformación y carga resulto muy favorable, ya que el 99% de las noticias recibidas por el periódico El Universal se encontraban en la codificación UTF-8 y no generaban errores de transformación, mientras que el otro 1% simplemente se evitaron ya que al momento de ser extraídas de la base de datos de Firebird generaban errores de ejecución en el programa (A) al contar con formato HTML.

Mediante esta transformación se lograron recuperan un total de 21,690 noticias almacenadas en MySQL, ver Figura 9.

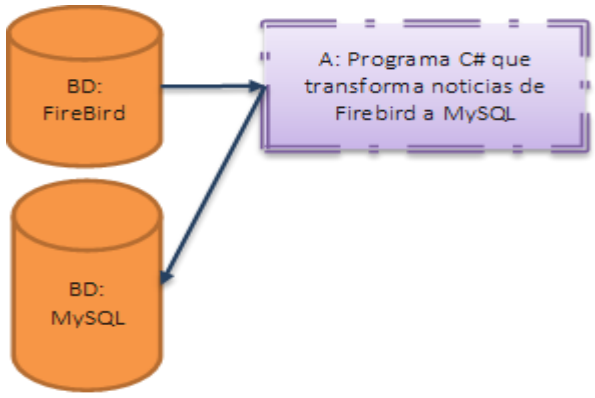


Figura 9. Programa (A) extracción, transformación y carga de noticias Firebird a MySQL.

4.2.3 Colección de reporteros mexicanos en Twitter.

Se pretende dar un peso mayor de interés mediante los tweets generados por periodistas de noticieros que se encuentren en Twitter como usuarios, ya que estos miembros pueden causar una gran cadena de interés hacia sus seguidores o personas que están interesadas en las publicaciones de dicho periodista.

La lista se obtuvo mediante la página web Twitter-México[20] (EC), que tiene una colección de 200 periodistas, que por el número de seguidores y nivel de tweets registrados tienen gran impacto en los mexicanos (ver Figura 10). Mediante esta lista de periodistas se están extrayendo los tweets que ellos generen y los RT que los usuarios están generando por ser *followers*.

#	Twittero	Seguido por	Sigue a	Tweets	Twitea desde	Ultimo Tweet	Categoria
1	@lopezdoriga Joaquín López-Dóriga	4,429,566	987	77,552	01/2011	10/12/2014	periodistas
2	@CarlosLoret Carlos Loret de Mola	4,280,316	745	14,901	08/2009	10/12/2014	periodistas
3	@Adela_Micha Adela Micha	3,208,493	318	19,743	09/2009	10/12/2014	periodistas
4	@LAURAGII Laura G	3,130,687	1,030	65,885	08/2009	10/12/2014	periodistas...
5	@AristeguiOnline Aristegui Noticias	2,870,874	14	76,114	03/2012	10/12/2014	periodistas
6	@aristeguicnn Carmen Aristegui	2,622,406	19	1,073	07/2010	10/12/2014	periodistas
7	@Javier_Alatorre Javier Alatorre	2,447,223	89	5,002	09/2009	06/10/2014	periodistas
8	@Denise_Maerker Denise Maerker	1,303,362	219	1,646	09/2010	09/12/2014	periodistas

Figura 10. Periodistas mexicanos más influyentes en Twitter Dic. 2014

[20] Periodistas en México con twitter y su importancia, http://www.twitter-mexico.com/categoria/periodistas

4.2.4 Programa que recolecta tweets mediante periodistas que se consideran importantes.

Mediante un programa generado(C) en código PHP (ver Figura 11) se creó una base de datos, la cual contiene tweets recuperados por medio de Twitter API 1.1 y se inició la recolección con lista manualmente procesada de 10 palabras claves que pertenecen a las categorías de política y ciencia representativas.

La colección de tweets recuperados se realizó durante lapsos de 10 minutos a lo largo de una semana, con el resultado de 200,000 tweets recuperados a partir de palabras como: Enrique peña, nieto, universal, nasa, narcotráfico, Tlatlaya, Ayotzinapa, justiciaparaayotzinapa, IPN, Instituto politécnico nacional. Se seleccionaron estas palabras debido a que en el análisis surgieron noticias muy importantes en el periodo de septiembre 2014 a octubre de 2014.

Para los periodos siguientes se prefirió usar como palabras claves, las menciones (@) en twitter de los periodistas que se tienen en la colección de bases de datos y mediante estas palabras claves se lograron identificar alrededor de 3, 200,000 tweets en relación al periodismo que se genera en el país.

Figura 11. Parte parcial de la arquitectura del sistema VITOIM la cual muestra la utilización de un programa (C) desarrollado para la captura de tweets sobre periodismo.

Por limitaciones de la memoria del equipo donde se desarrollaron las pruebas solo se pudo probar hasta 20,000 tweets por periodo de tiempo (semana, quincena y mes), por lo cual se procedieron a realizar las pruebas con noticias, que son menores en número.

4.2.5 Creación de un diccionario de todas la noticias en forma Término Frecuencia.

Para la utilización de la implementación seleccionada del modelo LDA se requieren tres tipos diferentes de archivos, donde se utilizan:

- SA: Título y cuerpo de la noticia o tweet (“*.titles”*).
- SB: Diccionario de palabras (“*.tokes*”).
- SC: Matriz término documento expresada en forma de índice invertido (*“.ldac”)*.

Los archivos son creados por un programa (D) (ver Figura 12).

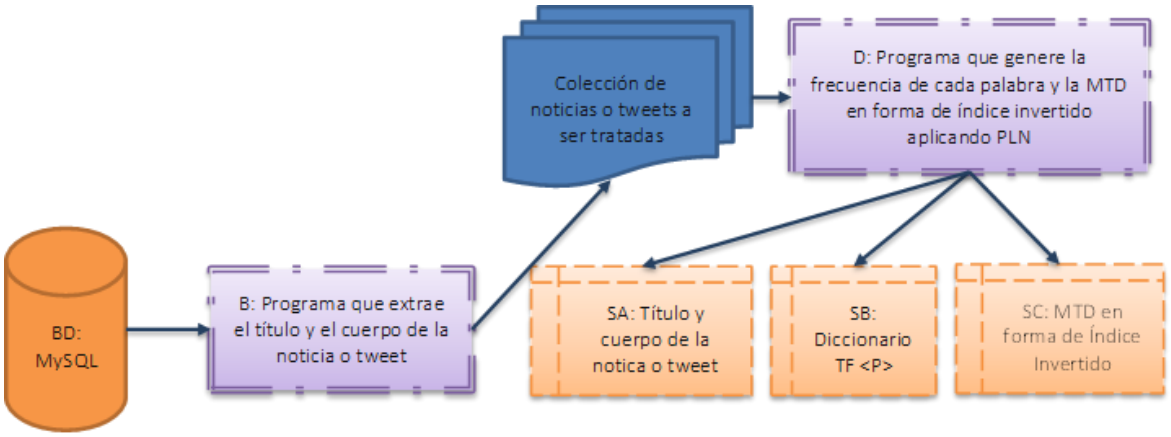


Figura 12. Parte parcial de la arquitectura del sistema VITOIM que muestra el desarrollo del programa (B) y el programa (D), los cuales generan tres archivos de salida para que el sistema VITOIM pueda procesar la salida mediante LDA.

Se desarrolló un programa (B), que realiza la extracción de la información almacenada sobre la colección de noticias que se encuentran en la base de datos.

La colección generada por el programa (B) es ocupada por el programa (D), el cual utiliza métodos de PLN como es TF para realizar el conteo de las palabras en un diccionario de la

forma *(llave, valor).* Una vez obtenido un diccionario son cargadas las llaves en un archivo de salida (SB) con terminación "*.tokens"* (ver Figura 13).

El programa (D) continúa generando la matriz término documento que será reducida en forma de índice invertido y con el cual se podrá generar el archivo de salida (SC) con terminación *".ldac"* (ver Figura 14).

Por último, toda la información que se encuentra dentro del título y el cuerpo de la noticia es almacenada por el programa (D) en el archivo de salida (SA) "*.titles"* (ver Figura 15).

4.2.6 Modelo LDA y su implementación.

La librería en Python que implementa el modelo LDA está basado en la utilización de tres diferentes archivos los cuales son: "archivo.tokens", "archivo.ldac" y "archivo.titles". Estos archivos se crean con un programa (D) que se desarrolló en java, los cuales son utilizados por otro programa (E) creado en Python que los transmite a la librería en Python que implementa el modelo LDA.

El contenido de los tres archivos es el siguiente:

- El primer archivo (.*tokens*) contiene el diccionario completo de palabras en las noticias seleccionadas ordenado alfabéticamente, el cual puede variar, dependiendo de la cantidad de noticias a examinar (ver Figura 13).

  Ejemplo:

almacenada
almacenados
almacenamiento
alta
alteraciones
alternativa
alto
alumnas
alumno

El segundo archivo (.*ldac*) (ver Figura 14) expresa un índice invertido (el cual ya se describió en la sección 3.2.4) de la forma

- {cantidad de palabras en las noticias, número de palabra: frecuencia número de palabra: frecuencia… número de palabra: frecuencia}

Ejemplo:

13 100:1 153:1 190:1 231:3 267:1 372:1 376:1 504:1 548:1 552:2 588:1 648:1 812:1

Donde:

- 13. Es la cantidad total de palabras en la noticia.
- 100, 153, 190, 231, 267, 372,…, etc. Son las palabras que aparecen en una noticia y su posición en el archivo ".tokens".
- :1, :1, :1,….,:2, :1, :1, :1. Es un separador (:), seguido de la frecuencia de cada palabra, la suma de todas las frecuencias da como resultado la cantidad de palabras que en este ejemplo son 13.

- El tercer archivo (.*titles*) cuenta con un formato de tipo {id texto}, donde el id es el número de noticia y el texto es el generado por la información que contiene dicha noticia (ver Figura 15).

  Ejemplo.

  12 Los bebedores suelen vivir más: estudio Un estudio reveló que aquellos que beben de forma moderada viven más que los abstemios CIENCIAS

  13 Cárdenas, Premio Nacional de Periodismo y Divulgación Científica Obtuvo el tercer lugar en la categoría Periodistas, modalidad Noticia, por el texto Las claves de la longevidad en México; CIENCIAS

  14 Usarán microalgas para reducir CO2 del cemento Las microalgas son tal vez 100 o 200 veces más eficientes que los árboles en la captura de CO2 CIENCIAS

  15 Los riesgos de la ropa usada El uso de las prendas podrían provocar dermatitis, tiña o sarna CIENCIAS

```
2649    aitchison
2650    aj
2651    ajedrecista
2652    ajedrez
2653    ajena
2654    ajenas
2655    ajeno
2656    ajenos
2657    ajetreadas
2658    ajl
2659    ajo
2660    ajolote
2661    ajolotes
2662    ajusco
2663    ajusta
2664    ajustado
2665    ajustados
2666    ajustan
2667    ajustando
2668    ajustar
2669    ajustaron
2670    ajustarse
2671    ajustará
2672    ajuste
2673    ajusten
2674    ajustes
```

Figura 13. Ejemplo del archivo con formato "archivo.tokens" donde se muestra el diccionario de palabras.

```
13 442:1 638:1 825:1 1035:1 1211:1 1340:1 1783:1 2023:1 2170:1 2197:1 2351:1 2460:1 2913:1
18 155:1 921:1 932:1 1051:1 1082:1 1150:1 1188:1 1308:1 1445:1 1580:1 1826:1 1871:1 2231:1 2462:1 2495:1 2510:1 2955:1 3154:1
19 14:1 19:1 38:1 107:1 598:1 775:1 1000:1 1174:1 1391:1 1742:1 1913:1 2127:1 2335:1 2560:1 2610:1 2915:1 3056:1 3062:1 3293:1
21 190:1 230:1 487:1 838:1 1109:1 1550:1 1931:1 1940:1 1942:1 1949:1 2029:1 2065:1 2142:1 2201:1 2573:1 2642:1 2835:1 2943:1 3169
16 199:1 242:1 962:1 1251:1 1344:1 1355:1 1742:1 1745:1 1913:1 2021:1 2079:1 2335:1 2563:1 3013:1 3056:1 3227:1
16 33:1 154:1 610:1 718:1 799:1 964:2 1184:2 1368:1 1445:1 1679:2 1745:1 1788:1 1793:1 1822:1 2134:1 2632:2
21 384:1 385:1 675:1 703:1 871:1 987:1 1144:1 1347:1 1355:1 1581:1 1671:1 2105:1 2127:1 2185:1 2271:1 2842:1 2882:2 2890:1 2947:1
29 52:1 148:1 329:1 490:1 607:1 641:1 810:1 848:1 968:1 1016:1 1237:1 1242:2 1274:1 1364:1 1699:1 1709:1 1862:1 1948:1 2153:1 223
18 222:1 457:1 849:1 922:1 1188:1 1580:1 1644:1 1798:1 1804:1 1874:1 1969:1 1980:1 2385:2 2510:2 3028:1 3221:1 3286:1 3327:1
20 442:1 445:1 530:1 532:1 1073:1 1274:1 1435:1 1445:1 1676:1 1747:1 1803:1 1979:1 2437:1 2441:1 2471:1 2488:1 3109:1 3123:1 3235
29 7:1 53:1 128:1 423:1 461:1 585:1 618:1 680:1 1089:1 1309:1 1352:1 1353:1 1358:1 1392:1 1428:1 1498:1 1567:1 1808:1 1921:1 2091
21 173:1 445:1 616:1 647:1 1229:1 1378:1 1593:1 1615:1 1913:1 2302:1 2335:1 2364:1 2560:1 2566:1 2610:1 2803:1 2915:1 3056:1 3062
21 299:1 410:1 547:2 559:1 619:1 623:1 1411:1 1503:1 1655:2 1833:1 2074:2 2197:1 2379:1 2484:2 2525:1 2744:1 2756:1 2770:2 2935:1
17 31:1 229:1 660:1 1033:1 1445:1 1506:1 1636:1 2015:1 2113:1 2130:1 2185:1 2457:1 2488:1 2489:2 2578:1 2769:1 3297:1
16 514:1 880:1 898:1 1010:1 1330:1 1421:1 1446:1 1484:1 1565:1 1841:1 1901:1 2457:1 2815:1 2817:1 3012:1 3059:1
15 265:1 323:1 1029:1 1042:1 1296:1 1742:1 1913:1 2157:1 2335:1 2560:1 2610:1 2915:1 3056:1 3060:1 3062:1
17 292:1 311:1 545:1 889:1 1216:1 1530:1 1538:1 1955:1 1964:1 2003:1 2033:1 2160:2 2171:1 2265:1 2385:1 2754:1 2900:1
14 354:1 786:1 837:1 1114:1 1585:1 2162:1 2354:1 2386:1 2425:1 2459:1 2591:1 2742:1 2743:1 2986:1
15 8:1 59:1 342:1 578:1 1072:1 1076:1 1445:1 2113:1 2187:1 2302:1 2488:1 2664:1 2949:1 2957:1 3074:1
15 445:1 514:1 622:1 655:1 1017:1 1184:1 2185:1 2456:1 2488:1 2603:1 2669:1 2746:1 3027:1 3317:1 3318:1
14 1227:1 1293:1 1423:1 1882:1 1918:2 1965:1 2017:1 2092:1 2173:1 2236:1 2336:1 2455:1 2810:1 3211:1
21 398:1 551:1 696:1 773:1 1133:1 1301:1 1723:1 1884:1 2033:1 2064:1 2123:1 2197:1 2206:1 2496:1 2665:1 2737:1 2760:1 2925:1 2949
17 69:1 466:1 602:1 747:1 1046:1 1521:2 1525:2 1661:1 1974:1 1997:1 2037:1 2070:1 2247:1 2402:1 2571:1 2640:1 3121:2
15 276:1 786:1 996:1 1115:1 1356:1 1545:1 1938:1 1997:1 2002:1 2114:1 2425:1 2479:1 2577:1 2854:1 2986:1
19 184:1 189:1 237:1 498:1 716:1 726:1 769:1 939:1 1250:1 2048:1 2197:1 2201:1 2447:2 2607:1 2831:1 2936:1 3139:1 3281:1 3337:1
22 69:1 377:1 413:1 464:1 466:1 557:1 1046:1 1184:1 1521:2 1525:2 1529:1 1661:1 1670:1 1974:1 2037:1 2272:1 2341:1 2402:1 2547:1
13 265:1 667:1 942:1 1067:1 1178:1 1427:1 1448:1 1847:1 1955:1 2130:1 2406:1 2467:1 2904:1
18 12:2 66:1 192:1 626:1 1058:1 1088:1 1150:1 1185:1 1195:1 1432:1 1433:1 1445:1 2008:1 2570:1 2601:1 2664:1 2935:1 2942:1
19 32:1 35:1 38:1 67:2 74:1 116:1 285:1 598:2 1000:1 1354:1 1742:2 2067:1 2118:1 2174:1 2533:1 2568:1 2750:1 2915:1 2935:1
14 326:1 400:2 799:1 893:1 894:1 1052:1 1565:1 1680:2 1793:1 1996:2 2126:1 2383:1 2386:2 2807:1
17 177:1 222:1 457:1 887:1 1325:1 1627:1 2197:1 2385:1 2429:1 2433:1 2510:1 2654:2 2711:1 2903:1 3028:1 3029:1 3176:1
20 311:1 476:2 519:2 872:1 993:1 1122:1 1799:1 2003:1 2033:1 2117:1 2160:1 2205:1 2265:1 2396:1 2433:1 2510:1 2629:1 2968:1 3025:
22 25:1 36:1 51:1 62:1 83:1 87:1 116:1 119:1 393:1 460:1 907:1 1085:1 1187:1 1211:1 1989:1 2118:3 2127:2 2209:1 2298:1 2449:1 255
20 30:1 263:1 371:1 640:1 754:1 993:1 1443:1 1516:1 1531:1 1576:2 1762:1 1764:1 2048:1 2201:1 2918:1 2934:1 3020:1 3151:1 3169:1
16 543:1 818:1 951:1 1384:1 1619:1 1938:2 1949:2 2183:1 2237:1 2459:1 2569:1 2743:1 2902:2 2943:2 2993:1 3168:1
```

Figura 14. Ejemplo del archivo .ldac que contiene del índice invertido para la implementación del modelo LDA

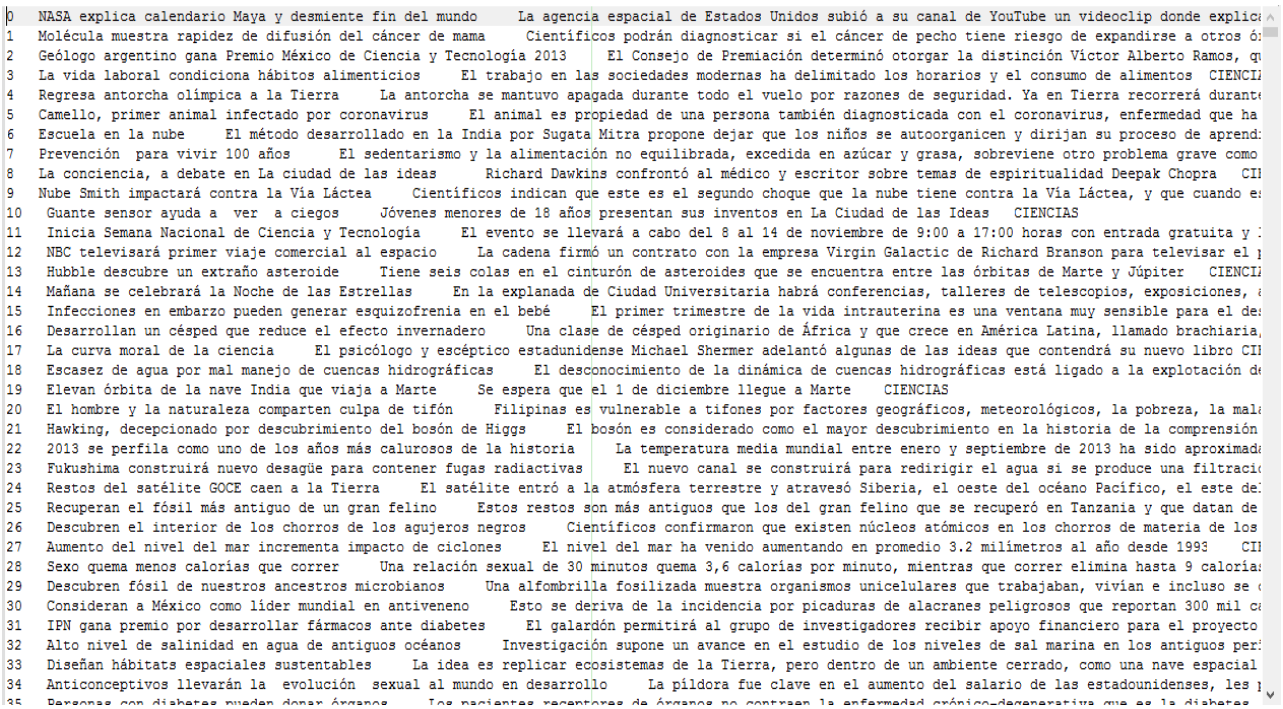


Figura 15. Ejemplo del archivo con formato "archivo.titles" donde se muestra el id de la noticia y su contenido.

Para la ejecución del programa (E), que utiliza la librería en Python que implementa el modelo LDA, son requeridas dos variables importantes:

1. Número k. El número máximo de tópicos, en los que el usuario desea que se clasifiquen las noticias que se generaron en un tiempo determinado de fechas.
2. Número w. El número de palabras que representarán cada tópico (Palabras_por_Tópico) desde uno hasta el máximo de palabras que contiene el diccionario.

El programa (E) genera una matriz completa (documentos X tokens) mediante el índice invertido que se encuentra en el archivo ".ldac" y el número de palabras en el archivo ".tokens", con la ejecución de la librería se asignan probabilidades de ocurrencia por medio de:

- El número de palabras por documento.
- El número de documentos.
- Y la frecuencia de las palabras en todos y cada uno de los documentos.

La salida que se obtiene de programa anterior (E), se muestra en la Figura 16.

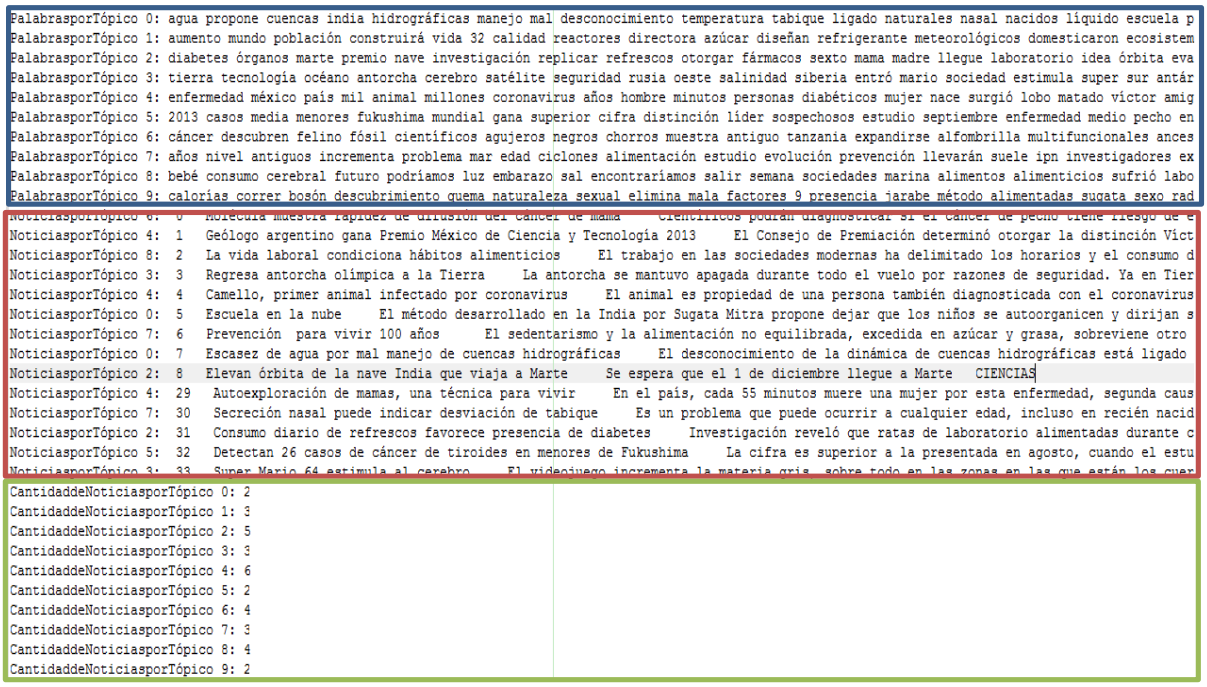


Figura 16. Ejemplo del archivo de salida del programa (E) que implementa el modelo LDA" generando Palabras_por_Tópico, Noticias_por_Tópico, y Cantidad_de_Noticias_por_Tópico, al lado izquierdo de los (:) siempre se localiza un identificador de un tópico y del lado derecho de los (:) se encuentra la información asignada a dicho tópico.

El resultado de la ejecución del programa (E) de la Figura 16, tiene la siguiente estructura:

1. Palabras_por_Tópico. Indica el conjunto de palabras obtenidas por el modelo LDA, que representan a cada tópico. La probabilidad (asignada por LDA) de ocurrencia de las palabras en un tópico disminuye de izquierda a derecha (aquí el tópico no tiene nombre, solo el conjunto de palabras que lo representan).

   Ejemplo donde el tópico 2 tiene las siguientes palabras:

   PalabrasporTópico 2: diabetes órganos marte premio nave investigación replicar refrescos otorgar fármacos sexto mama madre llegue laboratorio idea órbita evaluación tanques espera espaciales espacial elevan donar diciembre diagnosticar desarrollaron desarrollar crónicodegenerativa base apoyo antioxidantes alto síntesis

2. Noticias_por_Tópico. Son los documentos o noticias con una asignación a un tópico obtenido por el modelo LDA.

Ejemplo donde el tópico 2 se le asigna la noticia número 8:

NoticiasporTópico 2: 8 Elevan órbita de la nave India que viaja a Marte Se espera que el 1 de diciembre llegue a Marte CIENCIAS

3. Cantidad_de_Noticias_por_Tópico. Es el número de documentos o noticias asignados a un tópico.

   Ejemplo en el que el tópico 2 tiene 5 noticias asignadas:

   CantidaddeNoticiasporTópico 2: 5

4.2.7 Programa para determina tópicos importantes en las noticias durante un periodo de tiempo.

El sistema VITOIM mediante la ventana en la Figura 17 ejecuta el programa (D) y después de la generación de los tres archivos que utiliza el modelo LDA manda a ejecutar al programa (E) en Python que implementa modelo LDA y con los siguientes parámetros de entrada solicitados:

- Los k tópicos que se desean generar para cada intervalo de tiempo.
- Un número N de palabras representativas de cada tópico dependiendo del tipo de búsqueda: semana (200 palabras), quincena (500), mensual (2000).

En la Figura 17, se puede observar una ventana, en la cual el usuario podrá revisar el historial de tópicos en noticias seleccionando y alimentando:

1. El tipo de noticia: Espectáculos, Computación-Tecnología, Ciencia y México.
2. El intervalo de búsqueda semanal, quincenal o mensual.
3. Periodo, equivale a cuántos intervalos desea el usuario
4. El número de tópicos solicitados por el usuario, como mínimo 1 y como máximo 15.
5. La fecha de inicio del periodo 1 (el sistema lo despliega como cero).

- Los periodos que se seleccionan son consecutivos, termina un periodo e inicia inmediatamente el siguiente periodo.

- El número de tópicos deseable es 10; ya que si se seleccionan los límites el modelo LDA genera:
  - ✓ En caso de un solo tópico, todas las palabras del diccionario formarían parte del tópico.
  - ✓ Si seleccionamos 15 tópicos, serían muy puntuales dependiendo de las palabras que contengan las noticias y podrían quedar incluso como tópicos vacíos, si son relativamente un número pequeño de palabras en las noticias.

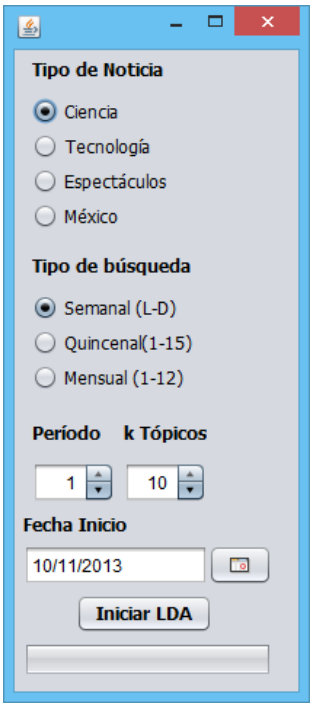


Figura 17. Ventana inicial del sistema VITOIM, la cual ejecuta primero el programa (D) que genera los tres archivos de salida que utilizara el programa (E) mediante los cuales implementa el modelo LDA.

El botón Iniciar LDA abrirá una nueva ventana donde se podrán observar los P periodos y los k tópicos solicitados por el usuario, mediante estos poder encontrar similitud entre tópicos de distintos periodos que pueden seguir ocurriendo (ver Figura 18).

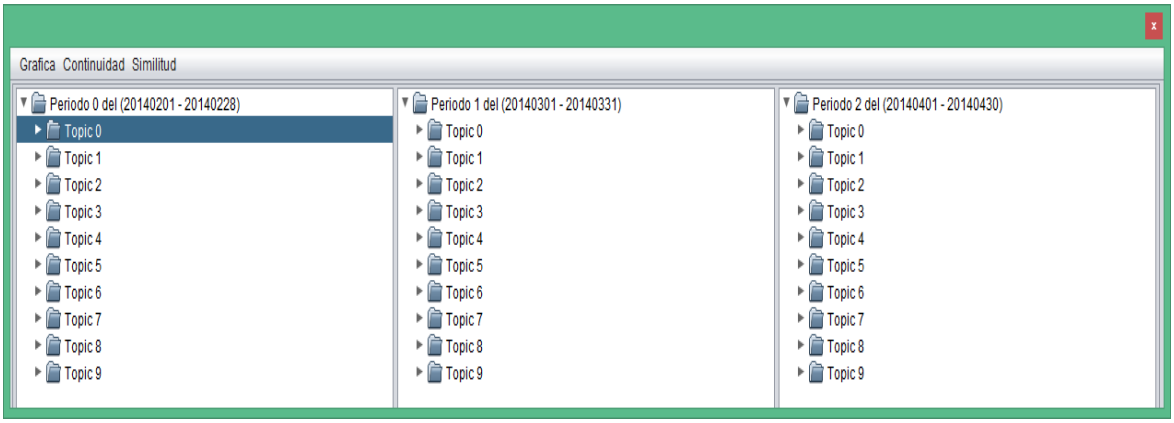


Figura 18. Visualización de 10 tópicos no etiquetados en tres periodos de tiempo (semana, quincena, meses).

La visualización de los tópicos por palabras en noticias, tiene como objetivo mostrar que mediante la estructura de un seudo árbol de archivos, se puede representar los tópicos que se mencionan en cada uno de los periodos de tiempo solicitado por el usuario.

Las palabras que nos entrega el modelo LDA y que pertenecen a un tópico, están representadas como nodos hoja dentro del árbol de seudo archivos y se encuentran ordenadas de mayor a menor probabilidad de existencia en un tópico, por lo cual la primeras palabras que se aparecen en la lista de nodos hoja son las más representativas de cada uno de los tópicos generados por el modelo LDA (ver Figura 19). Las palabras que aparecen en los tópicos sirven para poder identificar el tema del cual se está hablando en ellos y si se desea poder etiquetarlo de manera manual.

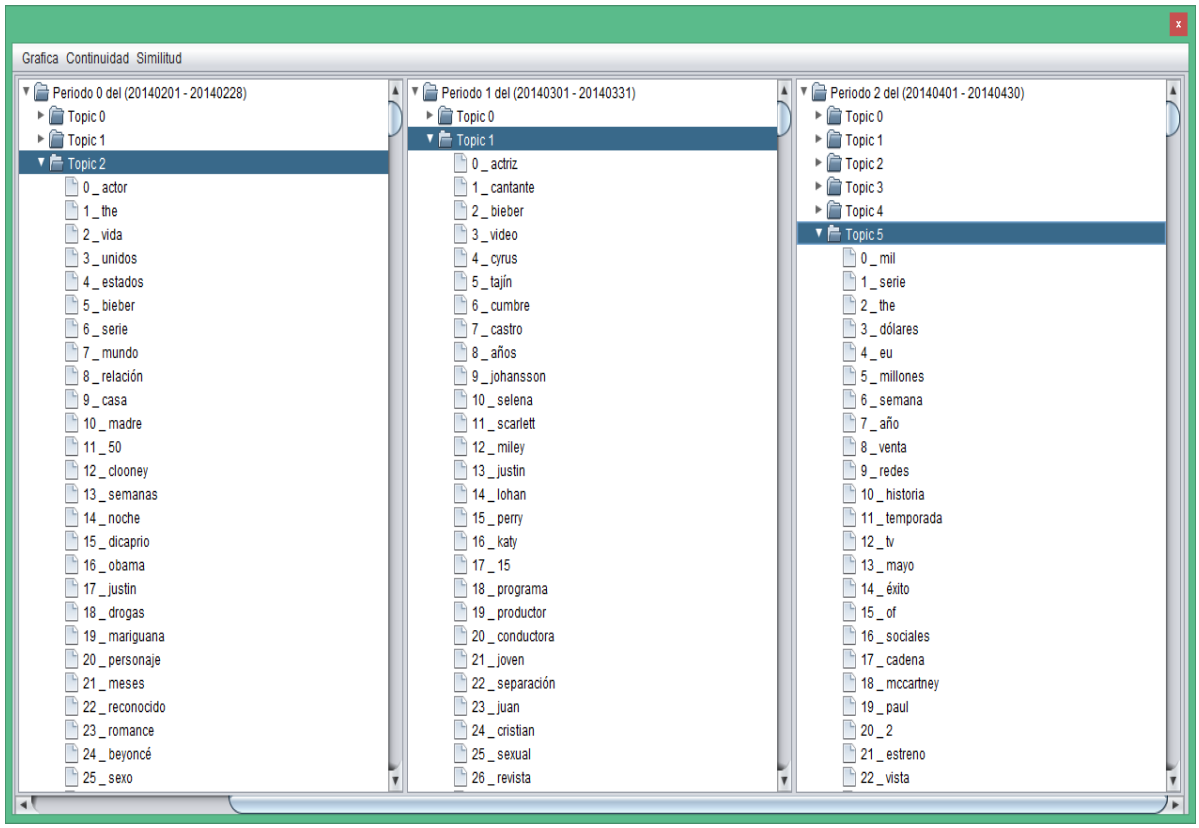


Figura 19. Palabras que representan al tópico 2, tópico 1 y tópico 5 en los periodos continuos periodo 0, periodo 1 y periodo 2.

En la Figura 20, Figura 21 y Figura 22 se explica cómo se generan las visualizaciones que serán mostradas en la sección 4.4.

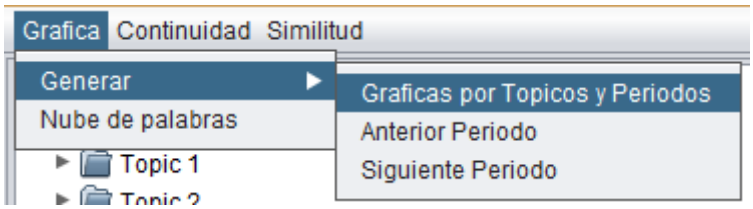


Figura 20. El botón Gráfica generara tanto las gráficas sobre el número de noticias en los tópicos por periodo, como la visualización por nube de palabras

Figura 21. Botón que genera la gráfica de continuidad entre tópicos de diferentes periodos.

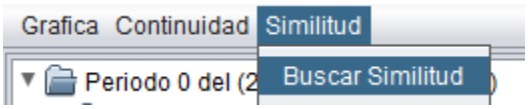


Figura 22. Botón que muestra las palabras que son similares en los dos tópicos de diferentes periodos.

La manera de entender cómo se realiza la comparación entre dos tópicos de diferentes periodos, es mediante la intersección de las palabras en dichos tópicos, disminuyendo su diferencia debido a que ocupan la misma posición en ambos tópicos o se encuentran cercanas dos palabras similares. Mediante la medición de la similitud se puede visualizar el comportamiento de un tópico y su variación en el tiempo.

Este comparativo se describe más a fondo en la siguiente sección.

4.3 Modelo Formal de comparación entre tópicos

4.3.1 Número de comparaciones entre los Tópicos por periodos.

El número de comparaciones que realiza el sistema VITOIM, es de manera exhaustiva sobre las palabras de todos los tópicos en todos los periodos siguientes y es a lo más:

$$C = (k * N) * \sum_{i=1}^{P-1} [(P - i) * (k * N)]$$

Donde:

- ***C*** es el número de comparaciones que realiza el sistema.
- ***P*** es el número de periodos de tiempo elegidos por el usuario.
- ***N*** es el número de palabras que representan a un tópico (ver sección 4.2.1).
- ***k*** es el número de tópicos que existen por periodo *P*.
- ***i*** esta entre 1 a P-1

Aquí se contempla el caso extremo, en caso de que todos los tópicos tengan el máximo número de palabras, cualquier otro caso, serán el menor el número de comparaciones.

Ejemplo:

Si k = 10 tópicos;

N = 100 palabras por tópico;

P = 5 periodos;

C es el total de comparaciones:

C = (10*100)*(4*10*100 + 3*10*100 + 2*10*100 + 1*10*100)

= (10*100)*(10*10*100)

= (103)*(104)

= (107)

= 10, 000,000 de comparaciones

4.3.2 Modelo de similitud y distancia entre tópicos.

En nuestra siguiente notación, i & j se desplaza entre el número de periodos P, y l & h entre 1 y el número k de tópicos por período P.

La distancia que existe entre dos tópicos que se encuentran en diferentes periodos P^i y P^j con $i \neq j$, donde i y j van desde 1 a P se puede expresar de la siguiente manera:

$$T_l^i = \{w_l^i 0, w_l^i 1, w_l^i 2, \ldots, w_l^i n_i\}$$

$$T_h^j = \{w_h^j 0, w_h^j 1, w_h^j 2, \ldots, w_h^j n_j\}$$

Donde:

- i y j son periodos distintos desde 0 a P.
- l y h son los tópicos que se encuentren en los periodos i y j respectivamente.

T_l^i y T_h^j pertenecen al conjunto de tópicos en los periodos P^i y P^j respectivamente.

w_l^i Es el conjunto de palabras que se encuentran en el tópico T_l del periodo P^i.

w_h^j Es el conjunto de palabras que se encuentran en el tópico T_h del periodo P^j.

La distancia se define como sigue:

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=1}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

Observación. Si $w \in T_l^i$ y $w \notin T_h^j$ su posición en h, *pos (w)* es 0.

La función *pos (w)* determina el valor de la posición de w dentro de los tópicos l y h.

Si la $Dist(T_l^i, T_h^j) = 0$, los tópicos son los mismos.

Si la $Dist(T_l^i, T_h^j) = 1$, los tópicos son completamente diferentes, no tienen nada en común.

Si la $0 <= Dist(T_l^i, T_h^j) < 1$ los tópicos tienen algunas palabras en común.

Observación. Cuando se compara un tópico con varios tópicos de otros periodos, se selecciona el tópico con menor distancia:

$Dist(T_l^i, T_h^j)$ y $Dist(T_l^i, T_m^j)$

$$Dist(T_l^i, T_h^j) = \min\left[\left(\frac{2}{N(N+1)}\right) * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|\right]$$

Ejemplos sobre distancias entre tópicos con N=13:

Posición Tópico l		Posición Tópico h	
1	mexicano	1	vive
2	musical	2	latino
3	años	3	festival
4	méxico	4	banda
5	vive	5	día
6	cine	6	escenario
7	teatro	7	musical
8	obra	8	edición
9	josé	9	13
10	latino	10	público
11	festival	11	rock
12	público	12	música
13	mexicana	13	grupo

2/[(N)(N+1)] =
0.01098901

Comparación entre l y h		
Posición en l	Posición en **h**	Pos(w en l) - Pos(w en **h**)
1	0	1
2	7	5
3	0	3
4	0	4
5	1	4
6	0	6
7	0	7
8	0	8
9	0	9
10	2	8
11	3	8
12	10	2
13	0	13
	Sumatoria	78

Distancia entre l y **h**
78*0.01098901 =
0.85714285

4.4 Gráficas de noticias.

4.4.1 Gráficas generadas del número de noticias en un tópico de cada periodo.

Al momento de utilizar el modelo LDA no solo se determinan las palabras representativas de todos los tópicos generados en un determinado periodo i, también se genera un listado del número de noticias en las que aparece cada uno de los tópicos l del periodo i (ver Figura 23). El sistema VITOIM se encarga de almacenar este resultado dentro de un esquema parecido al árbol que se muestra en la Figura 19 pero en vez de guardar *N* palabras, el esquema guarda el número de noticias asignadas a los tópicos.

Las gráficas que serán generadas son diferentes en cada instante del tiempo y por cada búsqueda distinta, siempre y cuando se varié el tipo de noticia, la fecha de inicio y el número de periodos.

Para ejecutar un ejemplo sencillo pondremos como entrada:

- ✓ Tipo de noticia: Espectáculos.
- ✓ Tipo de búsqueda: Mensual.
- ✓ #Periodos: 5 meses.
- ✓ k Tópicos: 10
- ✓ Fecha de inicio: 01/02/2014

Figura 23. Resultado de LDA sobre el número de noticias en las que aparecen los tópicos del periodo 0.

El resultado del programa (E) que ejecuta la librería en Python que implementa el modelo LDA es graficado mediante el botón graficas por tópicos y periodos (ver Figura 20) del programa (F) en el sistema VITOIM, así logrando una interpretación más entendible y con esta poder ayudar a determinar, ¿Qué tópico cuenta con más número de noticias distribuidas en un determinado periodo de tiempo? Figura 24, Figura 25, Figura 26, Figura 27, Figura 28. Recordando que los tópicos T_{ji} y T_{ki} son distintos el uno del otro y las gráficas no interpretan aun continuidad con los tópicos.

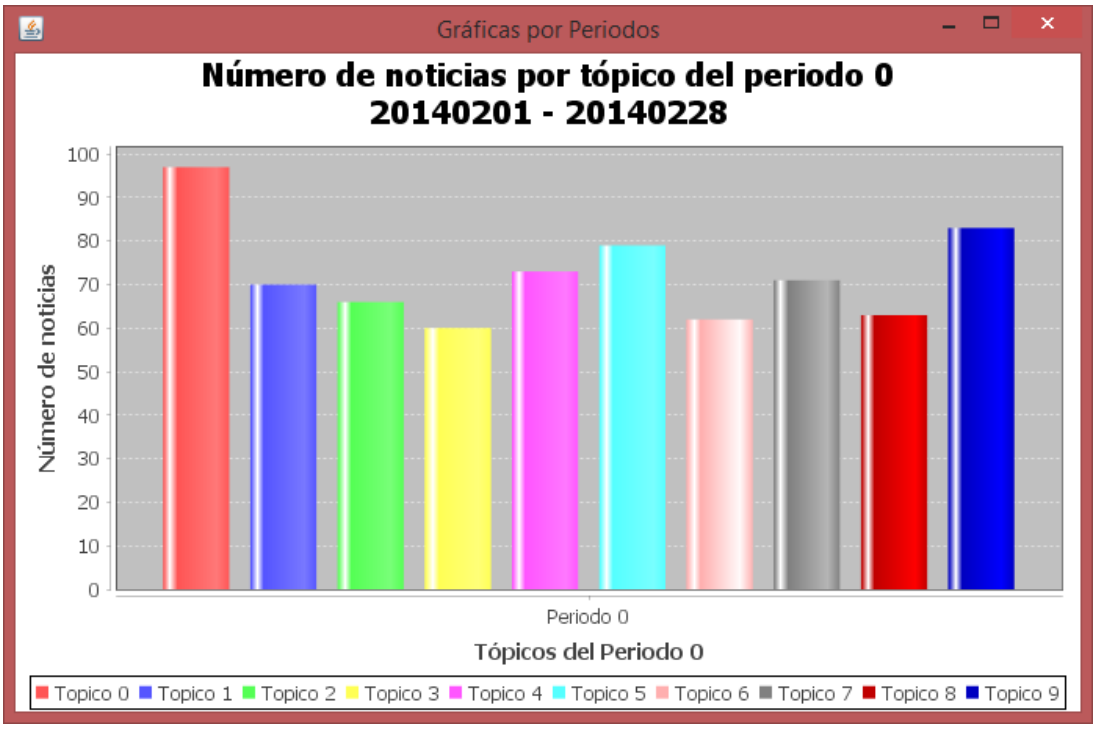


Figura 24. Gráfica que determina la importancia de los tópicos generados para el periodo 0 mediante el número de noticias en las que se encuentran en cada tópico.

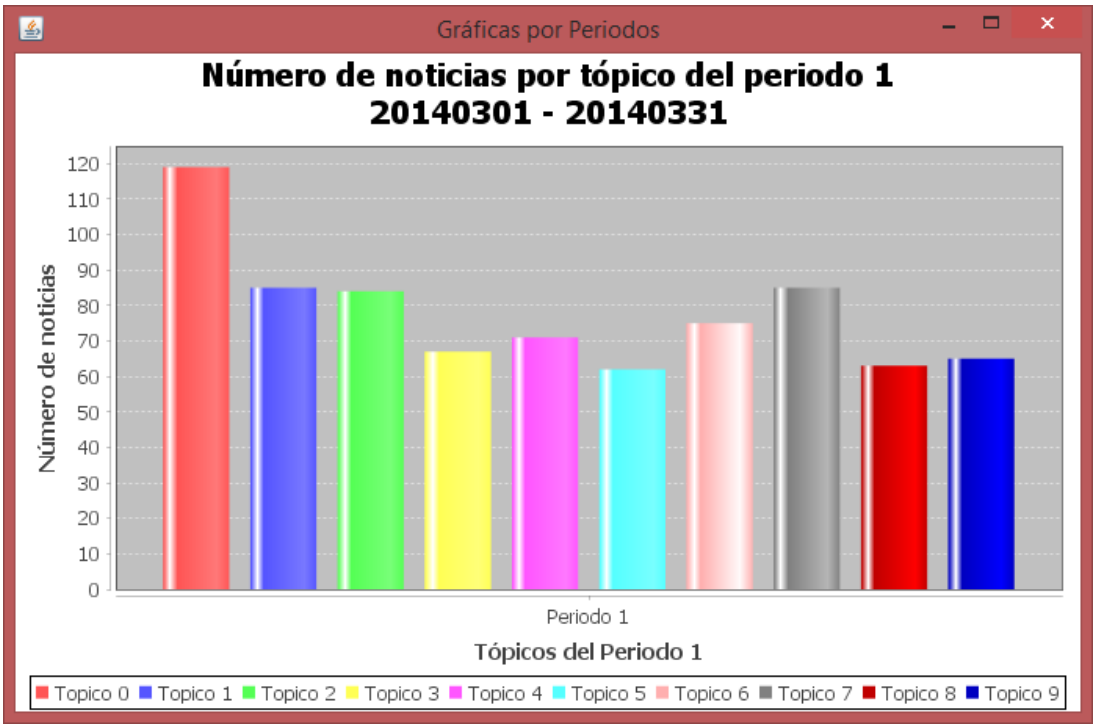


Figura 25. Gráfica que determina la importancia de los tópicos generados para el periodo 1 mediante el número de noticias en las que se encuentran en cada tópico.

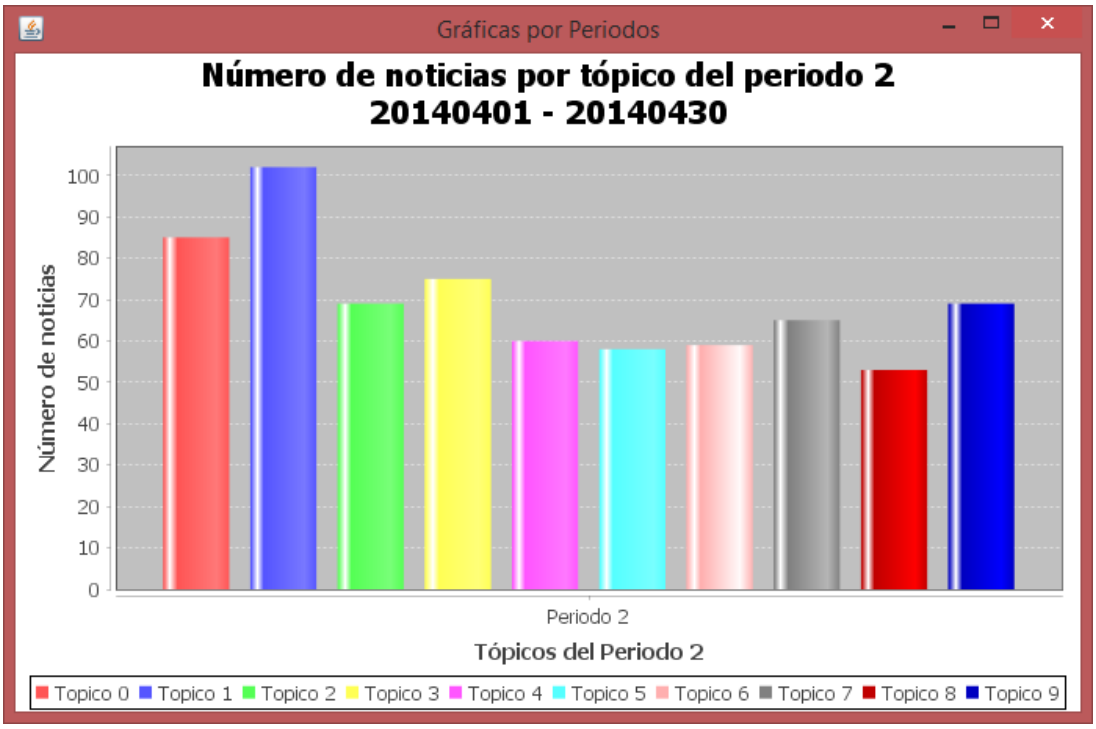


Figura 26. Gráfica que determina la importancia de los tópicos generados para el periodo 2 mediante el número de noticias en las que se encuentran en cada tópico.

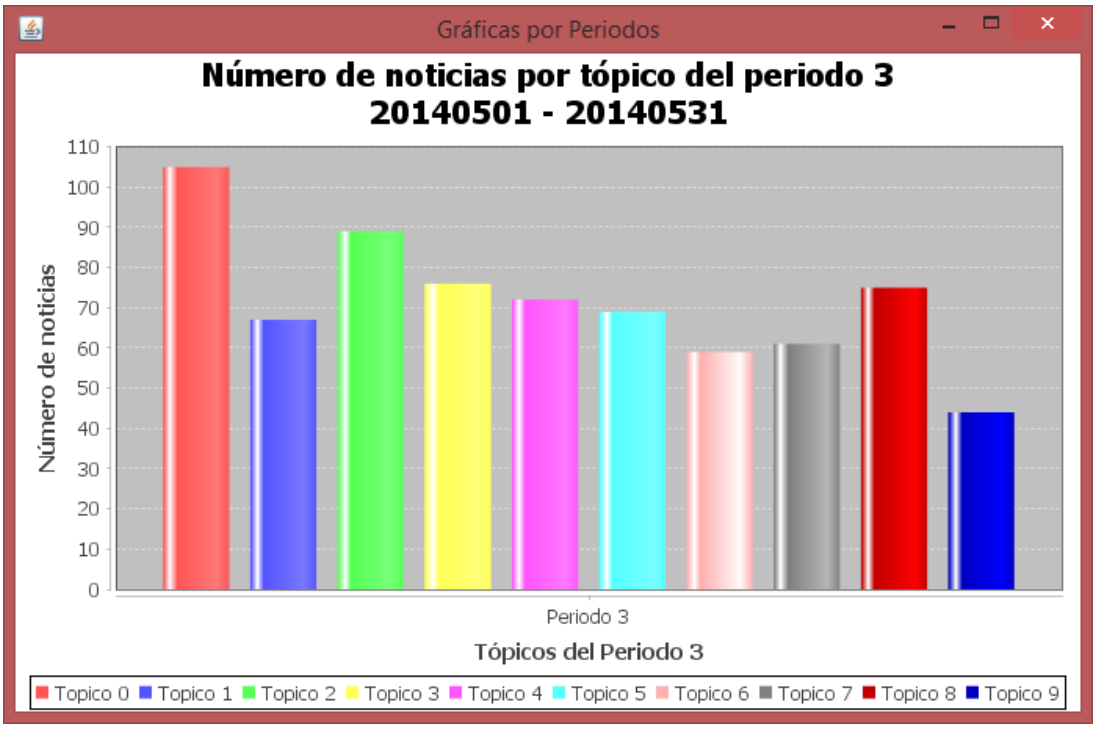


Figura 27. Gráfica que determina la importancia de los tópicos generados para el periodo 3 mediante el número de noticias en las que se encuentran en cada tópico.

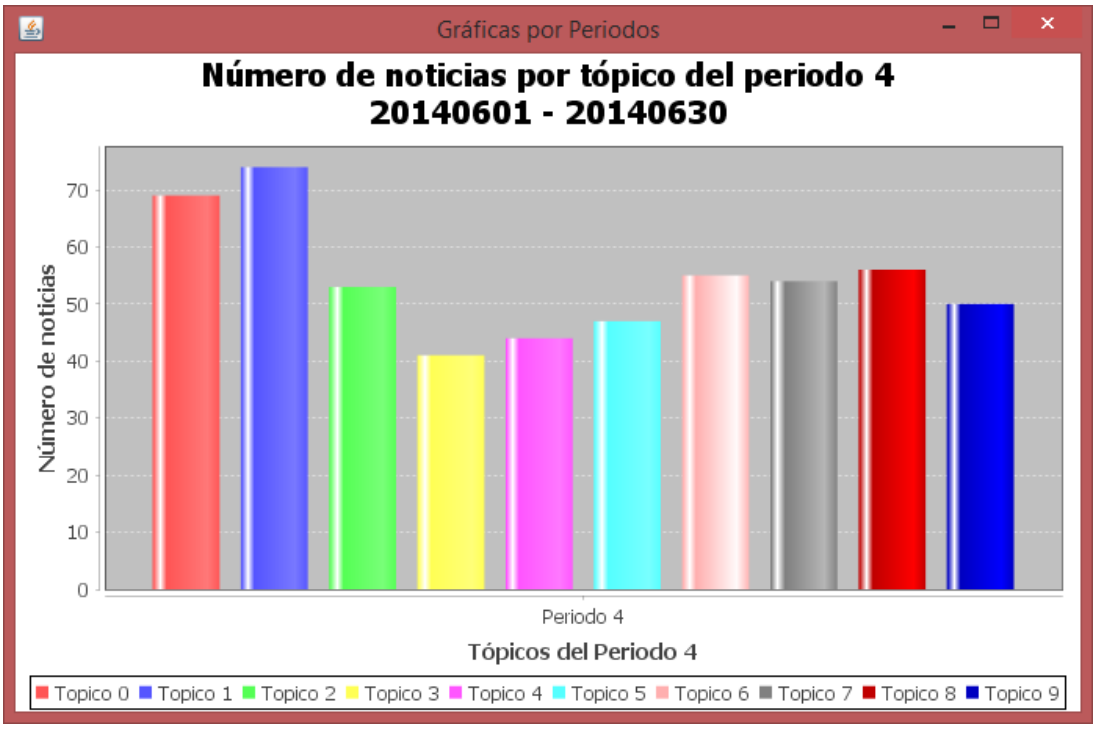


Figura 28. Gráfica que determina la importancia de los tópicos generados para el periodo 4 mediante el número de noticias en las que se encuentran en cada tópico.

Con estas gráficas se pueden observar el número de noticias en las que se encuentran los tópicos de un periodo *i*, usando el simple conteo de noticias se logra realizar esta forma de visualización, interpretando que el tópico con mayor número de noticias genera una mayor importancia.

Para entender de qué se trata el tópico de algún periodo *i* es indispensable observar las palabras que lo representan (ver Figura 19).

Las gráficas se generan cuando son requeridas, una por una para poder lograr su mejor comprensión y observación con respecto al árbol de periodos, tópicos y palabras determinando cual es el significado de cada uno de las clases (tópicos) que genera el modelo LDA, entendiendo el por qué algunos tópicos contienen una cantidad razonablemente más alta de noticias que otro y es debido al comportamiento de la información contenida en las noticias.

4.4.1 Gráficas generadas de similitud entre tópicos.

Una vez desarrollado el modelo de distancia entre tópicos, se pueden buscar similitudes entre los periodos que son generados a partir de las noticias subsecuentes a un periodo *i*, este modelo revisa el número de palabras existentes entre los tópicos de periodos consecutivos asignando una distancia entre [0, 1], un caso es cuando algunas palabras en el tópico T_{il} se encuentran en T_{jh}, el resultado estará en ese rango de valores.

Determinando una distancia mínima 0 y máxima de 0.85 para obtener similitud, nos dio como resultado que algunos tópicos T_{il} tienen un nivel de parecido con algunos tópicos T_{jh}.
Mientras más información se tenga en las noticias se puede determinar más fácilmente si existe una similitud en diferentes tópicos de periodos, puesto que para conocer si existen más palabras semejantes, se necesita cumplir alguno de los siguientes ejemplos:

1. Más noticias por periodos.
2. Noticias con un mayor número de palabras en su título y cuerpo (Espectáculos y México).
3. Seleccionar periodos de fecha largos (quincenas o meses).

Con base en el ejemplo anterior, el éxito de hallar un tópico similar en un periodo, se realizó con los siguientes parámetros:

- Tipo de noticia: Espectáculos.
- Tipo de búsqueda: Mensual.
- Periodos: 5 meses.
- k Tópicos: 10
- Fecha de inicio: 01/02/2014

Los resultados obtenidos mostraron que se generan más similitudes de tópicos con los parámetros solicitados debido a que se cuenta con más información.

En la Figura 29 se muestra la similitud de un tópico inicial (tópico 0 del periodo 0) con respecto a otros tópicos de diferentes periodos {tópico 9 del periodo 1, tópico 1 del periodo 2, tópico 6 del periodo 3} donde su importancia va disminuyendo con respecto al número de noticias publicadas a través de los diferentes periodos; lo que indica que el tópico puede desaparecer con el paso del tiempo.

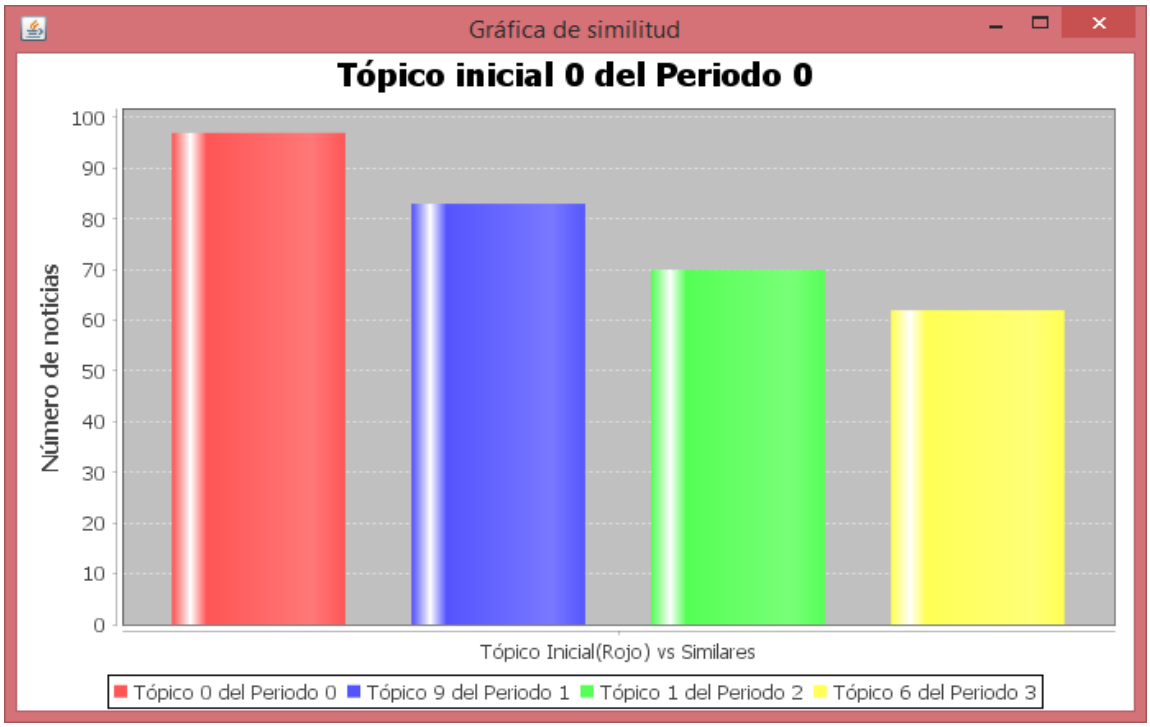


Figura 29. Número de noticias del tópico 9 del periodo 1, tópico 1 del periodo 2, tópico 6 del periodo 3 similares al tópico inicial 0 del periodo 0, con valor de similitud <= 0.85

Los valores de similitud del tópico 9 del periodo 1, tópico 1 del periodo 2, tópico 6 del periodo 3 con respecto al tópico inicial 0 del periodo 0, se pueden ver en la Tabla 2.

Tabla 2. Valores de similitud del tópico 9 del periodo 1, tópico 1 del periodo 2, tópico 6 del periodo 3 con respecto al tópico inicial 0 del periodo 0.

Tópico 0 del periodo 0	Similitud entre el Tópico 0 del periodo 0 con el Tópico 9 del periodo 1	Similitud entre el Tópico 0 del periodo 0 con el Tópico 1 del periodo 2	Similitud entre el Tópico 0 del periodo 0 con el Tópico 6 del periodo 3
0 – 0	9 – 1	1 – 2	6 – 3
Valor de Similitud	0.827879879879879	0.844138138138138	0.835371371371371

El valor de similitud identifica que tan cerca se encuentra el tópico principal que se quiere comparar, con respecto a los tópicos existentes en otros periodos, la similitud pude ser:

- Alta: Cercana a 0 o muy parecidos.
- Promedio: mayor a 0 y menor a 0.85.
- Lejana: mayor a 0.85 y menor a 1.

En la Figura 30, Figura 31, Figura 32 se visualizan el comportamiento de tópicos similares en periodos distintos, donde el número de noticias relacionadas a un tópico diferente del inicial pueden variar la importancia de los tópicos:

- Aumentando y dando importancia
- Aumenta en un instante de tiempo y después disminuye su importancia.
- Simplemente disminuye su importancia.

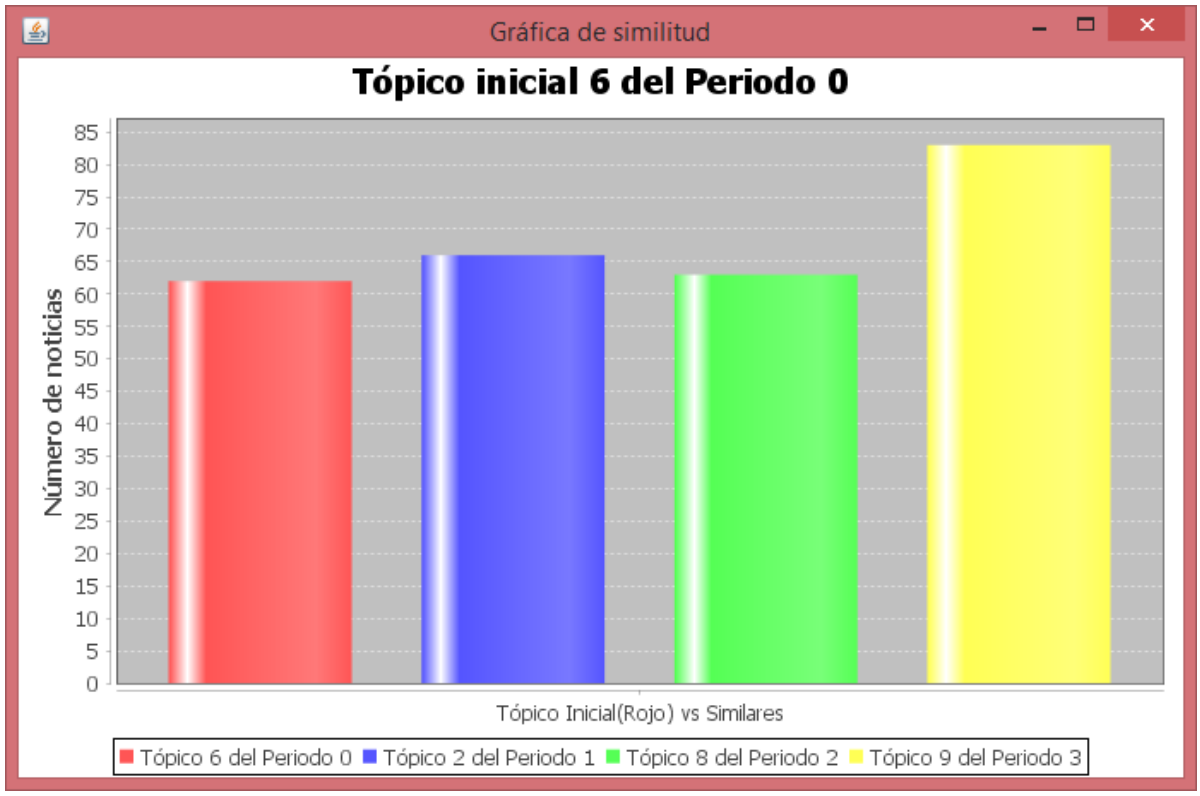


Figura 30. Número de noticias del tópico 2 del periodo 1, tópico 8 del periodo 2, tópico 9 del periodo 3 similares al tópico inicial 6 del periodo 0, con valor de similitud <= 0.85

Los valores de similitud del tópico 2 del periodo 1, tópico 8 del periodo 2, tópico 9 del periodo 3 con respecto al tópico inicial 6 del periodo 0, se pueden ver en la Tabla 3.

Tabla 3. Valores de similitud del tópico 2 del periodo 1, tópico 8 del periodo 2, tópico 9 del periodo 3 con respecto al tópico inicial 6 del periodo 0.

Tópico Inicial 6 del periodo 0	Similitud entre el Tópico 6 del periodo 0 con el Tópico 2 del periodo 1	Similitud entre el Tópico 6 del periodo 0 con el Tópico 8 del periodo 2	Similitud entre el Tópico 6 del periodo 0 con el Tópico 9 del periodo 3
6 – 0	2 – 1	8 – 2	9 – 3
Valor de Similitud	0.838564564564564	0.844828828828828	0.831219219219219

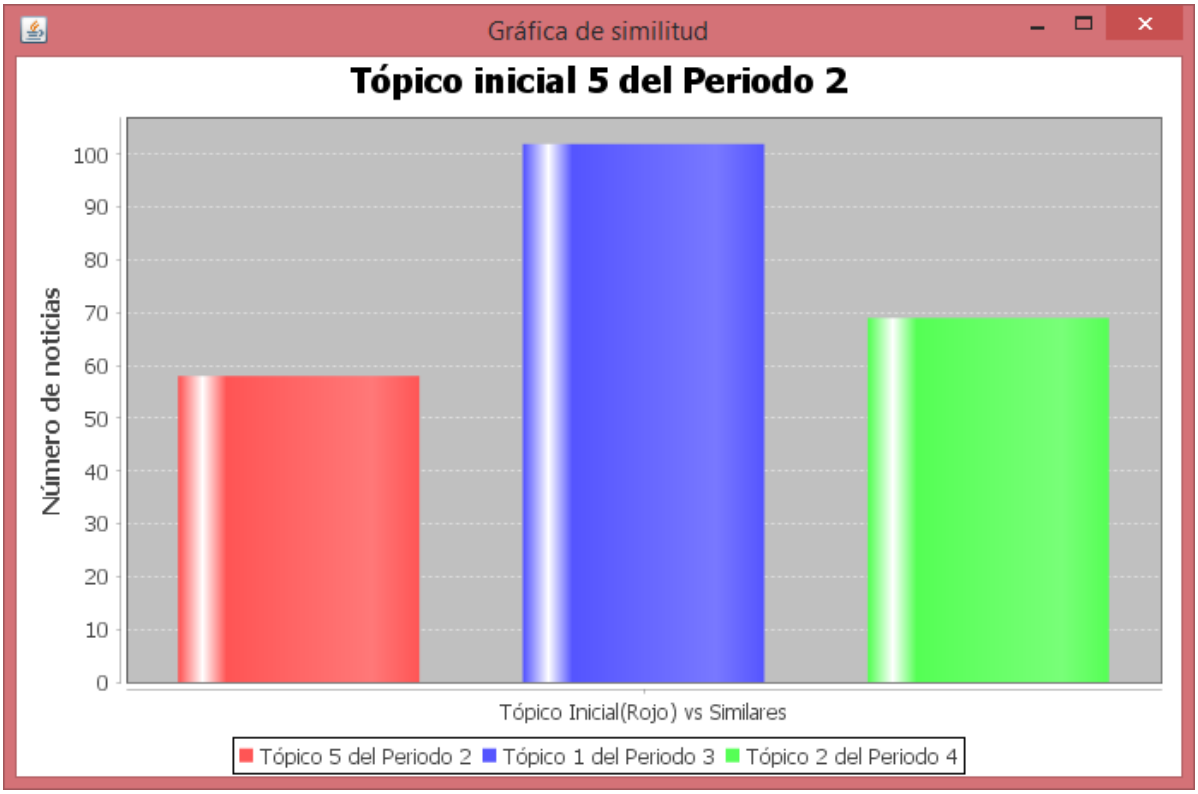


Figura 31. Número de noticias del tópico 1 del periodo 3, tópico 2 del periodo 4 similares al tópico inicial 5 del periodo 2, con valor de similitud <= 0.85

Los valores de similitud del tópico 1 del periodo 3, tópico 2 del periodo 4 con respecto al tópico inicial 5 del periodo 2, se pueden ver en la Tabla 4.

Tabla 4. Valores de similitud del tópico 1 del periodo 3, tópico 2 del periodo 4 con respecto al tópico inicial 5 del periodo 2,

Tópico 5 inicial del periodo 2,	Similitud entre el Tópico 5 del periodo 2 con el Tópico 1 del periodo 3	Similitud entre el Tópico 5 del periodo 2 con el Tópico 2 del periodo 4
5 – 2	1 – 3	2 – 4
Valor de Similitud	0.772930930930930	0.832144144144144

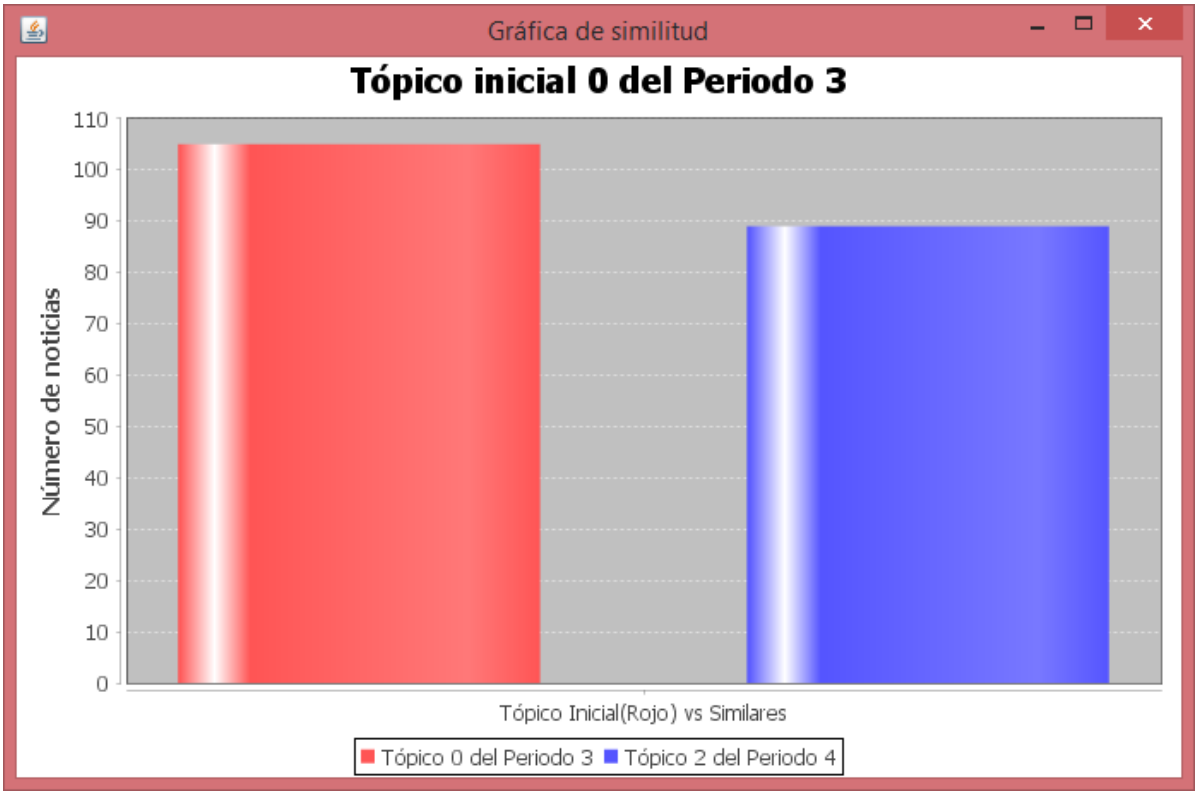


Figura 32. Número de noticias del tópico 2 del periodo 4 similar al tópico inicial 0 del periodo 3, con valor de similitud <= 0.85

Los valores de similitud del tópico 2 del periodo 4 con respecto al tópico inicial 0 del periodo 3, se pueden ver en la Tabla 5.

Tabla 5. Valores de similitud del tópico 2 del periodo 4 con respecto al tópico inicial 0 del periodo 3

Tópico Inicial 0 del periodo 3	Similitud entre el Tópico 0 del periodo 3 con el Tópico 2 del periodo 4
0 – 3	2 – 4
Valor de Similitud	0.835915915915915

Para entender fácilmente las gráficas de similitud, se puede analizar el árbol de periodos, tópicos y palabras (ver Figura 19), con el cual se podrá ver, por qué el tópico inicial visualizado por la barra en rojo tiene similitud a los tópicos de periodos siguientes (barra azul, barra verde barra amarilla, etc.), al observar las palabras que se encuentran en ambos tópicos similares.

Se desarrolló otra representación de la similitud entre los tópicos y poder conocer cuáles y cuantas palabras son iguales entre el tópico inicial y los tópicos similares, mediante un listado de su posición dentro de ambos tópicos, para dar un ejemplo se utilizan la gráfica anterior (ver Figura 29) donde el tópico 0 del periodo 0 es parecido al tópico 9 del periodo 1 y como resultado obtenemos una ventana con palabras iguales dentro de ambos tópicos y en qué posición se localizan (ver Figura 33).

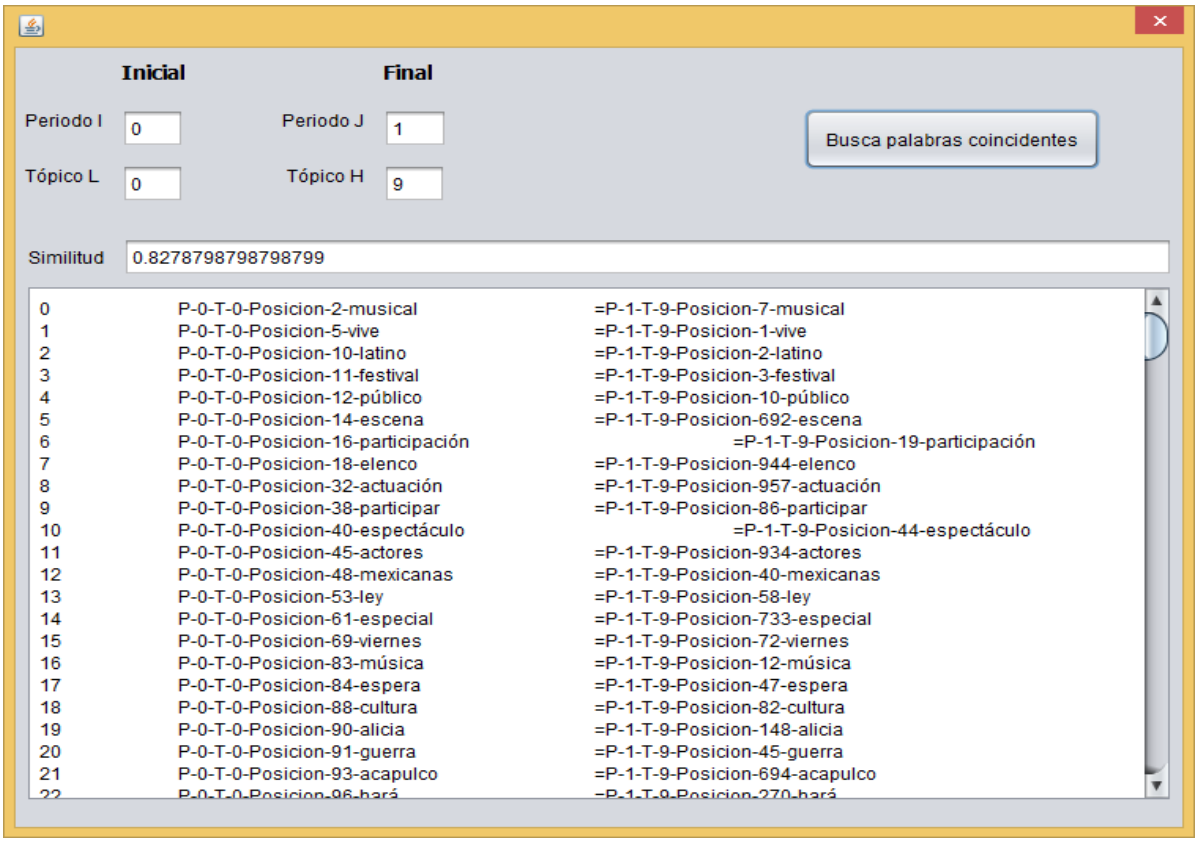


Figura 33. Lista de palabras entre tópicos con similitud <= 0.85, tópico 0 del periodo 0 con el tópico 9 del periodo 1

Los resultados dentro de la ventana se visualizan en un listado de palabras similares indicando:

- Cada línea el número del renglón, seguido por el periodo inicial, el tópico inicial, la posición dentro del total de palabras en el tópico inicial y la palabra a la que se refiere.
- Por otro lado se encuentra el periodo similar, el tópico similar y la posición dentro de todas las palabras en el tópico similar junto con la palabra similar. Si se desea corroborar los resultados de estas palabras se pueden observar dentro del árbol de periodos, tópicos y palabras generado ejemplo de la Figura 19.

Ejemplo:

1 P-0-T-0-Posicion-5-vive =P-1-T-9-Posicion-1-vive

P-0 representa el periodo 0 inicial.

T-0 es tópico 0 inicial del periodo 0.

P-1 representa el periodo 1 similar.

T-9 el tópico 9 del periodo 1 similar al tópico 0 del periodo 0

Posición n, de la palabra *w* inicial y similar.

"Vive" es la palabra *w* que se encuentra en ambos tópicos inicial y similar

4.4.2 Gráfica de nube de palabras.

Otra visualización que se desarrolló mediante el programa (F), es una gráfica de nube de palabras, la cual determina mediante las 50 primeras palabras de cada tópico en un periodo de tiempo la importancia de un tema dentro de todos los periodos y con soporte en el número de noticias en las que se encuentran las palabras.

Por cada periodo se realiza una gráfica de nube que mediante el tamaño de la palabra se puede identificar si es más importante que las demás dentro de un mismo periodo.

El ejemplo de nube de palabras se puede ver en la Figura 34.

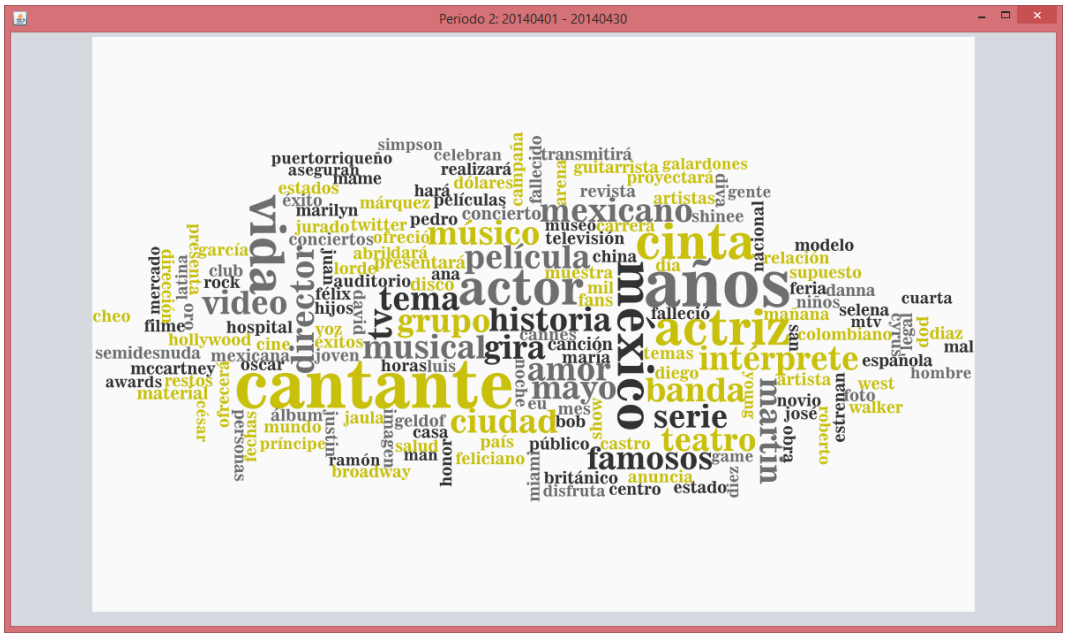


Figura 34. Visualización de la importancia mediante nube de palabras para un periodo P. donde se pude determinar que en un lapso de tiempo se **mantuvieron** *las palabras "cantante" y "méxico" como las más representativas de los tópicos del periodo P, mientras que las palabras "casa", "país", "latina" y "mal" son las menos importantes debido a su tamaño dentro de la nube.*

4.4.3 Gráfica de nube de palabras por tópicos.

Existen otras dos formas de visualizar la similitud entre dos tópicos de diferentes periodos.

- La primera, es mediante la gráfica de nube por cada uno de los tópicos similares al tópico inicial.
- La segunda, es mediante la visualización de nube de palabras que contiene la intersección de los tópicos similares en diferentes periodos.

Ejemplo de similitud mediante nube de palabras de un tópico general.

En la Figura 29. Se observa que el tópico 0 del periodo 0 tiene un cierto grado de similitud con el tópico 9 del periodo 1, que es respaldado mediante la Figura 33, otra forma de ver la similitud; es mediante la nube que genera el tópico 0 del periodo 0 y el tópico 9 del periodo 1 respectivamente ver Figura 35 y Figura 36.

Figura 35. Gráfica de nube de palabras sobre el tópico 0 del periodo 0 que será comparada con la gráfica de la Figura 36 para así poder encontrar similitud entre los tópicos 9 y 0 respectivamente.

Figura 36. Gráfica de nube de palabras sobre el tópico 9 del periodo 1 que será comparada con la gráfica de la Figura 35 para así poder encontrar similitud entre los tópicos 9 y 0 respectivamente.

Mediante esta visualización de nubes nos percatamos que la similitud entre tópicos de diferentes periodos es difícil de interpretar si se cuenta con la colección de palabras completas en el tópico, para lo cual se realizó la segunda visualización.

Ejemplo de visualización con nube mediante la intersección de las palabras en ambos tópicos de diferentes periodos.

Visualizar el total de palabras de un tópico en una sola nube no permite entender la similitud entre dos tópicos como se muestra en la Figura 33, para lo cual se realizó la visualización de la intersección de las palabras que se encuentran en ambos tópicos de diferentes periodos ver Figura 37 y Figura 38.

Figura 37. Palabras intersectadas de dos tópicos similares donde se puede observar la semejanza del tópico 0 del periodo 0 con el tópico 9 del periodo 1.

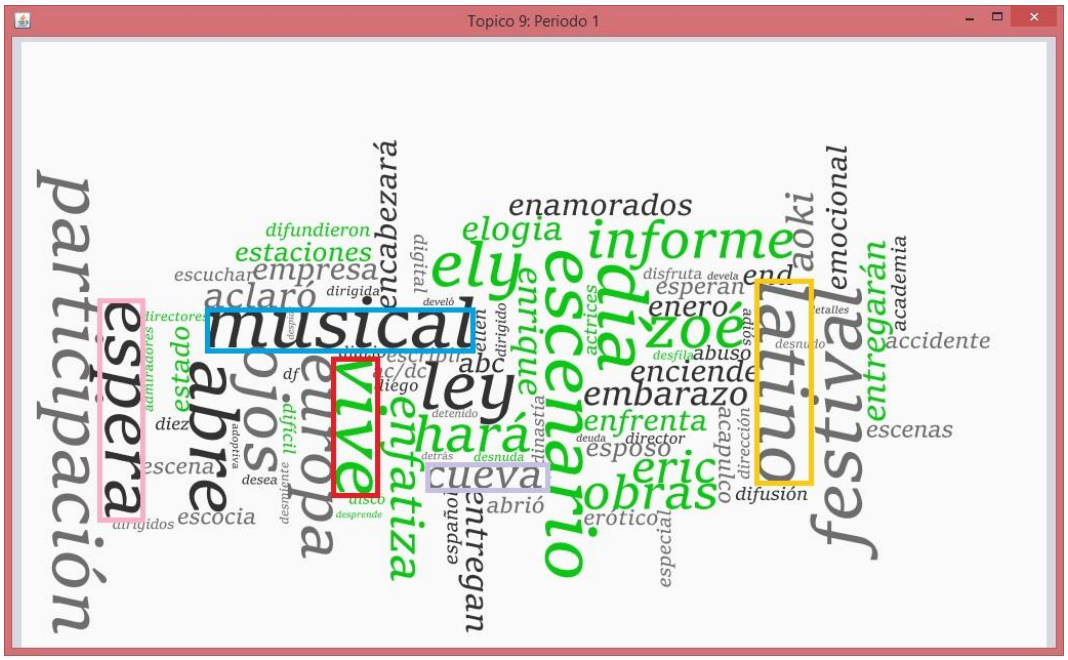


Figura 38. Palabras intersectadas de dos tópicos similares donde se puede observar la semejanza del tópico 0 del periodo 0 con el tópico 9 del periodo 1.

En estas dos figuras se puede observar mucho mejor la similitud entre ambos tópicos de periodos distintos, ya que se pueden ver las palabras iguales que existen en ambos tópicos, pero aún no es fácil identificar el total de las palabras iguales que se encuentran en los dos tópicos; por lo tanto podemos determinar que la visualización por nube ayuda a determinar importancia en los tópicos, pero no es tan útil para hallar similitud entre tópicos de distintos periodos como es el caso de la Figura 33, donde se observan todas las palabras y su posición de cercanía con respecto a ambos periodos.

Capítulo 5: Pruebas y Resultados.

En este capítulo se describen algunas pruebas y resultados que se obtienen durante el desarrollo del trabajo de tesis, algunas muestran el rendimiento que existe en memoria al utilizar algún modelo y otras pruebas determinan si la cantidad de resultados arrojados son correctos.

5.1 Rendimiento del modelo LDA en cuanto a tópicos conocidos.

Para comprobar el correcto funcionamiento del modelo LDA nos dimos a la tarea de realizar una prueba, que consiste en determinar las Palabras_por_Tópico y Noticias_por_Tópico, mediante el título y cuerpo de todas las noticias sin definir de qué sección del periódico se estaban tomando las noticias. Con esta prueba podremos determinar si la clasificación de noticias por el algoritmo utilizado del modelo LDA es correcta durante un solo periodo de tiempo (todo el periodo de recolección) dentro de los k tópicos que un usuario determino recolectar, es decir, clasificar bien en 4 tópicos el total de 28130 noticias que se encuentran en la base de datos.

Los resultados que nos proporcionó la base de datos, sobre cuantas noticias existen de cada uno de los temas son las siguientes (ver Tabla 6).

Tabla 6. Total de noticias por sección según la base de datos.

Sección	Número de noticias en la base de datos
Nación México	10806
Computación y Tecnología	3167
Ciencia	3206
Espectáculos	10978

Se realizó una prueba al modelo LDA simulando las cuatro secciones del periódico y mediante ella poder determinar ¿Qué tan eficiente es el modelo? Para lo cual se designó todo el tiempo de recolección como un solo periodo y que clasifique en 4 tópicos, los cuales se pretende que

representen cada sección del periódico, los resultados de recolección y clasificación fueron los siguientes (ver Tabla 7).

Tabla 7. Resultado de LDA para 4 tópicos

Tópico k	Número de noticias LDA	Sección	Número de Noticias en la base de datos
Tópico 0	6651	Nación México	10806
Tópico 1	6591	Computación y Tecnología	3167
Tópico 2	4819	Ciencia	3206
Tópico 3	10069	Espectáculos	10978

Recordando que los tópicos generados por el modero LDA no son etiquetados hasta analizar su contenido, por lo cual nos guiamos por las primeras palabras que aparecen en el archivo de salida que genera el modelo (ver Figura 39), que también son mostradas de mayor a menor importancia dentro de un árbol parecido de periodos, tópicos y palabras (ver Figura 19).

```
PTopic 0: méxico peña mil país presidente nieto nacional gobierno seguridad federal estado eu estados pgr mexic
PTopic 1: años cantante actriz actor vida estudio facebook cáncer twitter red imagen redes personas imágenes in
PTopic 2: prd senado nacional pan reforma diputados presidente pri cámara partido ley energética comisión líder
PTopic 3: cantante méxico cinta años festival cine actriz actor disco the película banda mexicano concierto mús
```

Figura 39. Resultados de las palabras representativas de los tópicos sobre todas las noticias con 4 tópicos de un periodo

Podemos analizar e interpretar los resultados que obtenemos por medio de la salida del programa (E) en la Tabla 8. en esta tabla se determinan las clasificaciones sobre los temas que representa cada tópico etiquetado manualmente.

Tabla 8. Temas etiquetados manualmente que indican la el contenido de cada tópico

Tópico k	Número de noticias LDA	Tema	Sección	Número de noticias en la base de datos
Tópico 0	6651	Nación México	Nación México	10806
Tópico 1	6591	Ciencia, Computación y Tecnología	Computación y Tecnología	3167
Tópico 2	4819	Nación México	Ciencia	3206
Tópico 3	10069	Espectáculos	Espectáculos	10978

Los resultados muestran una conclusión en la cual nos percatamos que al tener como sección de noticias dos temas muy parecidos como es Ciencia, Computación y Tecnología, el modelo LDA interpretará la clasificación como si fuera el mismo tópico general.

También al momento de clasificar dos secciones como un solo tópico, nos percatamos que el modelo divide una sección en dos tópicos distintos ya que puede tener algunas diferencias particulares, como es el caso de la sección Nación México.

Por lo tanto si sumamos el número de noticias en la Base de datos de Computación y Tecnología y Ciencia, tendremos los siguientes resultados (ver Tabla 9).

Tabla 9. Generalización de Noticias en la base de datos y LDA

Tópico k	Número de noticias LDA	Tema	Sección	Número de noticias en la base de datos
Tópico 0	6651	Nación México	Nación México	10806
Tópico 1	6591	Ciencia, Computación y Tecnología	Computación y Tecnología + Ciencia	3167 + 3206 =6373
Tópico 2	4819	Nación México	-	-
Tópico 3	10069	Espectáculos	Espectáculos	10978

El resultado de sumar los números de noticias generados por la Base de datos entre tópicos iguales, genera una aproximación muy parecida a los resultados obtenidos por LDA con respecto al número de noticias.

Los resultados mostrados en la Tabla 9, determinan la cercanía de los tópicos etiquetados manualmente con relación al número de noticias entre el modelo LDA y la colección de noticias en la base de datos sobre cada una de las secciones de los periódicos.

Los porcentajes de error generados por el modelo LDA con respecto a 4 tópicos se muestran en comparación al número real de noticias en la base de datos y el número de noticas generadas por el modelo LDA en la siguiente gráfica (ver Figura 40).

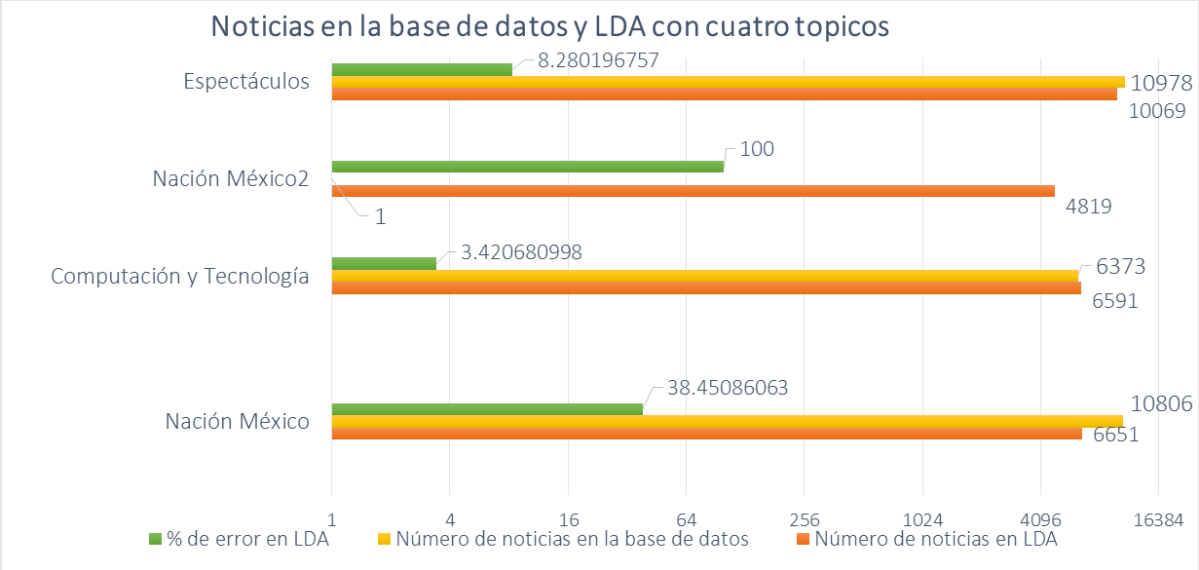


Figura 40. Comparativo en % de error en LDA, entre el número de noticias en la base de datos y el número de noticias generado por LDA con 4 tópicos

Para nuestra siguiente prueba se utilizaron solamente tres tópicos, un periodo y todo el conjunto de noticias; con el fin de determinar lo que en la prueba anterior se realizó manualmente, que fue sumar dos tópicos para comprobar la cantidad de noticias que se clasificaron de manera correcta en la sección de Computación y Tecnología y Ciencia (ver Tabla 10).

Tabla 10. Prueba 2 de LDA con tres tópicos

Tópico k	Número de noticias LDA	Sección	Número de noticias en la base de datos
Tópico 0	10616	Nación México	10806
Tópico 1	6819	Computación y Tecnología	3167
		Ciencia	3206
Tópico 2	10695	Espectáculos	10978

Continuando con el error de elegir dos temas muy parecidos (Ciencia, Computación y Tecnología), solo queda sumar las noticias en la Base de datos de Ciencia y Computación y Tecnología para determinar cómo se comporta el modelo LDA, con los tres tópicos seleccionados lo cual se puede observar (ver Tabla 11).

Tabla 11. Generalización de Noticias en la base de datos y LDA con 3 tópicos

Tópico k	Número de noticias LDA	Tema	Sección	Número de noticias en la Base de datos
Tópico 0	10616	Nación México	Nación México	10806
Tópico 1	6819	Ciencia o Computación y Tecnología	Ciencia o Computación y Tecnología	3167+ 3206 = 6373
Tópico 2	10695	Espectáculos	Espectáculos	10978

Los porcentajes de error generados por el modelo LDA con respecto a 3 tópicos se muestran en comparación al número real de noticias en la base de datos y el número de noticias generadas por el modelo LDA en la siguiente gráfica (ver Figura 41).

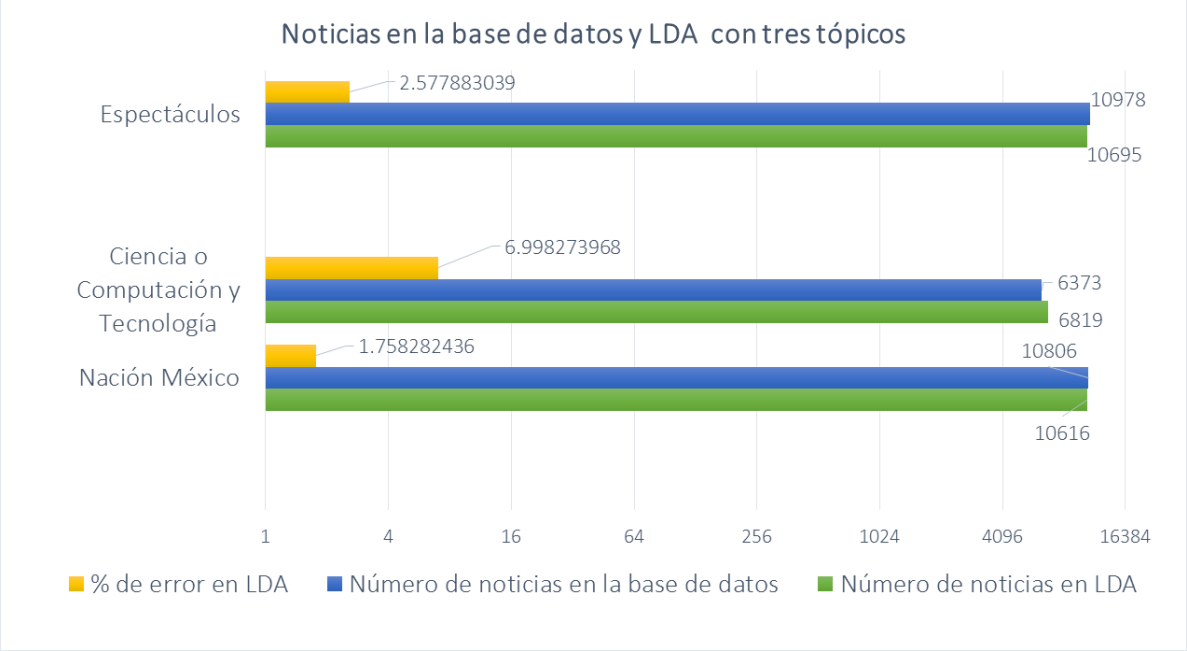


Figura 41. Comparativo en % de error en LDA, entre el número de noticias en la base de datos y el número de noticias generado por LDA con 3 tópicos

Donde se puede observar claramente que los porcentajes de error obtenidos por los resultados del periódico RSS y el modelo LDA sobre el número de noticias existentes en cada sección definida por el modelo es bajo.

5.2 Comportamiento de memoria durante la utilización del programa (E) que utiliza el modelo LDA.

El comportamiento de memoria nos permite identificar por qué los tweets son una fuente muy grande de información que es difícil de tratar, debido a la cantidad reducida con la que se cuenta de memoria y el número tan grande que se cuenta de tweets para un periodo de tiempo; lo que nos llevó a la utilización de noticias como fuente principal de información.

Para realizar la medición del rendimiento en memoria se utilizó toda la Base de datos de noticias con la que se cuenta (28130 noticias) que fueron descritas con anterioridad, utilizando las cuatro secciones del periódico conjuntamente y los recursos en de una laptop Intel i5 de 64bits con 8GB de RAM, se recolectaron resultados de 4 tópicos en una revisión con alrededor de 48000 tokens en el diccionario.

Se logró visualizar que la utilización de memoria para generar una MTD con su índice invertido, genera un consumo grande de memoria.

Se muestra paso a paso el consumo de memoria conforme el modelo LDA lo va solicitando.
En el primer paso (ver Figura 42). Se observa cómo se comporta nuestro sistema al no ejecutar el sistema VITOIM, ni el modelo LDA.

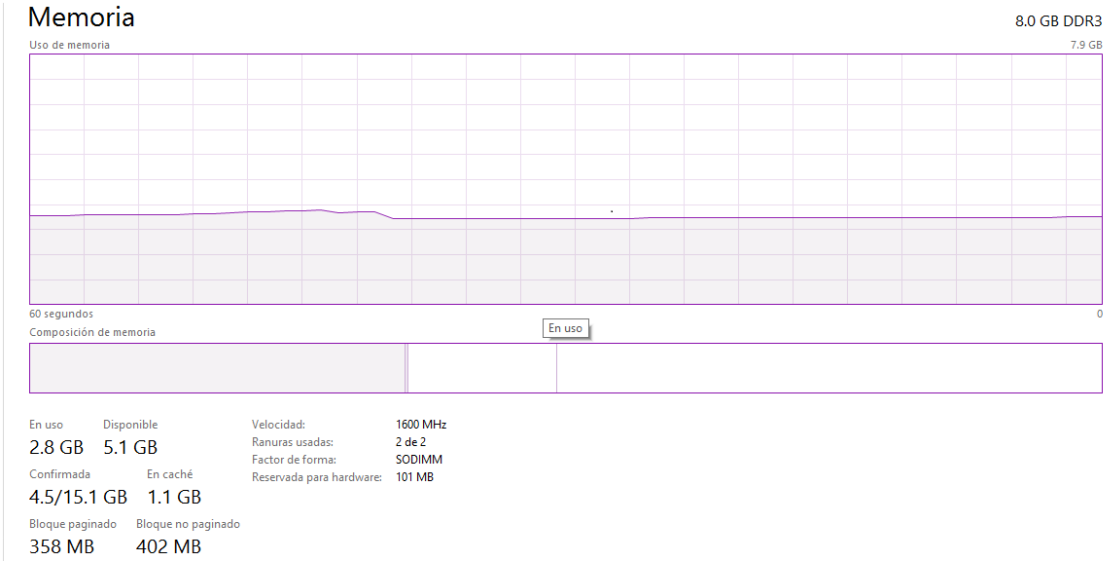


Figura 42. Recursos de memoria RAM sin utilizar el programa E.

En el segundo paso se muestra la creación del vocabulario de tokens o palabras del que son almacenadas en memoria para su primera evaluación antes de extender la MTD y el índice invertido (ver Figura 43).

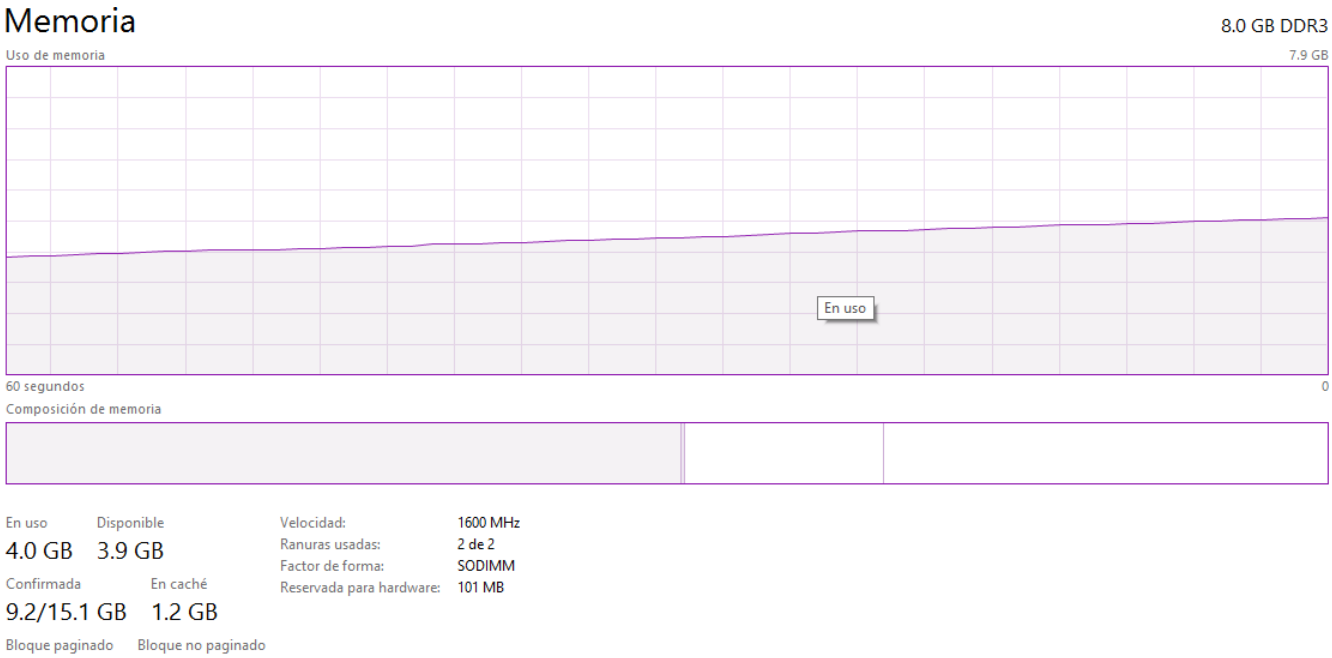


Figura 43. Recursos de memoria RAM utilizados por parte del programa E.

El programa (E) que utiliza el índice invertido que se encuentra en el archivo ".ldac" y generando una matriz total de palabras y el número de sus apariciones en las noticias para que el modelo LDA pueda realizar las probabilidades sobre las palabras que existen en los documentos, aquí es la parte donde se requiere el tamaño máximo de memoria disponible debido a que la matriz generada es de tamaño *M* Documentos * *N* Palabras, lo cual genera un consumo de memoria aproximado de 4GB sin contar el uso de palabras del segundo paso que llega a utilizar un 1GB de memoria (ver Figura 44).

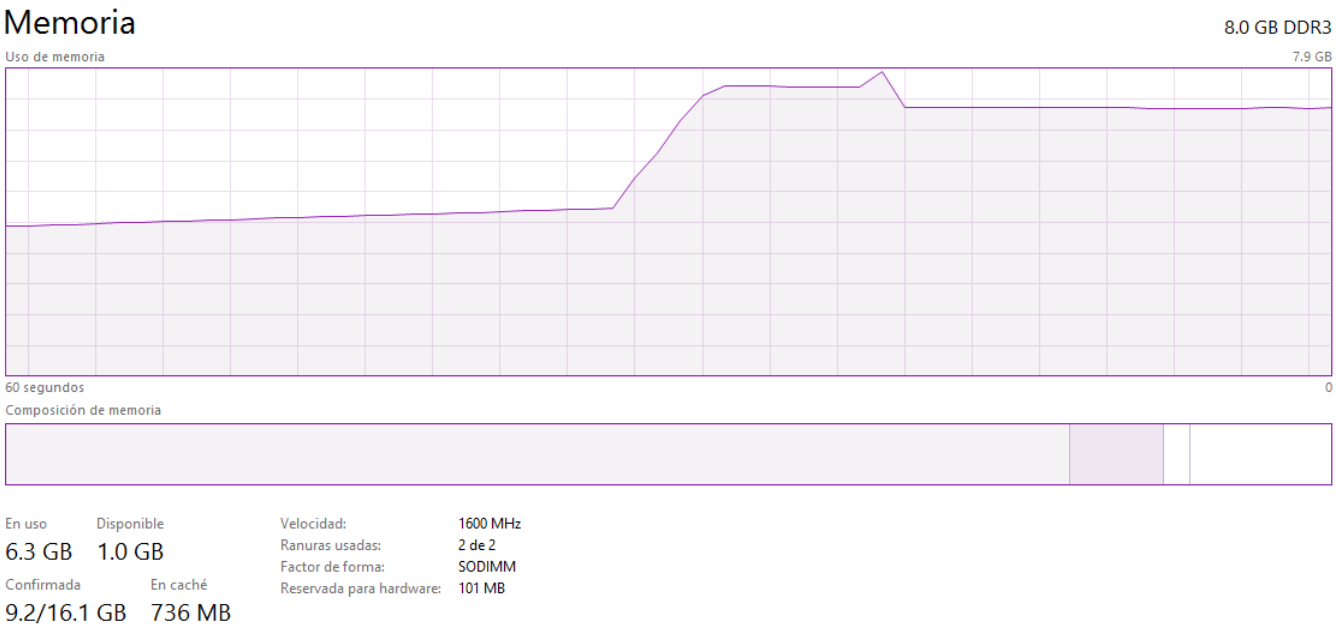


Figura 44. Uso de memoria RAM al implementar el índice invertido dentro del programa E.

Por último el programa (E) liberá la memoria para ser utilizada por el sistema VITOIM después de 5 minutos en procesar toda la colección de noticias (ver Figura 45).

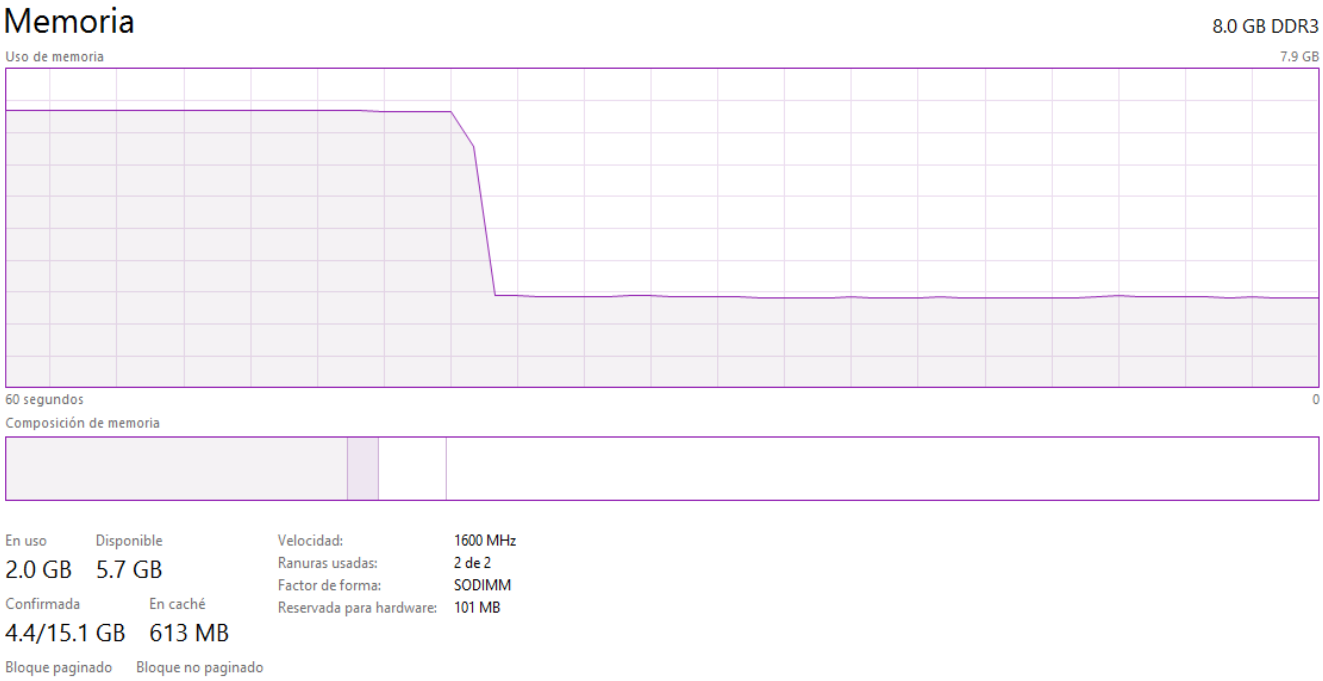


Figura 45. Liberación de memoria al término de la implementación del programa (E), liberando el espacio..

Este uso excesivo de memoria durante un periodo largo de tiempo, fue el motivo por el cual se decidió también dividir el problema de verificar que tópicos son importantes dentro de toda la colección de noticias.

Es decir que el tiempo se acorta, dependiendo del número de noticias que se pretendan analizar y así utilizar periodos de tiempo más cortos como son meses, quincenas o semanas; con lo cual se procesan una menor cantidad de noticias. Esta división de noticias en periodos, nos da la facilidad de entender como el tópico va cambiando a través el tiempo, debido a que una noticia de un mismo tópico puede surgir en un diferente lapso de tiempo y con ello poder analizar lo que nosotros consideramos como semejanza entre tópicos de periodos distintos.

Capítulo 6: Aportación computacional y Conclusiones.

En este capítulo se describe brevemente la aportación computacional que se obtuvo en este trabajo de tesis, así como las conclusiones a la que se llegó, utilizando el conocimiento aplicado de las diferentes disciplinas computacionales que se cursaron durante el desarrollo del trabajo de tesis y tener una perspectiva amplia sobre los resultados favorables al ser utilizadas.

6.1 Aportación.

La aportación computacional que se obtuvo, fue el modelo de similitud entre tópicos que resultó de gran utilidad a pesar de ser una forma sencilla de medir como cambia un tópico a través del tiempo, determinando si los tópicos decaen y cambian hasta el punto de envejecer o de ser tópicos completamente distintos. Se puede determinar que no todos los tópicos permanecen constantes durante un periodo largo de tiempo, lo que implica también, que el contenido cambia con el tiempo.

Se pudo obtener una similitud que nos determina si existen palabras iguales en tópicos de distintos periodos y poderlos caracterizar en un intervalo de [0, 1], mediante este intervalo se determina que tan cerca o lejos se encuentra un tópico de otro, debido a que ellos son publicados en diferente fechas y posiblemente por diferentes escritores de una misma sección.

La visualización de las gráficas generadas a partir del modelo de similitud ayuda a interpretar el comportamiento de algún tópico semejante que existe con el paso del tiempo, en el caso de que la importancia en el tópico aumente, continúe o disminuya.

6.2 Conclusión

Identificar los tópicos que pueden ser importantes en el ámbito de la ciencia, la computación, la tecnología, los espectáculos y la nación. Haciendo hincapié en las palabras más importantes que aparecen dentro de un tópico y con ellas determinar dentro de alguna visualización porque un tópico logra ser más importante en un periodo de tiempo que otro gracias a la utilización de un modelo conocido como LDA.

Las noticias son una fuente constante de información, que puede ser de un volumen difícil de procesar por un ser humano o simplemente no se tiene la oportunidad de examinar la totalidad de dicha información. Por lo tanto fue de utilidad realizar una herramienta que faciliten a los lectores o usuarios de medios sociales el análisis de temas importantes o de interés sobre sucesos que ocurren en su entorno, y mantenerse al tanto de los hechos actuales que suceden en el país.

Las nubes de palabras muestran en un gran contexto cómo se comporta el medio social dentro de un determinado momento del tiempo, con las palabras se puede determinar si la existencia de un tópico contiene ciertas palabras en común más frecuentes y poder interpretar su importancia conforme a las palabras de mayor tamaño que aparecen dentro de la nube.
Las nubes de palabras sirven de ayuda en la importancia de tópicos, pero no sirve para interpretar una visualización textual sobre palabras semejantes, debido a que la nube contendrá palabras similares de distintos tamaños que no son fácilmente identificables por el ojo humano al comparar dos nubes que pertenece a tópicos similares de periodos distintos.

La ventaja que proporciona el sistema de visualización de la información de tópicos más importantes generados en medios sociales, no solamente es determinar la importancia de un tópico, sino también el comportamiento que se da al encontrar tópicos semejantes en distintos periodos de tiempo, pudiendo determinar cómo es que el tópico envejece con el cambio de las palabras que se encuentran dentro de él.

Capítulo 7: Trabajo futuro.

En este capítulo se describe brevemente el trabajo futuro que se plantea resolver después de haber concluido satisfactoriamente el desarrollo de la tesis. Son algunos objetivos que se plantean para continuar con este trabajo y poder entregar aún mejores resultados que den una idea novedosa a lo ya realizado.

7.1 Trabajo futuro

Recolectar noticias no solamente de cuatro secciones del periódico, también de salud, o que contengan más información de interés.

En este momento la importancia de los tópicos está basada por la frecuencia de las noticias y un número de palabras que generan el tópico.

- El trabajo futuro es utilizar el valor semántico de las palabras como:
    - El daño económico en el precio de 100 computadoras laptop dañadas por una falla tecnológica a 100 automóviles por una falla tecnológica.
    - De manera similar el número de vidas afectadas por noticias de diferentes enfermedades como el chikungunya vs dengue.
    - Número de delitos registrados en una zona en específica de la ciudad.

Visualizar los tópicos de interés debido a la semántica de las palabras, como es el número de 100 o cien; que puede tener más impacto en un acontecimiento, debido a que cien muertos en un accidente de avión es más impactante que 10 por un accidente automovilístico.

Perder una cantidad en dólares puede ser más importante que la misma cantidad en pesos.

Otro ejemplo es, 10 muertos por ébola contra 10 muertos por cáncer, debido a la peligrosidad de la enfermedad o la rapidez con la se propaga y muere un enfermo de ébola.

Referencias

[1] J. Han y M. Kamber, Data Mining: Concepts and Techniques, Elsevier Inc, 2006.

[2] G. Sidorov, Construcción no lineal de N-gramas en la lingüística computacional, México D.F., 2013.

[3] J. Zhang, Visualization for information retrieval, Berlin Heidelberg: Springer, 2008.

[4] «Acura retirá dos modelos por fallas en frenos,» *El economista,* 11 06 2015.

[5] «Usuarios de iPhone 6 se quejan porque su armazón se dobla fácilmente,» *CNN México,* p. 1, 24 09 2014.

[6] J. Flores, «El primer caso de ébola diagnosticado en Norteamérica,» *La Jornada,* 07 10 2014.

[7] «Más de 100 personas mueren por influenza A H1N1 en una semana,» *CNN México,* 09 02 2014.

[8] «Presentan operativo de seguridad Tláhuac-Chalco,» *El Universal,* 27 05 2015.

[9] J. Fiaidhi, S. Mohammed, A. Islam, S. Fong y T.-h. Kim, «Developing a Hierarchical Multi-Label Classifier for Twitter Trending Topics,» *International Journal of u- and e-Service, Science and Technology,* p. 13, 2013.

[10] K. Lee, D. Palsetia, R. Narayanan, M. A. Patwary, A. Agrawal y A. Choudhary, «Twitter Trending Topic Classification,» *11th IEEE International Conference on Data Mining Workshops,* p. 8, 2011.

[11] H. Becker, M. Naaman y L. Gravano, «Beyond Trending Topics: Real-World Event Identification on Twitter,» *Association for the Advancement of Artificial Intelligence,* p. 11, 2011.

[12] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman y J. Sperling, «TwitterStand: news in tweets,» *ACM GIS'09,* p. 10, 2009.

[13] Z. Li, J. Liu, X. Zhu y H. Lu, «Multi-Modal Multi-Correlation Person-Centric News Retrieval,» *CIKM Toronto,* p. 10, 2010.

[14] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan y X. Li, «Comparing Twitter and Traditional Media using Topic Models,» *Advances in Information Retrieval,* 2011.

[15] C. D. Manning, P. Raghavan y H. Schütze, Introduction to Information Retrieval, Cambridge University, 2008.

[16] D. M. Blei, A. Y. Ng y M. I. Jordan, «Latent Dirichlet Allocation,» *Journal of Machine Learning Research 3,* p. 30, 2003.

Apéndice.

Código sobre el programa LDA de Python.

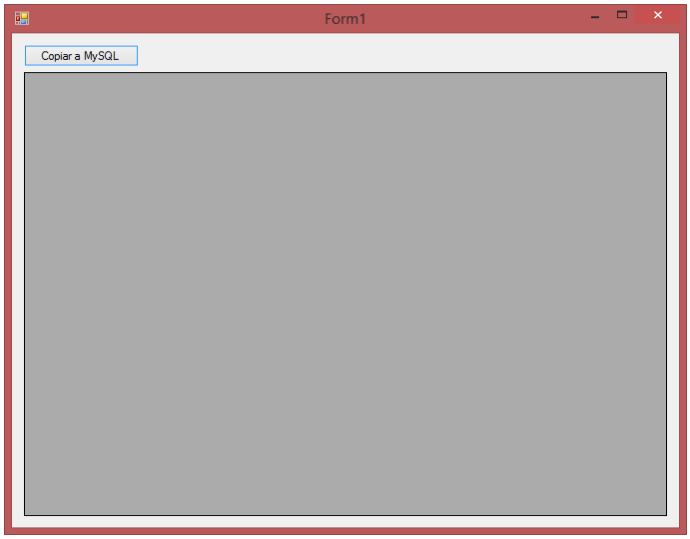
```
import numpy as np
import lda
import lda.datasets
import sys

#Metodo que determina digitos en las palabras
def contains_digits(s):
   for char in list(s):
      if char.isdigit():
         return True
         break
   return False

def Inicio(n_t, n_t_w):
            #inicio del LDA
            X = lda.datasets.load_reuters()
            vocab = lda.datasets.load_reuters_vocab()
            titles = lda.datasets.load_reuters_titles()
            X.shape
            #numero de topicos por seccion
            n_topics=n_t
            #  numero de Noticias, Numero de Palabras utilizadas
            model = lda.LDA(n_topics, n_iter=500, random_state=1)
            # model.components_ also works
            model.fit(X)
            topic_word = model.topic_word_
            #posible imprimir todo el vocabulario
            n_top_words = n_t_w
            for i, topic_dist in enumerate(topic_word):
                        topic_words = np.array(vocab)[np.argsort(topic_dist)][:-n_top_words:-1]
                        print('Palabras_por_Tópico {}: {}'.format(i, ' '.join(topic_words)))

            doc_topic = model.doc_topic_
            ListaNormal=[]
            for i in range(n_t):
                        ListaNormal.append(0)
            #ListaEspectaculos=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

            #Impresion de los  Palabras_por_Tópico y Ntopic CTopic
            for i in range(len(titles)):
                        print("NTopic {}:  {}".format(doc_topic[i].argmax(), titles[i]))
                        ListaNormal[doc_topic[i].argmax()]=ListaNormal[doc_topic[i].argmax()]+1
            for i in range(0,n_topics):
                        print "CTopic %d: %d" % (i, ListaNormal[i])

if __name__ == '__main__':
            Inicio(int(sys.argv[1]), int(sys.argv[2]))
   #intro2()
```

Programa en C# que transforma RSS de Firebird a MySQL.

Cadena de conexión a Firebird.

```
public string connectionString =
        "User=SYSDBA;" +
        "Password=masterkey;" +
        "Database=C:\\TOSHIBA\\noticias.fdb;" +
        "DataSource=localhost;" +
        "Port=3050;" +
        "Dialect=3;" +
        "Charset=ISO8859_1;" +
        "FbCharset=None;" +
        "Role=;" +
        "Connection lifetime=15;" +
        "Pooling=true;" +
        "MinPoolSize=0;" +
        "MaxPoolSize=100;" +
        "Packet Size=16000;" +
        "ServerType=0";
```

Imagen de la interfaz que realiza ETL de Firebird a MySQL



Código que realiza el ETL de Firebird a MySQL.

```
private void copiar_a_mysql(object sender, EventArgs e)
    {
      conn.MySqlCon.Open();
      FbConnection connection = new FbConnection(conn.connectionString);
      ArrayList objetos = new ArrayList();
      MessageBox.Show("Conexion exitosa");
      connection.Open();
      string sql = "select id, title, body, link, creator, created from articles where id>28769";
      FbCommand command = new FbCommand(sql, connection);
      FbDataReader reader = command.ExecuteReader();
      while (reader.Read())
      {
        try
        {
          for (int i = 0; i < reader.FieldCount; i++)
          {
            objetos.Add(reader.GetValue(i));
          }
          if (objetos[1].Equals(" ") || objetos[1].Equals(""))
          { }
          else
          {
            inserta(objetos);
          }
          objetos.Clear();
        }
        catch (Exception ex)
        {
          MessageBox.Show(ex.Message);
          //connection.Close();
        }
      }
      connection.Close();
      conn.MySqlCon.Close();

      MessageBox.Show("Conexion cerrada");
    }
    private void inserta(ArrayList Inser)
    {
      try
      {
        MySqlCommand comm = conn.MySqlCon.CreateCommand();
        string ins = Inser[0] + ",'" + remplaza(Inser[1].ToString()) + "','" + remplaza(Inser[2].ToString())
          + "','" + remplaza(Inser[3].ToString()) + "','" + remplaza(Inser[4].ToString()) + "','" +
remplazahora(Inser[5].ToString());

        comm.CommandText = "INSERT INTO noticiasviejas VALUES(" + ins + "')";
        comm.ExecuteNonQuery();
      }
      catch (Exception ex)
      {
        MessageBox.Show(ex.Message);
      }
    }
```

Código de recepción de twitter con PHP.

```
<?php
include 'inc/conexion.php';
include 'inc/twitteroauth.php';
define('BASE', dirname(__FILE__));
define('APP_CONSUMER_KEY', 'SNKhSsLZOdQL6qVhSEA');
define('APP_CONSUMER_SECRET', 'Ki7nIzsNwqjTLZjyrLJKqrZUaokTMPWqVnFMOa2pKw');
define('ACCESS_TOKEN', '2207569148-8YWWRZ4NibL847X6zBE3PGmyBWRlDCF9Cc9Lpw6');
define('ACCESS_TOKEN_SECRET', 'FtNoFVVoKAiZe2BubxEBmgu7HwWr5C5hP5DmG8cEWLy0F');

// Parsear el texto del tweet, aÃ±adiendo menciones, links y hashtags
function tweet_text($text) {
   // urls: lo primero para que no conviertan los links posteriores
   $text = preg_replace("/(http:\/\/[^\s]+)/", '<a href="$1" class="tweet-link" target="_blank">$1</a>', $text);
   // menciones
   $text = preg_replace("/(^|\W)@([A-Za-z0-9_]+)/", '$1<a href="http://twitter.com/$2" class="tweet-mention"
target="_blank">@$2</a>', $text);
   // hashtags
   $text = preg_replace("/(^|\W)#([^\s]+)/", '$1<a href="?q=%23$2" class="tweet-hash" target="_blank">#$2</a>', $text);
   return $text;
}

function guarda_tweet($r, $v) {
   if (count($r->statuses)):
      foreach ($r->statuses as $tweet):
         if ($tweet->{'retweeted'} == "") {
            $retweeteado = 0;
         } else {
            $retweeteado = 1;
         }
         if ($tweet->{'favorited'} == "") {
            $favorito = 0;
         } else {
            $favorito = 1;
         }
         if ($tweet->{'place'} == "") {
            $place = "";
         } else {
            $place = $tweet->{'place'}->{'country'};
         }
         if ($tweet->{'geo'} == "") {
            $geo = "";
         } else {
            $geo = $tweet->{'geo'}->{'type'};
         }
         if ($tweet->{'coordinates'} == "") {
            $coordinates = "";
         } else {
            $coordinates = $tweet->{'coordinates'}->{'type'};
         }
         if ($tweet->{'retweet_count'} == "100+") {
            $retweet_count = 100;
         } else {
            $retweet_count = $tweet->{'retweet_count'};
         }
         $texto = $tweet->{'text'};
         $texto = str_replace("'", "*", $texto);
```

```
        $cadena = "INSERT INTO tweets_tema VALUES ('" . $tweet->{'id_str'} . "','" . $coordinates . "',";
        $cadena.=$retweeteado . ",'" . $tweet->{'in_reply_to_status_id_str'} . "','";
        $cadena.=$tweet->{'in_reply_to_user_id_str'} . "'," . $retweet_count . ",'";
        $cadena.=$place . "','" . $tweet->{'contributors'} . "','" . $tweet->{'created_at'} . "',";
        $cadena.=$favorito . ",'" . strip_tags($tweet->{'source'}) . "','" . $geo . "','";
        $cadena.=$texto . "','" . $tweet->{'user'}->{'id_str'} . "', now(), 0,'" . $v . "')";
        $consulta = mysql_query($cadena);
        if (!$consulta) {
           //echo  mysql_error()."<br>".htmlentities($cadena);
           //die('Error en los datos: ' . mysql_error());
        } else {
           echo "insertado en la BDD <br>";
        }
        //echo "<br>";
      endforeach;
   else:
      echo ("No se ha encontrado ningÃºn tweet");
   endif;
   return 1;
}
//funcion que generara una lista dinamica para realizar
//la busqueda de palabras claves en Twitter
function lee_lista() {
   $fichero = fopen("C:\Tesis\Palabras.txt", "r");
   $numlinea = 1;
   while (!feof($fichero)) {
      $linea = fgets($fichero, 4096);
      $lista[] = $linea;
      //echo "linea numero: ".$numlinea." es: ". $linea."<br>";
      $numlinea++;
   }
   fclose($fichero);
   return $lista;
}
$Lista = lee_lista();
foreach ($Lista as $indice => $valor) {
   //codigo para referenciar una clase de twiiter y buscar en ella
   $twitteroauth = new TwitterOAuth(APP_CONSUMER_KEY, APP_CONSUMER_SECRET, ACCESS_TOKEN,
ACCESS_TOKEN_SECRET);
   $results = $twitteroauth->get('search/tweets', array(
      'q' => $valor, //$_GET['q'], // Nuestra consulta
      'lang' => 'es', // Lenguaje (espaÃ±ol)
      'count' => 200, // NÃºmero de tweets
      'include_entities' => false, // No nos interesa informaciÃ³n adicional. Ver: https://dev.twitter.com/docs/tweet-entities
   ));
   guarda_tweet($results, $valor);
   //echo "valor: ".$valor."<br />";
}
?>
```

Ejemplos teóricos de similitud entre tópicos de distintos periodos.

I. Si los tópicos T_l^i y T_h^j son los mismos, es decir los tópicos tienen las mismas palabras y estas están en la misma posición entonces.

$$Dist(T_l^i, T_h^j) = 0$$ (Cualesquier diferencia de posición es 0).

II. Si los tópicos T_l^i y T_h^j no tienen alguna palabra en común, las sumas de diferencias es desde 1 hasta N, la cual es $\frac{N*(N+1)}{2}$, entonces:

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[\frac{N(N+1)}{2}\right]$$

$$Dist(T_l^i, T_h^j) = 1$$

III. Si para toda $w_l^i n$ esta $w_h^j n$, pero se encuentra en una posición más cercana (o sea una diferencia de uno), como el siguiente ejemplo construido (se cree que puede haber más ejemplos).

$$T_l^i = \{w_l^i 1, w_l^i 2, w_l^i 3, w_l^i 4, \ldots, w_l^i N-1, w_l^i N,\}$$

$$T_h^i = \{w_l^i 2, w_l^i 1, w_l^i 4, w_l^i 3, \ldots, w_l^i N, w_l^i N-1,\}$$

Las palabras w_l^i están en posiciones cruzadas en w_j^i a una distancia de 1, se omite la aclaración de si N es impar, pero se toma el caso si es N par, entonces, aquí existen N diferencias de posición 1 y se tiene que:

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * N$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{(N+1)}\right]$$

En el caso de que m palabras coincidan en la misma posición, la distancia se acorta pues ahora, solo existen N-m diferencias de 1:

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * (N-m)$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{(N+1)}\right] * \left[\frac{N-m}{N}\right]$$

$$\left[\frac{2}{(N+1)}\right] * \left[\frac{N-m}{N}\right] < \left[\frac{2}{N+1}\right]$$

Por lo cual, los tópicos están más cerca (multiplicación de dos números menores que 1).

IV. Si para toda w_l^i esta w_l^i pero:

a) Todas las palabras se encuentran en una misma posición más lejana posible (o sea una diferencia de $\frac{N}{2}$) para todos.

b) O en otra posición en orden decreciente de $N-1, N-3, N-5, \ldots, 1$

Analicemos, primero el caso b), con el siguiente ejemplo construido (se cree que puede haber más ejemplos).

$$T_l^i = \{w_l^i 1, w_l^i 2, w_l^i 3, w_l^i 4, \ldots, w_l^i N-1, w_l^i N,\}$$

$$T_h^i = \{w_l^i N, w_l^i N-1, w_l^i N-2, w_l^i N-3, \ldots, w_l^i 2, w_l^i 1\}$$

La palabra w_l^i está en posición contraria en T_h^j , entonces, existen $\frac{N}{2}$ diferencias en el rango $1,3,5,\ldots,N-1$, pero dos de cada una y se tiene que su suma es $2*(N/2)^2$:

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[2 * \left(\frac{N}{2}\right)^2\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[\frac{N^2}{2}\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{N}{(N+1)}\right] < 1$$

Analicemos ahora el caso a), con el ejemplo construido (también, se cree que puede haber más ejemplos).

El primer caso posible de todos a una misma distancia posible de $\frac{N}{2}$ tiene la forma:

$$T_l^i = \{w_l^i 1, w_l^i 2, w_l^i 3, w_l^i 4, \ldots, w_l^i N-1, w_l^i N,\}$$

$$T_h^i = \{w_l^j (N/2)+1, w_l^j (N/2)+2, w_l^j (N/2)+3, \ldots, w_l^i N, w_l^i 1, w_l^i 2, w_l^i (N/2)\}$$

Aquí, existen N diferencias de $\frac{N}{2}$ y se tiene que su suma es $N\frac{N}{2}$; y por lo tanto la distancia es:

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[N * \frac{N}{2}\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{N}{(N+1)}\right] < 1$$

Observación 2. Para el caso a) y b) se obtienen la misma distancia.

En el caso de que m palabras coincidan en la misma posición, la distancia se acorta pues, ahora, existen solo N-m diferencias de $\frac{N}{2}$.

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[(N-m) * \frac{N}{2}\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{N-m}{N+1}\right] < 1$$

$$\left[\frac{N-m}{N+1}\right] < \left[\frac{N}{N+1}\right]$$

Por lo tanto, al tener ambos tópicos más palabras iguales y en la misma posición el tópico T_h^j se encuentra más cerca de T_l^i.

V. Analicemos los casos de iv) en que alguna w_l^i, ahora no está en w_l^i,

Primero nuevamente el caso b) y se obtenga la distancia máxima, que es cuando W tiene la posición N en T_l^i, la cual se debe de sumar en la parte de los valores absolutos, pero se debe de restar N-1 (lo que contribuía antes):

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[\left(2 * \left(\frac{N-1}{2}\right)^2\right) + (N-1)\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[\left(\frac{(N-1)*(N-1)}{2}\right) + (N-1)\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[(N-1) * \left(\frac{N-1}{2} + 1\right)\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[(N-1) * \left(\frac{(N-1)+2}{2}\right)\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[(N-1) * \left(\frac{(N+1}{2}\right)\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{N-1}{N}\right] < \left[\frac{N}{N+1}\right] < 1$$

Por lo cual T_h^j se acerca más T_l^i.

Analicemos ahora el caso a), en que alguna w_l^i, no se encuentre en w_l^i, aquí solo se reduce un $\frac{N}{2}$, con una palabra que no se encuentre, por lo que:

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \sum_{n=0}^{N} \left| pos(w_l^i n) - pos(w_h^j n) \right|$$

$$Dist(T_l^i, T_h^j) = \left[\frac{2}{N(N+1)}\right] * \left[N * \frac{N}{2}\right]$$

$$Dist(T_l^i, T_h^j) = \left[\frac{N}{(N+1)}\right] < 1$$

Sucediendo lo mismo que en el caso anterior y por lo cual T_h^j está más cerca de T_l^i.