ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE TÉCNICAS PARA MINERÍA DE TEXTO USANDO MODELOS DE TÓPICOS

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

CHRISTIAN EDUARDO CASTILLO MUÑOZ

christian.castillo@epn.edu.ec

DIRECTOR: MSC. MONTENEGRO ARMAS CARLOS ESTALESMIT

carlos.montenegro@epn.edu.ec

Quito, Febrero 2019

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Christian Eduardo Castillo Muñoz, bajo mi supervisión.

MSC. Carlos Montenegro

DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo Christian Eduardo Castillo Muñoz, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Christian Eduardo Castillo Muñoz

AGRADECIMIENTO

Grandes son las obras del SEÑOR, buscadas por todos los que se deleitan en ellas.

Sal 111,2

ÍNDICE DE CONTENIDO

CERTIFICACIÓN 1
DECLARACIÓN DE AUTORÍA 2
AGRADECIMIENTO 3
ÍNDICE DE CONTENIDO 4
RESUMEN 6
ABSTRACT 7
1. INTRODUCCIÓN 8
1.1 Objetivos 8
1.2 Alcance 9
1.3 Marco Teórico 9
1.3.1 Minería de Texto 9
1.3.2 Modelos Generativos 10
1.3.3 Procesamiento probabilístico del lenguaje natural 11
1.3.4 Modelo de Tópicos 12
1.3.5 Latent Dirichlet Allocation 13
1.3.6 Modelos de Tópicos basados en Latent Allocation 15
1.3.7 HMM-LDA Model 15
1.3.8 LDA-COL Model 16
1.3.9 Implementaciones de modelos LDA 17
1.3.10 Herramientas utilizadas en Minería de Texto 18
2. METODOLOGÍA 23
2.1 Preprocesamiento de Documentos 23
2.2 Extracción o selección de características 28
2.3 Selección del método 31
2.4 Entrenamiento y selección del clasificador 38
2.5 IMPLEMENTACIÓN 41
2.5.1 Inicialización de las librerías en R 42
2.5.2 Preprocesamiento de Texto 42
2.5.3 Reducción de términos 43
2.5.4 Selección de Parámetros para el Modelo 44
2.5.5 Modelo de Tópico LDA 45

2.5.6 Modelo de Tópicos en Matlab 45
2.5.7 Creación de archivos .mat:.....46
2.5.8 Transformación a datos de colocación 47
2.5.9 Selección de Parámetros y Obtención del Modelo 47
3. RESULTADOS Y DISCUSIÓN 49
3.1 Prueba 1: Selección del parámetro *k* (número de tópicos) 49
3.2 Prueba 2: Selección del parámetro Alpha 52
3.3 Resultado del Modelo LDA 55
4. CONCLUSIONES 70
5. REFERENCIAS BIBLIOGRÁFICAS.....71

RESUMEN

PALABRAS CLAVE: minería de texto, modelos de tópicos, bolsa de palabras, LDA.

La minería de texto permite la extracción de información relevante de grandes volúmenes de datos. Para realizar este proceso, la minería de texto explota técnicas y metodologías de las áreas de recuperación de información, extracción de información y lingüística computacional basada en corpus.

El principal insumo para la minería de texto son los documentos semi estructurados y no estructurados. Este insumo añade complicaciones al proceso de análisis, ya que se debe considerar las reglas del lenguaje natural.

Existen métodos de análisis probabilísticos para la extracción de características relevantes de documentos de texto, que en comparación con los métodos de análisis de tópicos que se basan en el análisis lingüístico, no se ven afectados por el contexto en el cual se encuentren las palabras, ya que estos modelos se basan en el criterio que los documentos son una “bolsa de palabras”.

Uno de los modelos del tipo mencionado es LDA(Latent Dirichlet Allocation), que asume que un documento es una mezcla aleatoria sobre temas latentes, donde cada tema se caracteriza por una distribución sobre las palabras; esta distribución de las palabras viene ligada a variables aleatorias. LDA pretende encontrar dichas variables para determinar los tópicos existentes en un conjunto de documentos.

Este trabajo muestra la implementación del modelo de tópicos usando LDA, aplicado a un caso de estudio, e integrando varias técnicas en diferentes entornos. Para el efecto, se utiliza las actividades recomendadas en las buenas prácticas para la recuperación de información.

ABSTRACT

KEYWORDS: text mining, topic model, bag of words, LDA.

Text mining allows the extraction of relevant information from large volumes of data. To carry out this process, text mining exploits techniques and methodologies in the areas of information retrieval, information extraction and corpus-based computational linguistics.

The main input for text mining is semi-structured and unstructured documents. This input adds complications to the analysis process, since the rules of natural language must be considered.

There are probabilistic methods of analysis for the extraction of relevant characteristics of text documents, which are not affected by the context in which the words are found, compared to the methods of topic analysis that are based on linguistic analysis. that these models are based on the criteria that documents are a "bag of words".

One of the models of the mentioned type is LDA (Latent Dirichlet Allocation), which assumes that a document is a random mix of latent themes, where each theme is characterized by a distribution over words; This distribution of words is linked to random variables. LDA intends to find these variables to determine the existing topics in a set of documents.

This paper shows the implementation of the topic model using LDA, applied to a case study, and integrating several techniques in different environments. For this purpose, the recommended activities are used in good practices for information recovering.

1. INTRODUCCIÓN

Existe gran cantidad de información digitalizada útil para la investigación, lo que hace que el proceso de seguimiento y búsqueda de datos sobre temas específicos se torne lento y difícil, por lo cual se recurre a la minería de texto, la misma que permite encontrar con mayor eficacia y rapidez. Para esto se busca la implementación de modelos probabilísticos que permitan la extracción de información relevante de grandes volúmenes de datos. A estos modelos se los denomina modelos de tópicos. Existen varios modelos que ayudan a este proceso; uno de ellos es Latent Dirichlet Allocation (LDA) caracterizado por ser un modelo generativo probabilístico para colecciones de tópicos ampliamente usado para detección, clasificación de similitud en documentos no estructurados con alta densidad de palabras [1].

Al analizar documentos no estructurados, estos se ven afectados por problemas de variación y ambigüedad lingüística [2]. Para tratar los problemas ligados al idioma, se aplican técnicas que permiten reducir el error de análisis en los mismos.

Este proyecto implementa modelos de tópicos apoyados en varias técnicas aplicables mediante herramientas informáticas para la minería de texto, procesamiento del lenguaje natural y los modelos de tópicos. Se debe tomar en cuenta que las herramientas actualmente disponibles, son específicas para ciertos casos, como es el caso de librerías de R, o son disímiles, como es el caso de MATLAB y R.

1.1 Objetivos

El objetivo general de este Proyecto Integrador es:

- Implementar un entorno de minería de texto usando modelo de tópicos LDA, apoyado en herramientas informáticas.

Los objetivos específicos de este Proyecto Integrador son:

- Levantar la información consultando al menos en 20 referencias sobre el Modelo de Tópicos y las mejores prácticas aplicables en la minería de texto.
- Seleccionar el grupo de herramientas para la implementación de Modelos de Tópicos, utilizando una estrategia de integración, que abarque las fases de

preprocesamiento de documentos, clasificación de contenidos y presentación de resultados.

- Desarrollar el caso de estudio donde se aplique las herramientas seleccionadas en el corpus de artículos utilizado en el artículo académico “Using probabilistic topic models to study orientation of sustainable supply chain research”.

1.2 Alcance

Se propone la implementación de un entorno de minería de texto aplicando modelo de tópicos; igualmente se propone realizar la selección de las herramientas informáticas que apoyen este proceso.

Se tomará como caso de estudio un conjunto de artículos científicos y publicaciones de diferentes fuentes de relevancia académica, para contribuir a la revisión y mejora de la publicación académica: “Using probabilistic topic models to study orientation of sustainable supply chain research” [3] de los autores Ing. Montenegro Carlos, Dr. Loza-Aguirre Edison, Dr. Segura Marco, profesores de la facultad de Ingeniería en Sistemas Informáticos y de Computación de la Escuela Politécnica Nacional.

1.3 Marco Teórico

1.3.1 Minería de Texto

La minería de texto es un proceso mediante el cual se extrae patrones interesantes y no triviales de documentos de texto, este proceso puede ser visto como una extensión de minería de datos [4]. En comparación con la minería de datos, la fuente de datos para la minería de texto son documentos no estructurados o semi estructurados.

Para extraer la información de manera eficiente, la minería de texto implica una serie de actividades a realizar: obtención de los documentos, preprocesamiento de los documentos, aplicación de diferentes técnicas para la obtención de patrones y la visualización de los resultados, ver Figura 1.1

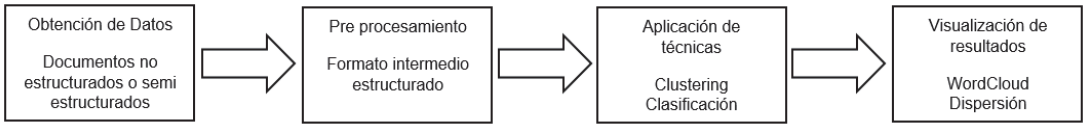


Figura 1.1 Actividades de la minería de texto

La minería de texto se centra en la identificación y extracción de características representativas en lenguaje natural, debido a que los documentos se encuentran en este lenguaje. La minería de texto se basa también en los avances alcanzados en otras disciplinas informáticas relacionadas con el manejo de lenguaje natural [5], tales como la recuperación de información, extracción de información y lingüística computacional basada en corpus, ver Figura 1.2.

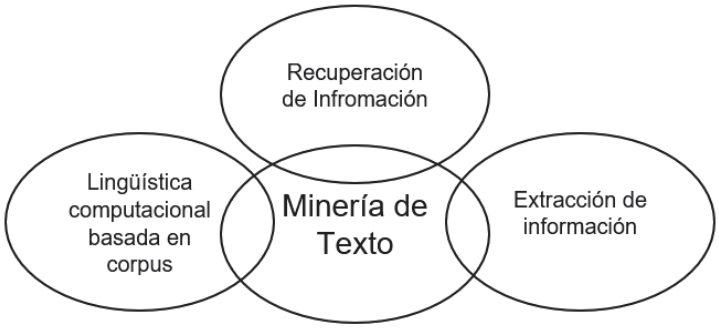


Figura 1.2 Áreas en las que se apoya la minería de texto [5].

1.3.2 Modelos Generativos

Un modelo generativo para documentos se basa en reglas de muestreo probabilísticas, las cuales describen como las palabras son generadas dentro de un documento basándose en variables aleatorias. El objetivo de seleccionar un modelo generativo es el encontrar el mejor conjunto de variables aleatorias que pueden explicar los datos observados, es decir palabras observadas en documentos, [6]. La Figura 1.3 ilustra el enfoque de modelado de temas basado en dos formas, como modelo generativo y como un problema de inferencia estadística

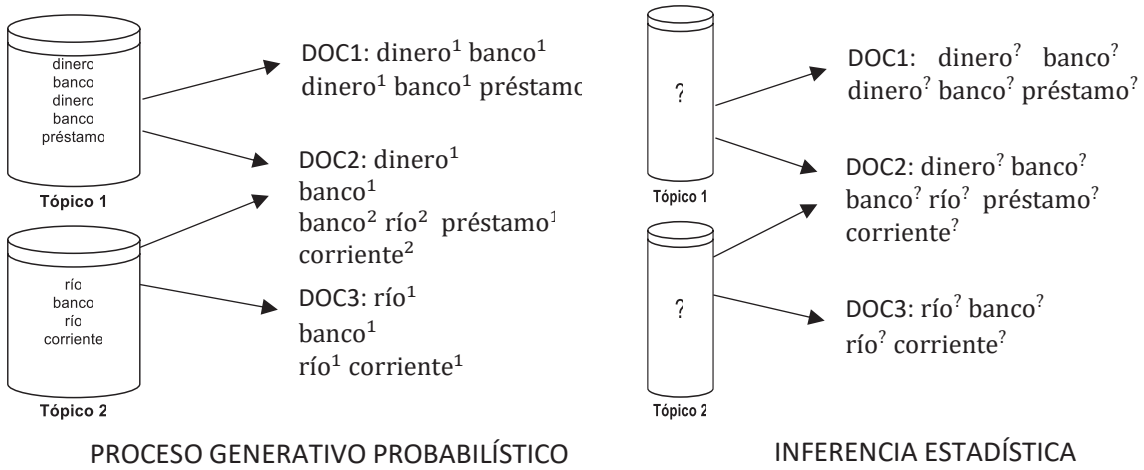


Figura 1.3. Ilustración del proceso generativo y el problema de la inferencia estadística subyacente a los modelos de temas [6].

A la izquierda, el proceso generativo es ilustrado con dos tópicos, los cuales están relacionados con dinero y ríos. Se ilustran como un conjunto de palabras que contiene diferentes distribuciones sobre las palabras. Así se denota que los Documentos 1 y 3 fueron generados de los temas 1 y 2 respectivamente, mientras que el documento 2 fue generado de una mezcla de los dos temas. Se puede evidenciar que los índices indican a que tema pertenecen las palabras, de esta forma el modelo define que no existe la noción de exclusividad mutua que restringe las palabras para ser parte de un tema solamente. Esto permite que los modelos de tópicos capturen polisemia, donde la misma palabra tiene diferentes significados; ejemplo de esto es la palabra banco, la cual puede estar tanto relacionado a temas de río como de dinero.

El lado derecho de la Figura 1.3 ilustra el problema de la inferencia estadística: dadas las palabras observadas en un conjunto de documentos, nos gustaría conocer qué modelo de tópicos es más probable que haya generado los datos. Para llegar a esta afirmación se necesita inferir la distribución de probabilidad sobre las palabras asociadas a cada tema, la distribución sobre los temas para cada documento y, a menudo, el tema responsable de generar cada palabra.

1.3.3 Procesamiento probabilístico del lenguaje natural

El procesamiento probabilístico del lenguaje natural [7] representa el modelo clásico de los sistemas de recuperación de información [2], ya que se tiene que cada documento se

encuentra definido como un conjunto de palabras claves a las cuales se las denomina términos índice.

Este enfoque se basa en “bag of words” (o bolsa de palabras), esto describe que cada documento está compuesto por términos índices, además de esto cada índice tiene un peso según su importancia (frecuencia de aparición en el documento), esta aproximación permite discriminar la estructura el significado y el orden de las palabras. Estos modelos son los más utilizados en la actualidad, esto se debe a su simplicidad y eficacia [2].

Este modelo se describe en dos etapas:

- **Preprocesador de los documentos:** Esta etapa permite limpiar los documentos, esto se consigue eliminando las palabras que no agregan significado a los documentos, estas palabras se las denomina stop words.
- **Parametrización:** Esta etapa es de dificultad mínima, ya que al tener identificado los términos relevantes se procede a la cuantificación de los términos claves según su frecuencia.

1.3.4 Modelo de Tópicos

Dentro del procesamiento del lenguaje natural y machine Learning los modelos de tópicos (*topics models),* son modelos probabilísticos, que se basan en dos suposiciones. La primera es que en una gran colección de documentos existen varios grupos o fuentes de texto diferentes, por ejemplo, si se poseen varios artículos científicos, los grupos pueden estar asociados con diferentes revistas, autores, subcampos de investigación o periodos de investigación. La segunda suposición es que los textos de diferentes fuentes tienden a usar un vocabulario diferente, es decir si se compara dos artículos de diferentes campos de investigación el vocabulario que estos utilizan difiere entre ellos [8].

El modelado de tópicos proporciona una forma conveniente de analizar textos grandes no clasificados (no estructurados). Dentro de los modelos de tópicos existen varios métodos, entre los cuales se nombra Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA) y Latent Dirichlet Allocation (LDA) [9].

Este trabajo está enfocado en la utilización del modelo de tópico LDA. El estudio realizado para la selección de este método se detallará en el Capítulo 2.

1.3.5 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) es un modelo probabilístico generativo empleado para encontrar contenido semántico en un corpus de texto. Este modelo toma a cada documento como una mezcla aleatoria, donde cada tema se caracteriza como una distribución probabilística sobre las palabras [10]

Además, LDA es un modelo jerárquico de tres niveles de los cuales cada nivel subyacente es un conjunto finito de su nivel inferior respectivo [1], así se definen los siguientes niveles:

Palabra: Es la unidad básica discreta de datos, se define una palabra como un elemento de un vocabulario $\{1, \dots V\}$.

Documento: es la secuencia de N palabras denotado como $w = (w_1, w_2, \dots w_N)$.

Corpus: la colección de M documentos denotado como $D = \{w_1, w_2, \dots w_M\}$.

Para ejemplificar el modelo generativo de tres niveles considérese que se tiene un corpus con 5 documentos.

Documento 1: Tuve un sándwich de mantequilla de maní para el desayuno.

Documento 2: Me gusta comer almendras, cacahuetes y nueces.

Documento 3: Mi vecino consiguió un pequeño perro ayer.

Documento 4: Los gatos y los perros son enemigos mortales.

Documento 5: No debes alimentar con cacahuetes a tu perro.

Los algoritmos de los modelos de tópicos analizan las palabras de una colección de documentos para descubrir los temas que en estos se trata, y como estos están conectados entre sí [11]. Así LDA descubrirá 2 temas, esto lo consigue según la distribución de las palabras dentro de cada documento:

Tópico 1: 30% maní, 15% almendras, 10% desayuno …

Tópico 2: 20% perro, 10% gato, 5% cacahuete …

A su vez se tiene que los temas se encuentran distribuidos de la siguiente manera en los documentos:

Documentos 1 y 2: 100% tópico 1

Documentos 3 y 4: 100% tópico 2

Documento 5: 70% Tópico 1, 30 % tópico 2

En la Figura 1.4 se ilustra el resultado del ejemplo planteado.

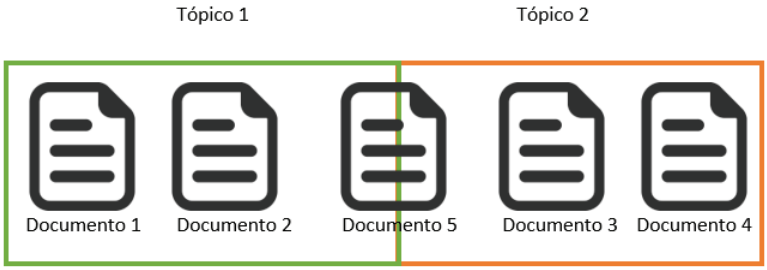


Figura 1.4 Representación gráfica de la distribución de los tópicos en cada documento

Lo temas recuperados seleccionan aspectos significativos de la estructura de los artículos y revelan las relaciones entre diferentes artículos de diferentes disciplinas. Los resultados del algoritmo tienen varias aplicaciones interesantes que pueden facilitar a las personas a comprender la información contenida en los grandes dominios de conocimiento, incluida la exploración de la dinámica de los temas y la función que desempeñan las palabras en el contenido semántico de los documentos. Al tratarse de un modelo basado en la bolsa de palabras (Bag Of Words *BOW*), la probabilidad de una secuencia de palabras no se ve afectada por el orden en el que aparecen. Los modelos de Tópicos usando LDA, se representa como se muestra en la Figura 1.5

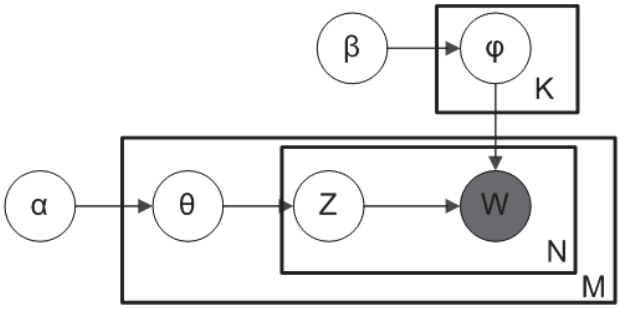


Figura 1.5. Modelo gráfico de LDA (PLate Notation) [1]

Parámetros del modelo:

- Cajas:
    - K: es el número de tópicos
    - N: El número de palabras en el documento

- M: Número de documentos a ser analizados

- β: Es el mismo parámetro de la distribución de palabras por tema
- α: Parámetro de concentración anterior Dirichlet de la distribución por documento
- Φ(T): Es la distribución de palabras por tópico (T)
- θ(i): Es la distribución de tópicos por documento(i)
- w(i,j): es la j(esima) palabra en el i(esimo) documento
- z(i,j): Es la asignación de tema para w (i,j)

1.3.6 Modelos de Tópicos basados en Latent Allocation

A continuación, se presentan varios modelos gráficos y la descripción de algunos tipos de modelos basados en LDA:

1.3.7 HMM-LDA Model

Los enfoques estadísticos para el aprendizaje de idiomas generalmente se enfocan en dependencias sintácticas de corto alcance o dependencias semánticas de largo alcance entre palabras. El modelo HMM-LDA usa ambos tipos de dependencias, y es capaz de encontrar simultáneamente clases sintácticas y temas semánticos a pesar de no tener conocimiento de sintaxis o semántica más allá de la dependencia estadística. Según Griffiths, Blei, & Tenenbaum [12], este modelo es competitivo en tareas como etiquetado de voz parcial y clasificación de documentos con modelos que usan exclusivamente dependencias de corto y largo alcance, respectivamente. HMM-LDA Model se define con el modelo grafico presentado en la Figura 1.6.

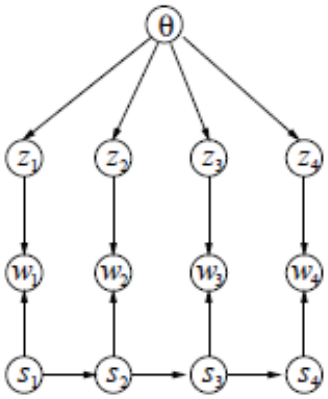


Figura 1.6. Modelo gráfico de HMM-LDA *[12]*

1.3.8 LDA-COL Model

LDA-COL (LDA - Collocation) o modelo de Colocación, es parte del modelo de tópicos LDA, que introduce un nuevo conjunto de variables aleatorias, además de los dos conjuntos de variables aleatorias z y w en LDDA, estas nuevas variables permiten determinar el estado del bigrama, esto denota si el bigrama puede ser formado por el token anterior adicional [13]. Por lo tanto, se puede definir con mayor certeza si se genera un bigrama o un unigrama. El modelo LDA-COL, asume una representación gráfica que combina el modelo de tópicos de Bigram con LDA, esto se representa en la Figura 1.7.

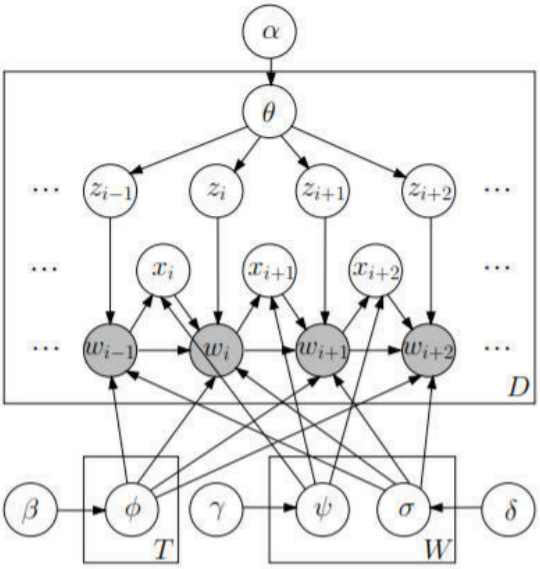


Figura 1.7. Modelo Grafico de LDA-Col *[13]*

La figura 1.8 muestra como LDA-Col conjuga LDA y el modelo de n-gramas, este último pretende predecir x_i basado en $w_{i-(n-1)}, \ldots, w_{i-1}$, en términos de probabilidades se tiene que $P(x_i | x_{i-(n-1),\ldots,} x_{i-1})$, si se asume el modelado de idiomas, se hace la suposición de independencia de esta forma cada palabra depende de las ultimas n - 1 palabras [13].

De esta forma el tópico queda definido como un conjunto de unigramas o n-gramas, esto permite agrupar palabras que pudiesen ser desconocidas para el modelo.

1.3.9 Implementaciones de modelos LDA

A continuación, se muestran dos casos de estudio que implementaron modelos de tópicos basados en LDA:

Topic and Role Discovery in Social Networks [14]**:** Análisis de Redes Sociales (*Social Network Analysis,* SNA) es el estudio de modelos matemáticos para las interacciones entre personas, organizaciones y grupos [9]. SNA se basa en el análisis de la relación entre diferentes entidades, pero no se centra en el contenido del idioma o de los tópicos de dichas relaciones. McCallum, Corradda-Emanuel y Wang propusieron el estudio de dicho contenido de los enlaces entre diferentes entidades. En este estudio se introdujo un modelo denominado *Author Recipient-Topic (ART)*, este se basa en el análisis de las redes sociales, este modelo se construyó en base a LDA y en el modelo *Author-Topic(AT)* el cual permite enlazar cada tópico tratado a un emisor y a un receptor, lo cual da como resultado el descubrimiento de los temas de acuerdo con las relaciones entre las personas.

Una aplicación exitosa de SAN es en la detección de redes criminales y terroristas y de las entidades más influyentes dentro de dicha red, esto se determina a partir de los rastros de comunicación que recopilan entidades de seguridad nacional.

Anti-Phishin: Los mensajes de Phishing se han convertido en una manera efectiva de obtener información personal en el ciberespacio, usando mensajes de correo electrónico, para posteriormente dar mal uso a esta información, por lo que el filtrado de correos electrónicos o de páginas web, no son las formas efectivas de contrarrestar estos ataques. Ramanathan y Wechsler [15] usaron modelos de tópicos, asignación LDA para extraer características y la técnica de clasificación AdaBosst para la implementación de un clasificador de páginas web, el cual permitió desarrollar un método de detección de phishing, logrando que sea un modelo robusto frente a cambios tecnológicos, cambios de palabra, neutralidad en cuando al idioma, además de tener amplio campo de aplicación ya que es aplicado en clientes móviles y de escritorio.

1.3.10 Herramientas utilizadas en Minería de Texto

Dentro de los leguajes de programación en los que se apoya la minería de texto tenemos a R y Matlab, los cuales cuentan con librerías y paquetes que apoyan los diferentes procesos de la minería de texto y son aplicadas en los modelos de tópicos.

R

R tiene una naturaleza doble ya que se define como un sistema para análisis estadísticos y gráficos, además de ser considerado como lenguaje de programación. Como lenguaje de programación R es un lenguaje interpretado y no compilado, es decir que, sin necesidad de construir ejecutables, los comandos ingresados son ejecutados directamente. R es una multiplataforma ya que corre independientemente del sistema operativo que se utilice [16].

R a diferencia de lenguajes de programación clásicos, este almacena los resultados del análisis para poder ser manipulados posteriormente en su totalidad o parcialmente. En la Figura 1.8 se esquematiza el funcionamiento de R.

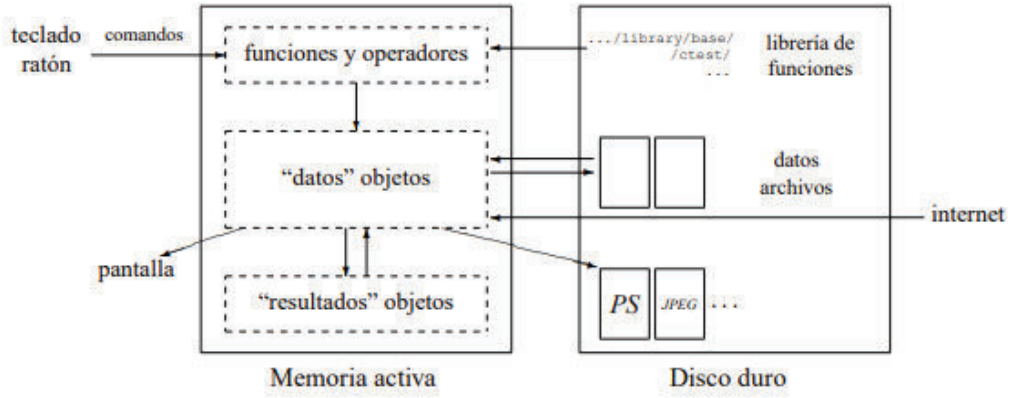


Figura 1.8. Visión esquemática del funcionamiento de R [19].

R cuenta con un conjunto extenso de librerías aplicables a diferentes campos del procesamiento de datos. Estas librerías pueden ser obtenidas desde el sitio de internet *Comprehensive R Archive Network (CRAN),* además R da la posibilidad de agregar paquetes de terceros, ya sea desde un archivo .zip o de archivos binarios [19].

R cuenta con varias librerías específicas para la minería de texto, a continuación, se describe las librerías aplicadas en este proyecto.

Text Mining Package (tm):

Es un framework para aplicaciones de minería de texto, esta librería cuenta con funciones que ayudan a la creación corpus, la manipulación de dichas estructuras, adicional permite asociar los documentos con los términos pertenecientes en dicho corpus, es decir la creación de la matriz de termino-documento.

SnowballC:

Es una interfaz de R para la biblioteca C libstemmer la cual implementa el algoritmo de derivación de palabras de Porter, esta función permite la derivación de las palabras a su raíz, para hacer la comparación de vocabularios.

Rstem Package:

Este paquete proporciona a R la posibilidad de hacer uso de funciones de derivación de palabras proporcionadas por Martin Porter´s Snowball.

Ldatuning:

Este paquete apoya al análisis necesario para estimar el número de iteraciones y el total de tópicos con los cuales se va a correr el modelo de tópicos. Para lo cual este paquete realiza 4 métricas ("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"), estas métricas ayudan al desempeño de la selección del número de tópicos.

Stringr:

Es una librería que permite la manipulación de cadenas de texto, esta librería es ampliamente utilizada en la limpieza de texto.

Wordcloud:

La librería worcloud, permite generar plots, de wordclous o tag clouds, este método de visualización de palabras es utilizado para tener una visión de la relevancia de ciertos términos sobre otros.

Topicmodels:

Proporciona una interfaz para el código c, para modelos de asignación de LDA, adicional es una interfaz para el código C++ que permite ajustar modelos LDA utilizando el muestreo de Gibbs.

R.matlab:

Interfaz que permite la comunicación entre MATLAB y R, para lo cual se cuenta con métodos de readMat() y writeMat(), esta interfaz es compatible con MATLAB v6 o superiores.

Easypackages:

Es un paquete utilitario, el cual proporciona la facilidad de instalar múltiples paquetes, esto se hace para facilitar la importación y activación de los diversos paquetes usados en el proyecto.

Parallel:

Este paquete está disponible desde la versión de R 2.14.0, esta librería permite el paralelismo a grano grueso, el paralelismo de grano grueso hace referencia a la frecuencia con la que subtareas se comunican entre sí, se considera paralelismo de grano grueso si no se comunican muchas veces por segundo.

Reshape2:

Esta librería permite la reagrupación y agregación de nuevos datos a estructuras preexistes, como de nuevas estructuras, esto lo lleva a cabo usando dos funciones dcast y acast. Esta librería permite la agrupación de los diferentes parámetros que se utilizan en ggplo2 para presentar los resultados de manera gráfica.

Ggplot2:

Es un sistema que permite la creación de gráficos de forma declarativa, basado en gramática de grafos. Para esto se necesita proporcionar los datos y ggplot2 asocia las variables a la estética del gráfico.

Matlab:

MATLAB es un lenguaje de programación, utilizado por disciplinas que necesitan hacer cálculos matemáticos. El éxito de Matlab radica en que es un sistema interactivo en el cual su elemento básico de datos es el arreglo, los cuales no necesitan un dimensionamiento previo.

MATLAB integra:

- Análisis numérico
- Cálculo matricial
- Procesamiento de señales
- Gráficos

Se debe considerar que MATLAB puede integrarse con un conjunto de librerías y herramientas a las cuales se denomina “toolbox”. Estas herramientas permiten la integración de entornos específicos para diferentes propósitos es decir que la arquitectura de Matlab se ve influenciada de los diferentes “tollbox”, ver Figura 1.9.

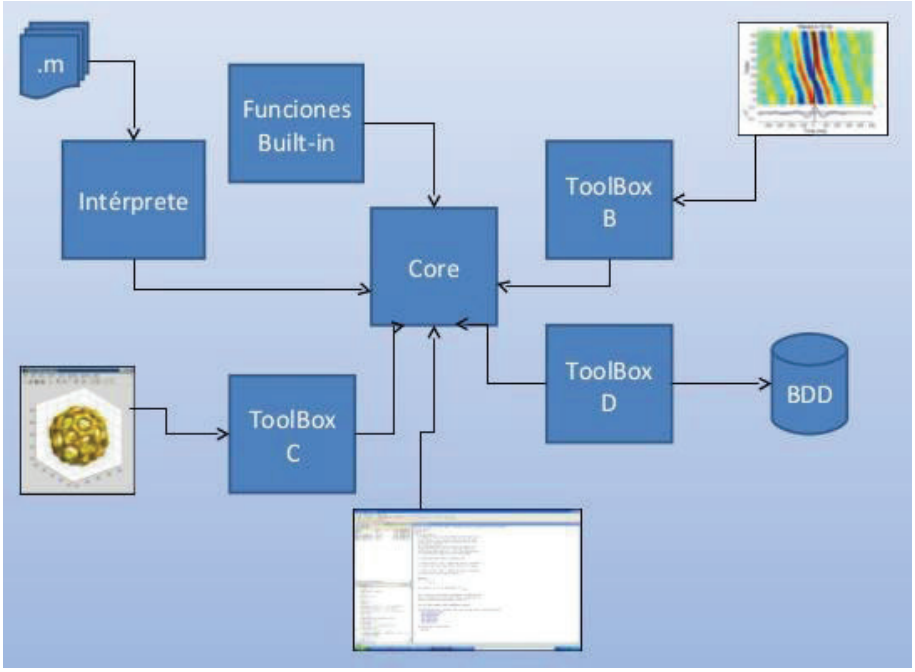


Figura 1.9. Arquitectura de Matlab

Matlab Topic Modeling Toolbox 1.4

Es un conjunto de funciones y de librerías que permiten la implementación de modelos de tópicos. Topic Modeling trabaja con compiladores de 64 bits. Es importante conocer que, para hacer uso de los modelos de tópicos dentro de Matlab, se deben construir variables, las cuales tiene un formato específico dependiendo del modelo de tópicos implementado.

A continuación, se especifica los requerimientos de datos utilizados por los modelos aplicados en este proyecto.

WO: Es un array de strings de [1, W] donde WO{k} contiene el k-ésimo elemento del vocabulario y w es la cantidad de términos. Este array no es necesario para la generación del modelo usando Gibbs, no obstante, es utilizado para presentar los resultados.

DS: es un vector de [1, N] donde DS(k) contiene el índice del documento del token de la k-ésima palabra, los índices de los documentos empiezan en 1 y terminan el número total de documentos.

WS: es un vector de [1,N] donde WS(k)contiene el índice de vocabulario del k-ésimo token de palabra, y N es el número de tokens palabra, al igual que DS el mínimo índice de WS es 1 y el máximo índice de WS es el número de palabras distintas en el vocabulario.

2. METODOLOGÍA

Este trabajo se apoya en la investigación básica para la extracción de información relevante, y su posterior implementación se lleva a cabo mediante la investigación aplicada [17].

La información producto de la investigación pura o básica es ampliamente aplicada a los campos de procesamiento del lenguaje natural y a modelos probabilísticos de minería de texto, utilizando la investigación aplicada, los avances de dichas disciplinas ayudan a la implementación de modelos de tópicos aplicables al análisis y evaluación de corpus.

La información obtenida para el desarrollo del proyecto se obtiene de libros, documentos escritos, revistas científicas y sitios web de relevancia académica. Las técnicas de recolección de información utilizadas son:

- Análisis de documentos: Se utiliza para el desarrollo del marco teórico, para esto se analizan publicaciones y libros.
- Sitios Web: Existe investigaciones y varias aplicaciones de modelos de tópicos usando LDA en diversos sitios web como ACM, IEEE y blogs de profesores universitarios.

Mucha de la literatura sobre minería de texto, trata a los modelos de tópicos como sistemas de recuperación y tratamiento de información. Para la implantación de dichos sistemas se tiene una estrategia genérica que según Zaveri y Dalal [18] consta de cuatro etapas, las cuales se describen a continuación:

2.1 Preprocesamiento de Documentos

El primer paso en el proceso de data mining es el preprocesamiento de documentos, en esta etapa se reduce las ambigüedades del lenguaje natural para obtener conjuntos de palabras manejables por sistemas de clasificación de datos. Según Vijayarani, Ilamathi y Nithya [19], existen dos tareas claves que definen el preprocesamiento de documentos los cuales son: la remoción de stop words y stemming (derivación). Las técnicas de preprocesamiento de data mining se definen en la Figura 2.1.

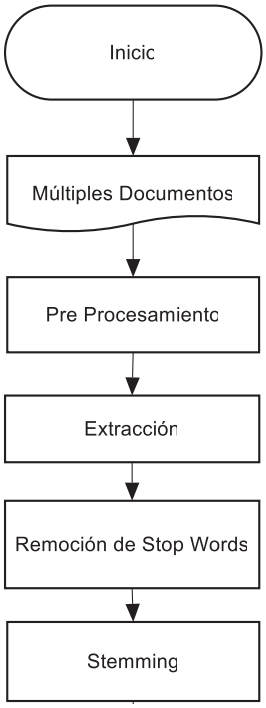


Figura 2.1. Técnicas de preprocesamiento de minería de texto

A continuación, se detalla las diferentes técnicas aplicadas en los métodos de preprocesamiento de documentos.

Extracción:

Este método se usa para convertir el contenido de un archivo en una cadena de palabras (tokenize) [19]. Esta idea se ejemplifica a continuación:

Se tiene la oración

A: Los gatos y los perros son enemigos mortales.

La extracción transforma cada una de las palabras en un token, es decir:

A: [‘Los’, ‘gatos’, ‘y’, ‘los’, ‘perros’, ‘son’, ‘enemigos’, ‘mortales’]

El objetivo de transformar la oración en un vector de palabras es para facilitar el proceso de eliminación o reducción de palabras.

Remoción de Stop Words:

Las Stop Words son palabras propias del lenguaje natural que generan ambigüedad en los documentos, por consiguiente, estas palabras afectan en el proceso de análisis de los documentos ya que hacen que el resultado de dicho análisis se vea afectado por ruido

(palabras que no agregan significado al análisis realizado) y que a su vez dicho análisis sea complejo de realizar.

Por lo general se considera que las Stop Words son un conjunto único de palabas las cuales varían dependiendo del contexto y del estudio que se desee hacer. Por ejemplo, si se desea hacer análisis de sentimientos la remoción de palabras como “bueno” y “no” pueden generar un análisis de sentimientos erróneo.

Entre el conjunto de stop words se tiene a los artículos, preposiciones, pronombres, adverbios.

Los modelos probabilísticos toman las palabras relevantes para los diferentes tópicos, por esta razón el conjunto de stop words a considerarles son los artículos, preposiciones, pronombres y adverbios. La remoción de estas palabras no afecta al modelo que se va a implementar. Porter [20] habla sobre 4 métodos que se pueden seguir para la remoción de Stop Words, de los cuales se tomará el método clásico, ya que este permite la adición de palabras que pudiesen afectar al estudio realizado.

Método Clásico para la remoción de Stop Words:

Este método se basa en una lista pre compilada de palabras [21],estas palabras son buscadas en los documentos y si coinciden son removidas, la Figura 2.2 ilustra este proceso.

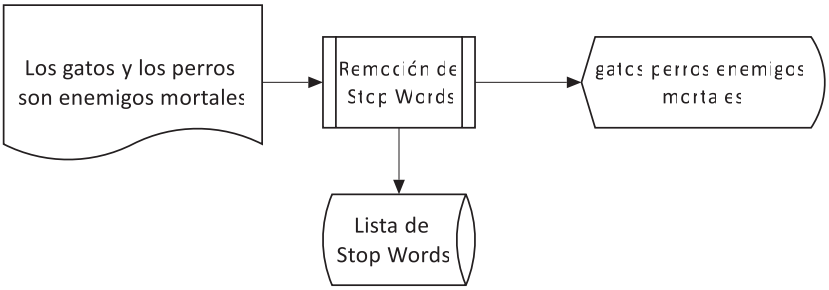


Figura 2.2. Remoción de Stop Words

Se tiene de la siguiente oración:

```
[1] "Even the way wineries seem to be using connections with the environment and the land itself as a
 differential advantage seems to be part of this willingness to experiment"
```

Las palabras señaladas de azul son stop words que deben ser extraídas del texto.

El proceso de remoción de stop words da el siguiente resultado:

```
[1] "Even  way wineries seem   using connections   environment   land    differential advantage seems
   part   willingness  experiment"
```

Adicional a la remoción de stop word existen dos métodos los cuales son Stemming y Lemmatización, estos métodos reducen las palabras a su morfema o tema es decir su forma básica.

- **Stemming**

Stemming o derivación es una técnica de preprocesamiento que permite eliminar la variedad morfológica de las palabras e identificar el tema raíz que las define.

Para entender cómo funciona stemming, es importante establecer las siguientes definiciones:

Morfema: Mínimo fragmento de una palabra que expresa un significado.

Stem (tema): Palabra raíz, a la cual se le agregan **afijos.**

Affixes (Afijos): Son los prefijos, sufijos e interfijos que se añaden a un tema.

La palabra “Playin” se descompone en:

$$\underbrace{\text{play}}_{\text{Tema}} \quad \underbrace{\text{ing}}_{\text{Afijo}}$$

El algoritmo más común de stemming es el de Porter, el cual es un algoritmo de truncado que consiste en 5 fases de reducciones de palabras aplicadas secuencialmente. La reducción se la realiza seleccionando la regla de cada grupo de reglas que se apliquen al sufijo más largo. La primera fase, la convención se usa con las siguientes reglas:

Regla			**Ejemplo**		
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
SS	→	SS	caress	→	caress
S	→		cats	→	cat

Las reglas consiguientes usan la medida de una palabra dicha medida es el número de sílabas, para ver si una palabra es lo suficientemente larga para poder considerar la

porción correspondiente de una regla como un sufijo en lugar de ser una palabra completa.

Las palabras resaltadas de azul sufren variedad morfológica.

```
[1] "The findings show managers within New Zealand wine supply chains trying find ways leverage sustainability
-related competencies competitive advantage now highly competitive industry"
```

Luego de aplicar el algoritmo de Porter se tiene lo siguiente:

```
[1] "the find show manag within new zealand wine suppli chain try find wai leverag sustain relat compet competit
advantag now highli competit industri"
```

Como se evidencia las palabras finfings, managers suplly, chains trying, se redujeron a find, manag, supllim, chain y try

- **Lemmatización**

La Lemmatización es un proceso que agrupa una o varias palabras en una sola, dicho proceso es similar a Stemming ya que mapea varias palabras en una raíz común, por ejemplo, en el lenguaje ingles el verbo “walk” muchas veces aparece como “walk”, ”walked”,”walks”,”walking”, la forma básica del verbo es “walk” esta última es lo que se conoce como lema de una palabra.

El siguiente texto presenta un conjunto de palabras que se pueden ser agrupadas en su lema.

```
[1] " emergent theme searching advantage sustainability involves: pursuing leveraging sustainability; telling
 story involves sustainability; managing supply chain relationships around sustainability; experimenting sust
ainability initiatives."
```

Luego de realizar el proceso de Lemmatización se tiene que las palabras emergent, searching, advantage sustainability entre otras cambiaron al lema: emerg, search, advanteg, sustain.

```
[1] "emerg theme search advantag sustain involv pursu leverag sustain tell stori involv sustain manag suppli chain
 relationship around sustain experi sustain initi"
```

Adicional a la lemmatizacion y al stemming para la limpieza de texto se añaden dos procesos más, los cuales son la remoción de caracteres especiales (signos de puntuación y números), así como también la remoción del contenido de texto que se encuentre dentro de símbolos de agrupación como paréntesis, llaves y corchetes

La Figura 2.3 muestra un fragmento de un documento afectado por la ambigüedad lingüística, adicional se tiene el resultado del preprocesamiento del documento.

```
ustainability in building competitive advantage within the supply chain. Quantitative research, e.g. surveys,
quasi-experiments, or content analysis of secondary data, could then be used to further develop and test theor
ies. Future research may also examine other competitive industries to determine how they are attempting to lev
erage sustainability in their supply chains for a competitive edge.\nWe would like to conclude with an implica
tion for supply chain managers, and that is although sustainability initiatives such as organic, bio-dynamic,
and low carbon emissions can all be used to differentiate companies and supply chains in some way, marketing p
urposes need not be, nor should they be, the sole reason for pursuing them. For some, being organic is merely
a way of doing business. Curt described a well-known winery in the region for example:\nOrganic is how they do
 things, it's not a marketing thing for some wineries; it is what they stand for; it is integral to who they a
re; (Curt's winery) is not at that level ... (Curt).\nHowever, it is beneficial if you can also grow a busines
s and achieve an advantage by doing what is sustainable as well as integral to who you are \n\016Notes\n1. For
 this research, we are limiting our definition of sustainability to only the environmental dimension.\n2. Fran
ce, Italy, Germany and Spain.\n3. Please note that we have disguised all names of participants and their compa
nies in the findings and discussion by using pseudonyms. While we used company web sites and marketing literat
ure as data sources, we do not include these in the reference list for con fidentiality purposes.

                                                                                    sustainability searching competitive adv
antages further validation models emerge accomplished future research draw aggregate data originality value ap
proach context within sustainability explored unique seeking deep insights managers cutting edge sustainabilit
y initiatives close strategic thinking explore impact distribution relationships keywords economic sustainabil
ity competitive advantage supply chain management wines new zealand paper type research paper introduction dis
tribution networks serve means competitive advantage beneficial grow business achieve advantage sustainable we
ll integral notes research limiting definition sustainability environmental dimension france italy germany spa
in please note disguised names participants companies findings discussion using pseudonyms used company web si
tes marketing literature data sources include reference list con fidentiality purposes
```

Figura 2.3. Comparación entre texto sin preprocesamiento y texto limpio

Como se evidencia en la figura 2.3, el preprocesamiento de documentos da como resultado un texto simplificado, y sin palabras y caracteres que afecten al modelo probabilístico.

2.2 Extracción o selección de características

La extracción de características permite determinar la frecuencia con la cual se presentan los términos dentro de los documentos, para determinar la relación que existe entre documentos y a su vez de los diferentes tópicos que estos pudiesen tratar. Para determinar esto se tiene la Matriz de Documento Termino (DTM)

- **Matriz de Documento Termino (DTM Document-Term Matrix)**

La matriz de Documento Termino (DTM) es una matriz matemática que describe la frecuencia de los términos en un corpus, en esta matriz las filas corresponden a los documentos, las columnas corresponden a los términos y cada valor representa el número de apariciones de un término en dicho documento.

Para ejemplificar la extracción o selección de características tomemos una porción del corpus de lo cual se puede obtener lo siguiente:

```
docs = {
    'doc1': ['industri','chain', 'company', 'environment',
'industri','chain', 'firm', 'industri','chain', 'model', 'industri',
'chain', 'chain', 'industri', 'company', logist', …],
```

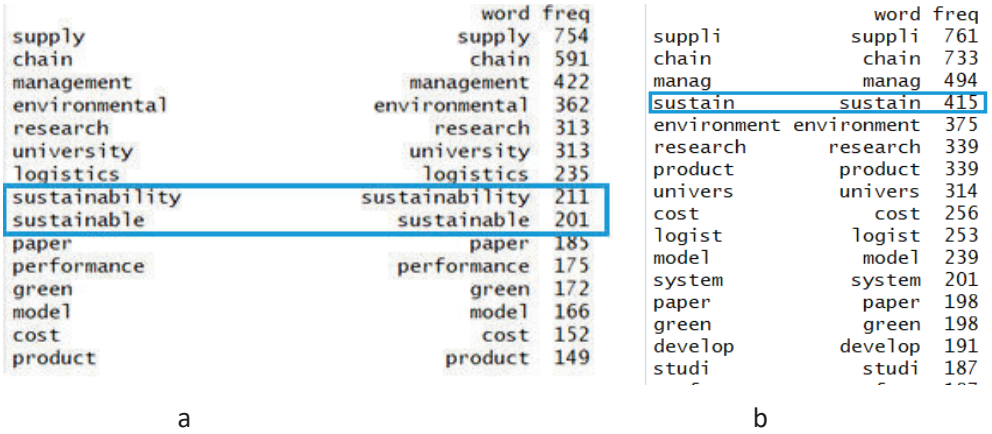
```
    'doc11': ['chain', 'environment', 'industri', logist', 'industri',
'chain', 'chain', 'firm', 'industri', …],
    'doc17': ['logist', 'company', 'chain', 'environment', 'chain',
'chain', 'chain', 'chain', logist', 'chain'…],

    'doc49': ['chain', logist','company', 'algorithms', 'chain',
'company', 'environment', 'industri', …]  }
```

De lo cual se obtiene como resultado la siguiente matriz de documento termino:

Docs	chain	compani	environment	firm	industri	logist
1	5	2	1	1	5	1
11	19	0	1	0	3	1
17	16	1	10	0	0	12
49	2	14	3	0	4	1

Se debe considerar que palabras como company e industri fueron reducidas a su morfema, esto se debe al paso previo de limpieza de texto.

Si no se realizara la reducción a morfemas se tendría algo como se muestra en la Figura 2.4(a), las palabras sustainability y sustainable son palabras diferentes con diferente frecuencia, estas palabras pueden ser consideradas una sola haciendo uso de la reducción a su morfema como se evidencia en la Figura 2.4(b)



	word	freq
supply	supply	754
chain	chain	591
management	management	422
environmental	environmental	362
research	research	313
university	university	313
logistics	logistics	235
sustainability	sustainability	211
sustainable	sustainable	201
paper	paper	185
performance	performance	175
green	green	172
model	model	166
cost	cost	152
product	product	149

a

	word	freq
suppli	suppli	761
chain	chain	733
manag	manag	494
sustain	sustain	415
environment	environment	375
research	research	339
product	product	339
univers	univers	314
cost	cost	256
logist	logist	253
model	model	239
system	system	201
paper	paper	198
green	green	198
develop	develop	191
studi	studi	187

b

Figura 2.4. Palabras con mayor frecuencia en una Matriz Documento Termino

Para determinar el número de veces que dicha palabra aparece en el corpus solo bastaría con sumar las columnas de la matriz.

Para analizar la incidencia de palabras dentro del corpus se puede hacer uso de la matriz de Termino por Documento, con lo cual se puede estimar un top de las palabras más utilizadas en el corpus ver la figura 2.5.

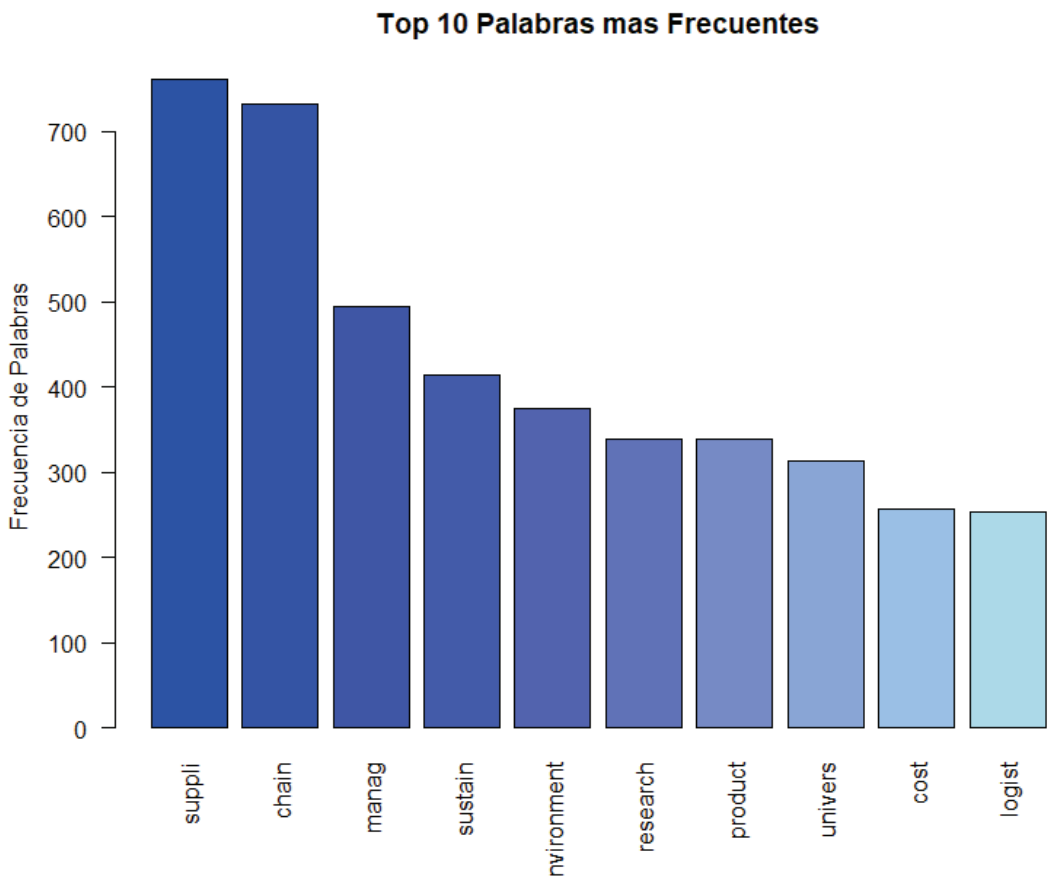


Figura 2.5. Gráfico de barras Top 10 de palabras presentes en el corpus

- **Word cloud**

Dentro de las diferentes alternativas de visualización de texto existen las nubes de palabras (word cloud) también conocidas como nube de etiquetas (tag cloud), esta técnica es ampliamente usada para visualizar las palabras claves de una matriz de Documento Termino, dichas palabras destacan mejor que en una tabla o en un histograma, la Figura 2.6 muestra la nube de palabras del corpus de estudio, se tiene que las palabras con mayor frecuencia se encuentran posicionadas en el centro de la nube y según la posición en el top de frecuencia esta tiene mayor tamaño y un color más llamativo, adicional las palabras menos significativas se encuentran de un solo color y tamaño.

Figura 2.6. Nube de palabras del corpus de estudio

La matriz de documento termino nos ayuda a identificar la relación entre los términos más frecuentes y las de más palabras del corpus, por ejemplo, la palabra suppli se relaciona con las siguientes palabras.

```
$suppli
  chain keyword   paper  econom    need
   0.95    0.45    0.45    0.44    0.42
```

2.3 Selección del método

Los modelos de tópicos hacen la distinción de diferentes métodos, los cuales se basan en que la semántica de cada documento se construye basándose en variables “latentes” las cuales no se observan. Los modelos de tópicos buscan estas variables para poder identificar los diferentes tópicos que describen un documento. A continuación, se discuten los diferentes métodos con los cuales se cuenta en modelos de tópicos.

LSA (Latent Semantic Analysis/Análisis Semántico Latente)

Esta técnica toma una matriz de documentos y términos y la descompone en una matriz separada de documentos y tema y otra matriz de tema y término.

Para esto el primer paso es generar una matriz de documento-termino (Document-Term-Matrix DTM). En la versión más simple de LSA, cada entrada puede ser un recuento de la cantidad de veces que apareció la j-ésima palabra en el i-ésimo documento, sin embargo, en la practica el recuento no funciona bien porque no tiene en cuenta el significado de cada palabra en el documento. Por ejemplo, la palabra “nuclear” probablemente informe más sobre el tema x de un determinado documento antes de la palabra “prueba”.

En consecuencia, LSA reemplaza el conteo de las filas en DTM con tf-idf (Term frecuency – inverse docuent frecuency), el cual asigna peso al termino j en el documento i siguiendo la ecuación 2.1:

$$W_{i,j} = tf_{i,j}\ x\ log\frac{N}{df_j}$$

Ecuación 2.1. Modelo de LSA

Donde:

$tf_{i,j}$ = Ocurrencia de termino en el documento

N = Total de documentos

df_j = Documentos que contienen la palabra

Con esto se puede asumir que cuando más frecuente es el termino en el documento, mayor es su peso, y cuando más infrecuente aparece en el corpus, mayor es su peso.

Luego de tener la Matriz de Documentos y Términos, se puede empezar la búsqueda de los temas latentes, sin embargo, existe un problema, la matriz es muy escasa, ruidosa y redundante en todas sus dimensiones, por consiguiente, para encontrar los tópicos latentes relevantes que describan las relaciones entre las palabras y los documentos, es necesario realizar una reducción de dimensiones de la matriz.

Para esta reducción se puede implementar el truncado SVD (Singular Value Decomposition / Descomposición del valor singular), la cual es una técnica que factoriza cualquier matriz M en el producto de 3 matrices separadas, como lo denota la ecuación 2.2.:

$$M = U * S * V$$

Ecuación 2.2. Truncado SVD

Donde:

S: matriz diagonal de los valores singulares de M

U y V: t mayores valores singulares (t: híper termino que permite seleccionar y ajustar a los tópicos que se deseen encontrar), la ecuación que describe el ajuste del híper termino se muestra en la ecuación 2.3

$$M \approx U_t S_t {V_t}^T$$

Ecuación 2.3. Reducción de términos SVD

El truncado SVD se ejemplifica en la Figura 2.7.

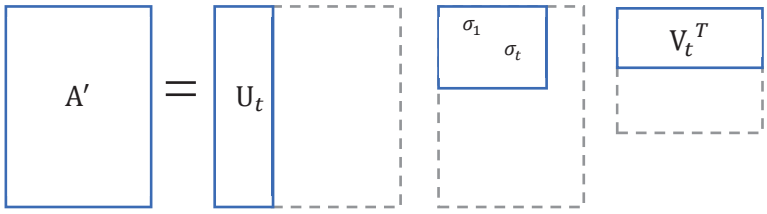


Figura 2.7. Representación de SVD

En este caso, $U \in \mathbb{R} \wedge (m \times t)$ emerge como la matriz de documento tema, y $V \in \mathbb{R} \wedge (n \times t)$ se convierte en una matriz de termino tema. En U, las filas representan vectores de documentos expresados en términos de temas; en V las filas representan vectores de términos expresados en términos de temas [22].

pLSA (Probabilistic Latent Semantic Analysis/Análisis Semántico Probabilistico Latente)

Esta técnica a diferencia de LSA usa un método probabilístico en lugar de SVD para afrontar el problema de reducción de términos a los más significativos. La idea central de pLSA es encontrar un modelo probabilístico con tópicos latentes que pueden generar los datos que se presentan en la matriz de documento termino.

La representación gráfica del modelo PLSA se muestra en la Figura 2.8

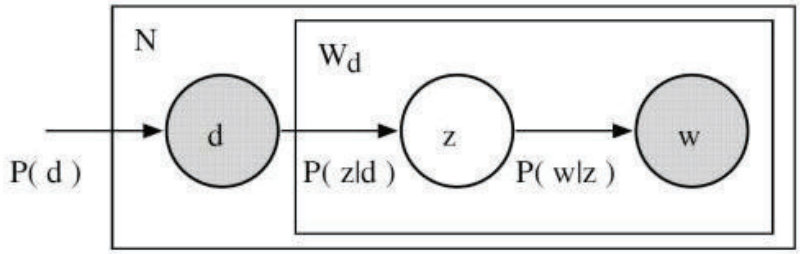


Figura 2.8. Representación gráfica de PLSA

De lo cual se define que la probabilidad conjunta de ver un documento dado y una palabra se expresa en la ecuación 2.4.

$$\mathrm{P}\ (\mathrm{D}|\mathrm{W}) = \mathrm{P}\ (\mathrm{D}) \sum_{z} P(Z|D)\ P(W|Z)$$

Ecuación 2.4. Probabilidad de documento y palabra

Donde:

D: Índice del documento

Z: El tópico de la palabra tomado de $P(c|d)$

W: Palabra extraída de $P(c|d)P(w|c)$

Con esto se tiene que $P(Z|D)$ describe cuan probable es ver un documento, y luego, en función de la distribución de temas de este documento, que tan probable es encontrar una determinada palabra dentro del documento [22].

Tanto $P(Z|D)$ y $P(W|Z)$ se modelan como distribuciones multinomiales, y puede ser entrenado usando el algoritmo de maximización de esperanza (EM).

EM es un método para encontrar las estimaciones de parámetros más probables para un modelo que depende de variables latentes no observadas (tópicos).

El modelo $P(D|W)$, puede ser parametrizado usando un conjunto diferente de tres parámetros como se muestra en le ecuación 2.5

$$\mathrm{P}\ (\mathrm{D}|\mathrm{W}) = \sum_{z} \mathrm{P}(\mathrm{Z})\ P(Z|D)\ P(W|Z)$$

Ecuación 2.5. Definición P(D|W) en función de tres términos

Esta equivalencia se puede concebir observando el modelo como un proceso generativo. En donde la primera parametrización se empieza con el documento con P(d), y luego se genera el tema con P (Z | D), y luego se genera la palabra con P (W | Z). Para la parametrización se comienza con el tema P (Z), y luego se genera el documento de forma independiente con P (D | Z) y la palabra con P (W | Z). La figura 2.9 describe la parametrización del modelo PLSA.

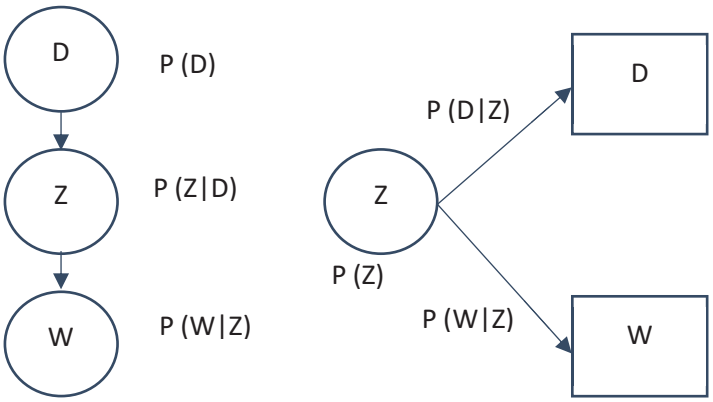


Figura 2.9. PLSA descomposición en tres parámetros

LDA (Laten Dirichlet Allocation / Asignación Latente de Dirichlet)

LDA es la versión bayesiana de pLSA, esta técnica utiliza los antecedentes de *Dirichlet* para las distribuciones de documento – tópico y palabra - tópico.

La distribución Dirichlet se define como una distribución sobre distribuciones lo que en términos probabilísticos se define como un muestreo sobre una probabilidad simple. Es decir, un grupo de números suman uno.

Partiendo de pLSA en el cual se toma una muestra de un documento, luego de un tema basado en ese documento, y luego una palabra basada en ese tema. LDA se define en la figura 2.10

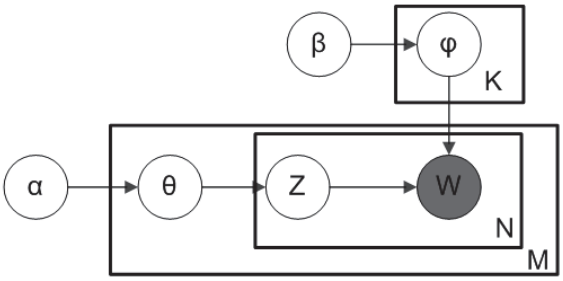


Figura 2.10. Representación gráfica del modelo LDA

A partir de una distribución Dir(α), se dibuja una muestra aleatoria la cual representa la distribución de los tópicos en un documento en particular. Esta distribución de tópicos es θ, partiendo de θ se selecciona un tópico particular Z que responde a dicha distribución.

Luego para la otra distribución Dirichlet Dis(β), se selecciona una muestra aleatoria que representa la distribución de palabras del tópico Z, esta distribución de palabras es φ, tomando φ de referencia se elige la palabra w.

Griffiths y Steyvers [6], describen una representación especial sobre el LDA, la Figura 2.11 muestra la explicación grafica del modelo LDA

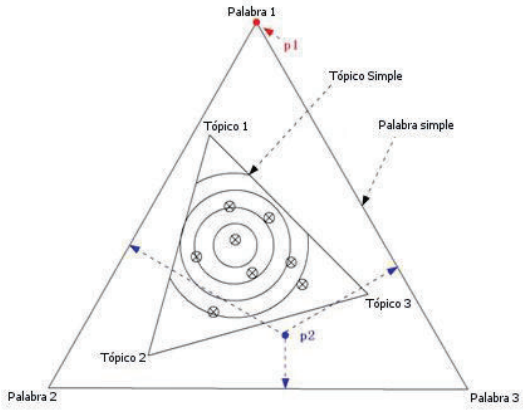


Figura 2.11. Representación espacial de LDA

Se asume que el documento contiene 3 palabras, y que el número de tópicos son 3, de lo cual se tiene que el triángulo que se encuentra por fuera son palabras simples, y cada punto en este representa una probabilidad de distribución, es decir la probabilidad de generar las tres palabras. Por ejemplo, la probabilidad que el punto p2 genere la palabra 1, es la distancia del punto p2 al lado de la palabra p1, la cual es la distancia de p2 a la parte inferior. Ya que la suma de las distancias desde cualquier punto hasta los tres lados en el triángulo equilátero es igual a alta, la suma de las probabilidades es similar a 1, lo cual permite asegurar la precisión de la probabilidad de usar la distancia.

El triángulo pequeño es un tópico simple, donde cada punto representa la distribución de probabilidad que produce los tres tópicos, similar a la palabra simple es decir si tomamos el tópico 1 la representación es “la distribución de probabilidad de tres palabras generadas por el tópico 1”.

Luego de explicar los diferentes modelos dentro de los modelos de tópicos, se debe seleccionar el modelo a implementarse, por lo cual se resume lo siguiente:

La implementación de LSA es rápida, sin embargo, presenta los siguientes inconvenientes:

- Para obtener resultados precisos es necesario contar con un gran conjunto de documentos y vocabulario.
- Es necesario determinar los temas, adicional las palabras que los componen pueden ser positivas o negativas.

pLSA incorpora modelos probabilísticos, lo cual le permite dejar de lado el valor de las palabras (positivo o negativo) para el documento, a pesar de, se tiene que:

- No se cuenta con parámetros que permitan modelar P(D), lo cual no permite agregar probabilidades a nuevos documentos.
- Si el número de documentos crece, pLSA debe aumentar el número de parámetros de forma lineal, por lo cual es susceptible a un sobreajuste.

LDA, es una implementación más compleja que LSA, y presenta mayor complejidad al usar probabilidades bayesianas, no obstante, se tiene las siguientes ventajas al implementar LDA:

- LDA permite la agregación de nuevos documentos.
- En LDA, el conjunto de documentos sirve como dato de entrenamiento para la distribución Dirichlet de distribución de documentos y tópicos, es decir, si no se conoce el documento se puede muestrear fácilmente de la distribución de Dirichlet y continuar desde ahí, contrario a lo que sucede con pLSA, en el cual la probabilidad del documento es un punto fijo en el conjunto de datos.
- Existen variantes del modelo LDA, que permiten analizar corpus extensos tomando diferentes enfoques, ejemplo de esto son los modelos LDA Col, el cual ayuda a buscar la relación entre palabras dentro de un tópico basándose en n gramas, dicho modelo resulta útil al momento de conocer la relación existente entre palabras.

Tomando en consideración lo expuesto, se decide tomar LDA, ya que permite agregar nuevos documentos sin necesidad de ajuste al modelo, adicional a esto se tiene que se basa en un modelo de jerarquía lo cual permite determinar la relación existente entre los tópicos tratados en los diferentes documentos y las palabras que estos los conforman usando la distribución Dirichlet, la cual ayuda a determinar la existencia de tópicos y su relación entre ellos en un corpus.

2.4 Entrenamiento y selección del clasificador

Partiendo de que el modelo LDA, es un modelo que se basa en determinar el valor de la distribución de un tema basándose en la distribución de las palabras en los documentos (distribución sobre distribuciones), se necesita determinar los parámetros que mejor permitan determinar los tópicos, existen varios enfoques para la estimación de los mejores parámetros para el modelo de tópicos usando LDA, sin embargo, se destacan dos, el de maximización y de minimización.

Algoritmo EM (Esperanza - Maximización)

El algoritmo de (Esperanza – maximización) permite encontrar estimadores de máxima verosímil de parámetros en modelos probabilísticos que dependen de variables latentes. Dempster, Laird y Rubin [23] definen el algoritmo EM como un proceso iterativo de estimación con datos perdidos, donde los datos son estimados después de haber imputado los datos perdidos con valores determinados al inicio. Estos valores imputados son actualizados por sus valores predichos usando estos estimadores iniciales de los parámetros. Los parámetros son entonces reestimados, y así sucesivamente, procediendo iterativamente hasta la convergencia.

Para cada iteración el algoritmo realiza dos pasos los cuales son paso – E (esperanza o expectación), el cual consiste en generar datos para simular un problema de datos completos partiendo del conjunto de datos observados del problema inicial (problema de datos incompleto) y el valor actual de los parámetros, el siguiente paso es el paso – M o paso de maximización el cual toma el conjunto de datos completos estimado en el paso – E y aplica logaritmo de la verosimilitud. Ya que se basa en datos no observados, es reemplazado por sus esperanzas condicionales dados los datos observados, donde el paso-E es efectuado usando el ajuste actual para los parámetros desconocidos, esto lo efectúa hasta la convergencia.

Sin embargo, se tiene que al basarse en máximos locales este algoritmo puede llegar a presentar problemas de verosímil, lo cual ha motivado a la búsqueda de nuevos algoritmos de estimación. Griffths y Steyvers [6] proponen que en vez de estimar las distribuciones tema – palabra y las distribuciones de tópico - documento, se debe más bien estimar directamente la distribución posterior sobre Z (asignación de palabras a los tópicos), dadas las palabras observadas W.

Algoritmo utilizando el muestreo de Gibbs

El algoritmo de Gibbs o muestro condicional alternativo, es un algoritmo de la cadena de Markov Monte Carlo, lo cual se define como un proceso discreto, en el cual la probabilidad de que un evento ocurra depende del aventó anterior a este. Esto permite simular una distribución de alta densidad donde cada subconjunto está condicionado al valor de todos los demás. Este muestreo se realiza de manera secuencial hasta que los valores muestreados se aproximan a la distribución objetivo.

Al tratarse LDA de un modelo parametrizado, se debe determinar el número de temas (*k*), para lo cual se toma como premisa los algoritmos de maximización y minimización, teniendo como resultado lo que se muestra en la figura 2.12.

Según Murzintcev Nikita [24], la estimación más acertada del número de tópicos (*k*), es un proceso en el cual se debe considerar el número de iteraciones y la estimación de tópicos, ya que si se toma un número muy grande o muy pequeño esto infiere en el verosímil del valor de *k*.

Si *k* es demasiado pequeño, los tópicos se dividen en pocos contextos semánticos muy generales, mientras que si se *k* es grande los tópicos serian demasiados, así algunos se solaparían, y otros difícilmente serian interpretables.

- Estimación de *k* usando un valor del número de iteraciones muy pequeño:

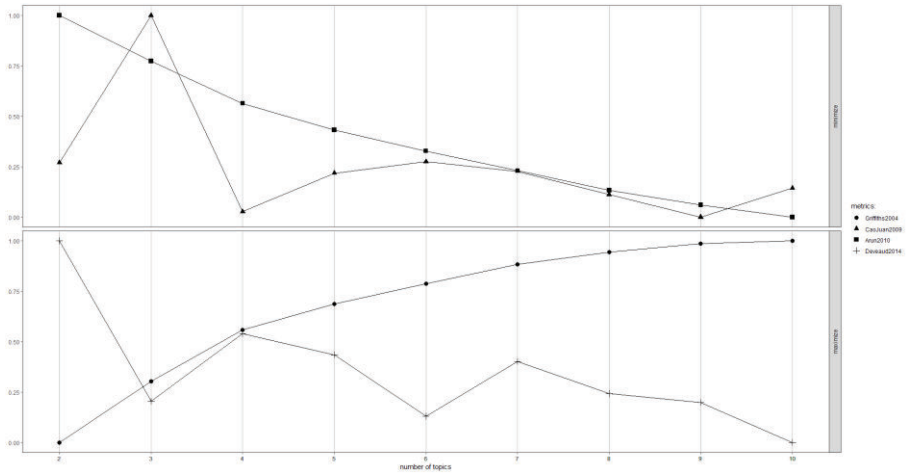


Figura 2.12. Grafica de maximización y minimización para un número de tópicos pequeño

- Estimación de *k* usando un valor del número de iteraciones muy grande:

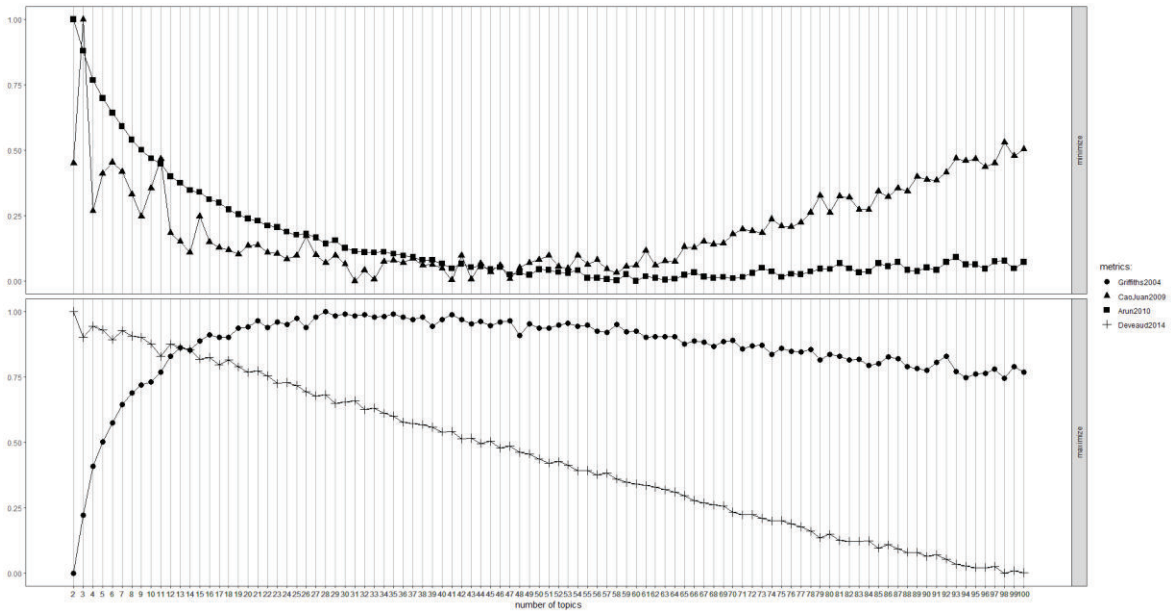


Figura 2.13. Grafica de maximización y minimización para un número de tópicos grande

- Estimación de k usando un valor del número de iteraciones aleatorio.

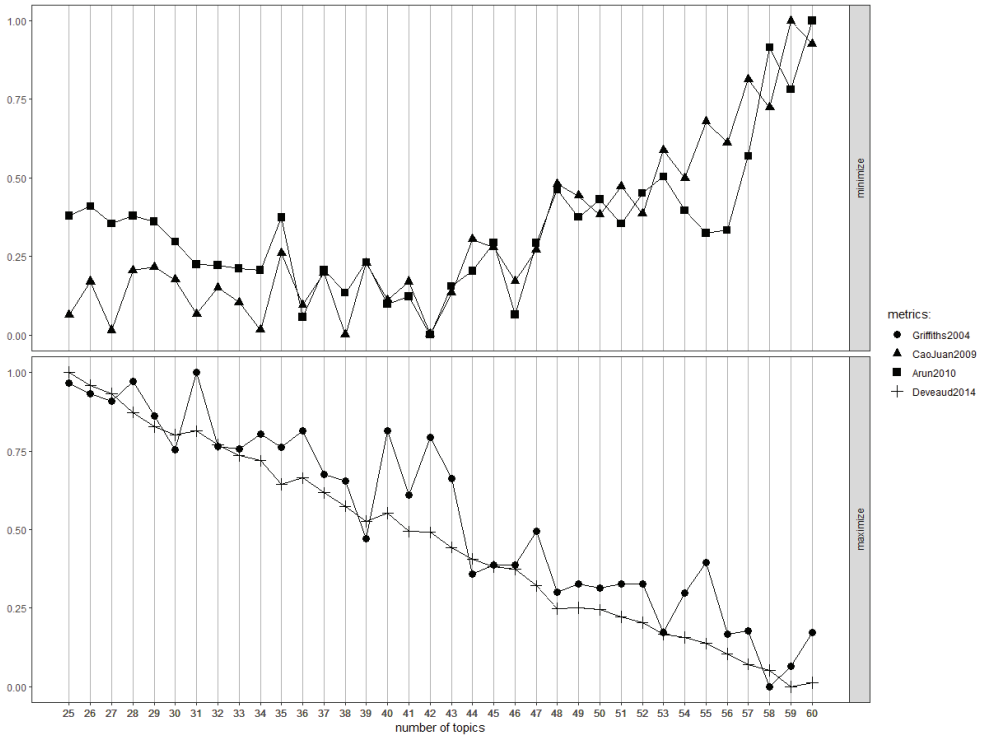


Figura 2.14. Estimación de (*k*) usando maximización y minimización

2.5 IMPLEMENTACIÓN

La implementación practica de modelos de tópicos usando LDA se la realiza utilizando R y MATLAB, así como el conjunto de librerías y funciones.

Como ambiente de desarrollo se utilizó el IDE (Intergated Development Enviroment /Entorno de Desarrollo Integrado) R Studio. Este incluye una consola, editor de resaltado de sintaxis que admite la ejecución directa de código, así como herramientas de depuración y gestión de espacio de trabajo [16].

La implementación de MATLAB se la hace bajo el IDE, propio del lenguaje de programación, el mismo que contiene una consola de ejecución, un editor de código, y un administrador de espacios de trabajo.

A fin de tener documentado las diferentes versiones existentes del código, este se aloja en una plataforma de desarrollo colaborativa GitHub el mismo que brinda el servicio de versionamiento basado en Git, en la figura 2.1 se ilustra el repositorio del proyecto.

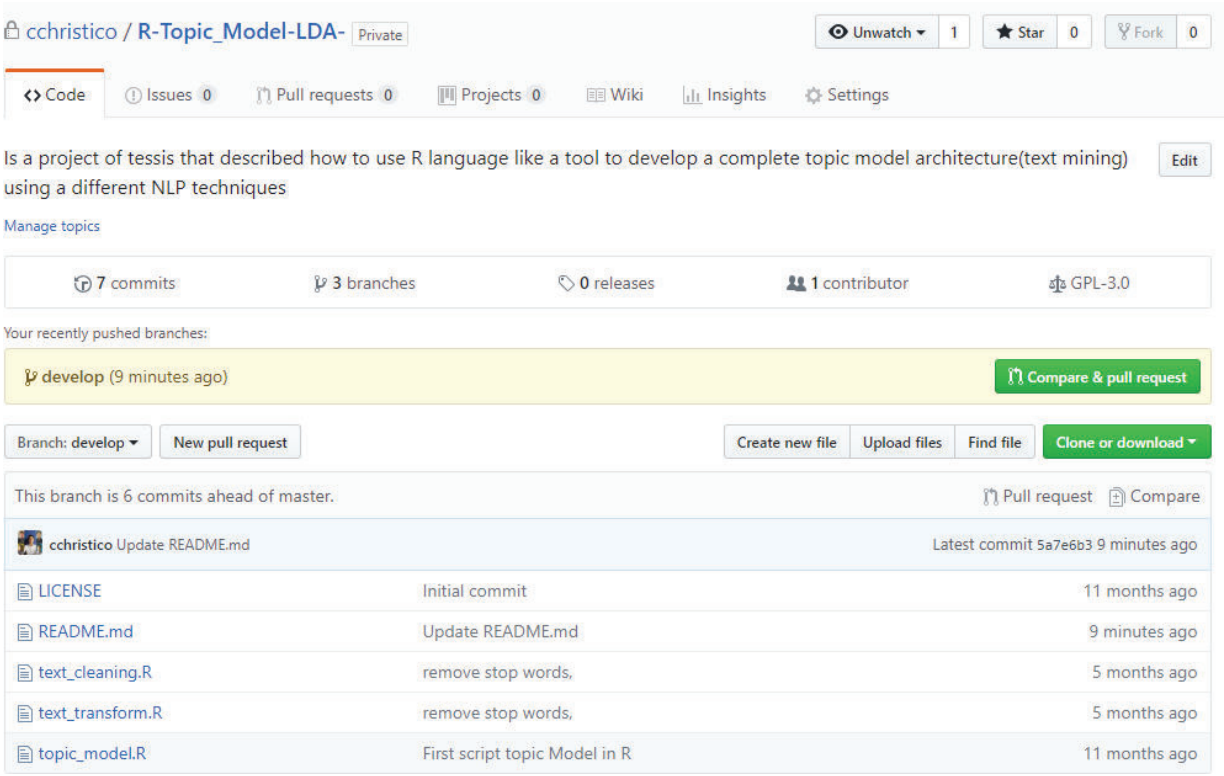


Figura 2.15. Repositorio del Proyecto (GitHub)

2.5.1 Inicialización de las librerías en R

En R se utiliza la librería “easypackages”, la cual permite activar o instalar los paquetes necesarios. La sentencia utilizada para cargar los paquetes se detalla en la figura 2.16.

```
load.lib<-c("easypackages","tm","zoom","stringr","Rstem","ldatuning",
            "parallel","SnowballC","koRpus","wordcloud","topicmodels",
            "tokenizers","R.matlab")
print("#######Validate Packages installed")
install.lib<-load.lib[!load.lib %in% installed.packages()]
for(lib in install.lib) install.packages(lib,dependencies=TRUE)
sapply(load.lib,require,character=TRUE)
print("load libraries")
library("easypackages")
libraries("tm","zoom","stringr","Rstem","ldatuning","parallel","SnowballC",
          "koRpus","wordcloud","topicmodels","tokenizers","R.matlab")
```

Figura 2.16. Inicialización de los paquetes necesarios

2.5.2 Preprocesamiento de Texto

Previo al preprocesamiento de texto, se debe transformar la colección de documentos a una estructura admisible por R, esto se lo realiza con el paquete *tm*, es importante notar que la codificación de los documentos afecta al proceso de limpieza, por lo cual la sentencia a ser utilizada para descartar estos problemas se especifica en la figura 2.17.

```
# Create a Corpus to load all files
# change the codification to UTF-8
text <- Corpus(DirSource("C:/TopicModel/", encoding="UTF-8"),
               readerControl = list(reader = readPlain, language = "english"))
```

Figura 2.17. Creación de Corpus en R

La función “fn_text_clean”, Figura 2.18, integra varias utilidades de la librería *tm* para limpieza de texto, adicional a estas utilidades, se implementan funciones propias, que aseguran este proceso.

```
fn_text_clean <- function(corpus_txt)
{
  for(i in 1:length(corpus_txt))
  {
    corpus_txt[[i]]$content <-bracketXtract(corpus_txt[[i]]$content)
    corpus_txt[[i]]$content<-str_replace_all(corpus_txt[[i]]$content,"\016"," ")
    corpus_txt[[i]]$content <-gsub("[^a-zA-Z0-9]","  ", corpus_txt[[i]]$content)
  }

  # Remove stop word using tm package
  corpus_txt <- tm_map(corpus_txt, removeWords, stopwords('english'))

  # change tolower words
  corpus_txt <- tm_map(corpus_txt, content_transformer(tolower))

  # remove numbers
  corpus_txt <- tm_map(corpus_txt, removeNumbers)
  # custom stop words
  stopwords_C <- read.csv("C:/TopicModel/topicModel_utilities/stop-word-list.csv", header = FALSE)
  stopwords_C <- as.character(stopwords_C$V1)
  #remove custome stop words
  corpus_txt <- tm_map(corpus_txt, removeWords, stopwords_C)
  #remove extra white spaces
  corpus_txt <- tm_map(corpus_txt, stripWhitespace)

  return(corpus_txt)
  }
```

Figura 2.18. Función de limpieza de texto(fn_text_clean)

2.5.3 Reducción de términos

La reducción de texto se la realiza por fuera del proceso de limpieza, ya que estos procesos pueden o no ser implementados. Para el proceso de Stemming y Lemmatización, se crea la función “steam_lema”, ver Figura 2.19.

```
##########################################################
#                                                        #
# text transform                                         #
# This funcion helps to lematize and steam               #
# words in english language, use porter                  #
# Parameters: text                                       #
# return text lematized                                  #
#                                                        #
##########################################################
steam_lema <-
  function(txt){
    if (!require("pacman")) install.packages("pacman")
    pacman::p_load_gh("trinker/textstem")

    for(i in 1:length(txt))
    {
      words <- tokenize_words(txt[[i]]$content)
      y<-stem_words(words[[1]],lenguage="english")
      y<-lemmatize_words(y,dictionary = lexicon::hash_lemmas)
      txt[[i]]$content<-paste(y, collapse = ' ')
    }

    return(txt)
  }
```

Figura 2.19. Función de Lemmatización y stemming

Posterior a la limpieza de texto, se realiza la extracción de características, lo cual se lo realiza con la obtención de la matriz de documento – término, dicha matriz es el principal insumo para el modelo LDA, razón por la cual se debe considerar que pueden existir

términos que no aporten valor a los tópicos, dichos términos deben ser eliminados de la matriz, por lo que se debe aplicar la función que se especifica en la Figura 2.20, dicha función permite eliminar palabras que tengan una baja frecuencia dentro del corpus.

```
dtm <- removeSparseTerms(marizTerm , 0.990)
```

Figura 2.20. Remoción de términos menos frecuentes

2.5.4 Selección de Parámetros para el Modelo

Una vez obtenida la matriz de documento término se debe seleccionar los parámetros aplicables al modelo, la selección de dichos parámetros consiste en la obtención de *k.*

Es importante apoyarse en la librería *parallel,* para que este proceso pueda dividirse en subtareas, esto se lo realiza con el fin de disminuir el tiempo de obtención de *k*, se debe considerar que al disminuir el tiempo de ejecución se aumenta los recursos necesarios para cumplir esta tarea.

La figura 2.21 muestra los recursos utilizados para la selección de *k*, esto se lo lleva a cabo usando 6 cores, cada subrutina ejecutada por cada core ocupa en promedio en 15% del uso total de la CPU, adicional dichas subrutinas utilizan el 1,2% de memoria total del sistema.

RStudio (32 bits) (9)	90,8%	11,0%
R for Windows front-end	15,1%	1,1%
R for Windows front-end	15,0%	1,1%
R for Windows front-end	15,0%	1,1%
R for Windows front-end	15,0%	1,1%
R for Windows front-end	15,0%	1,1%
R for Windows front-end	15,0%	1,1%
RStudio (32 bits)	0,5%	1,8%
RStudio R Session	0%	2,8%
Host de ventana de consola	0%	0,1%

Figura 2.21. Recursos utilizados en la obtención de *k*

La Figura 2.22, muestra el tiempo que toma el proceso de selección de *k,* de lo cual se tiene que los primeros dos tiempos hacen referencia a tiempos totales de CPU del usuario y del sistema del proceso de R, y cualquier proceso secundario tiene que esperar para su ejecución, el tercer tiempo es el tiempo total que tomo la ejecución del proceso.

```
fit models... done.
calculate metrics:
  Griffiths2004... done.
  CaoJuan2009... done.
  Arun2010... done.
  Deveaud2014... done.
   user  system elapsed
   5.32    0.22  124.38
```

Figura 2.22. Tiempos de ejecución para la selección de k

2.5.5 Modelo de Tópico LDA

La ejecución del modelo LDA, sobre la matriz documento – término se la realiza con el método de Gibbs, adicional a esto se seleccionan los parámetros de α, el cual es la distribución de los tópicos por documento. Si se selecciona un valor alto de α la distribución de los tópicos será homogénea, mientras que un valor bajo de α previene que el proceso de inferencia distribuya el porcentaje de probabilidad en algunos temas.

2.5.6 Modelo de Tópicos en Matlab

El preprocesamiento del corpus y la generación de la matriz de documento – término y su posterior reducción de términos menos frecuentes y la implementación de LDA se lo realiza en R, además se implementa LDA_Col en Matlab para lo cual se hace uso de topicmodeltoolbox. Este conjunto de librerías propias de Matlab tiene sus inputs, los cuales viene del proceso previo realizado en R.

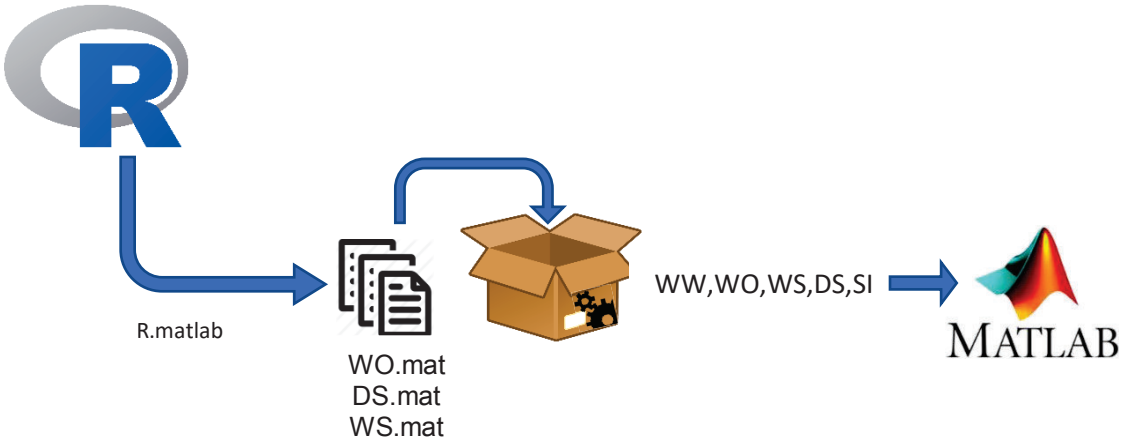


Figura 2.23. Paso de estructuras de R a Matlab

En la figura 2.23, se visualiza el paso de estructuras, R hace uso de la librería R.matlab para exportar documentos WO.mat, DS.ma y WS.mat los cuales pasan a topictoolbox de

Matlab, para transformar estos documentos en estructuras aceptadas por Matlab, las cuales pasan a ser transformadas en WW, WO, WS, DS y SI, estos parámetros son el insumo que utiliza Matlab para la generación de los modelos LDA y LDA_Col.

2.5.7 Creación de archivos .mat:

La creación de los archivos .mat, se lo lleva a cabo con la función writeMat(), esta función toma estructuras de R, y las coloca en una estructura de archivo MAT, que luego se escribe en una conexión binaria.

Los documentos resultantes son:

WO : array de las palabras que se encuentran en la matriz de documento – termino, esta es la estructura que demora más en ejecutarse, ver la figura 2.24

```
+   result1 <-writeMat("D:/WO.mat",WO=vocabulary))
   user  system elapsed
 440.81    0.08  441.73
```

Figura 2.24. Tiempo transcurrido para la generación de WO

DS: Vector que denota el índice de los documentos, y la relación de las palabras con dichos documentos. Ver la figura 2.25.

```
> system.time(
+   result2 <-writeMat("D:/DS.mat", DS = dtm[[1]]))
   user  system elapsed
   0.19    0.05    0.77
```

Figura 2.25. Tiempo transcurrido para la generación de DS

WS: Vector del índice del vocabulario y su incidencia por documento. Ver la figura 2.26

```
> system.time(
+   result3 <-writeMat("D:/WS.mat", WS = dtm[[2]]))
   user  system elapsed
   0.19    0.02    0.61
```

Figura 2.26. Tiempo transcurrido para la generación de WS

2.5.8 Transformación a datos de colocación

En la lingüística computacional basada en corpus una colocación (collocation), es una secuencia de palabras o términos que coocurren con mucha frecuencia. Esto permite la obtención de los n-gramas, insumo necesario para implementación de LDA_COL.

La obtención de los datos collocation, se lo realiza desde Matlab, partiendo de los documentos WO, DS, WS. Ver figura 2.27.

```
% Load Data From R
load('DS.mat');
load('WO.mat');
load('WS.mat');

%transpose to generate a collation data
WS1=transpose(double(WS));
DS1=transpose(double(DS));

% generate a collocation data
    [ WW , WO , WS , DS , SI]=stream_to_collocation_data(DS1,WS1,WO,'stopwordlist.txt');
    save 'chain_suplly' WW WO DS WS SI;
```

Figura 2.27. Obtención de los datos collocation

La figura 3.13, muestra la carga de los archivos obtenidos en R, y su procesamiento para la obtenciond e los datos collocation, este procesamiento consiste en:

1) Transponer las estructuras DS y WS.

2) Función de conversión stream_to_collocation_data

3) Extracción de Stop Words(opcional)

2.5.9 Selección de Parámetros y Obtención del Modelo

Al igual que en R, los parámetros seleccionados se muestran en la figura 2.28.

```
T = 45;
OUTPUT = 1;
BETA   = 0.01;
ALPHA  = 0.02;
GAMMA0 = 0.1;
GAMMA1 = 0.1;
DELTA  = 0.1;
MAXC   = 4;
N = 200;
%The seed number
SEED = 123;
```

Figura 2.28. Parámetros para le ejecución de LDA_COL

La figura 2.29, muestra el proceso que se realiza para la obtención del modelo LDA_COL.

```
tic
[ WP,DP,WC,C,Z ] = GibbsSamplerLDACOL( WS , DS , SI , WW , T , N , ALPHA , BETA , GAMMA0, GAMMA1 , DELTA , SEED , OUTPUT  );
toc

% convert topics to include collocations
[ WPNEW , DPNEW , WONEW ] = CreateCollocationTopics( C , Z , WO , DS , WS , T , MAXC );

fprintf( 'Writing collocation topics to file: %s\n' , filenm );

 S = WriteTopics( WPNEW , BETA , WONEW , 20 , 0.7 , 5 , filenm );
    save 'ldacol_psychreview' WPNEW DPNEW WONEW WP DP WC C ALPHA BETA GAMMA0 GAMMA1 DELTA SEED N Z T;
```

Figura 2.29. LDA_COL y COLLATION_TOPICS

De lo cual se tiene que:

GibbsSamplerLDACOL: Permite obtener el conteo de palabra-tópico (WP) y documento – tópico (DP), para todas las palabras asignadas a un tópico, así como también el número de veces que la palabra i condujo a la collocation con la siguiente palabra (WC). C determina si existe una relación entre palabras (n-gramas) y Z contiene las asignaciones de temas para todos los tokens de palabras.

CreateCollocationTopics: Crea un nuevo vocabulario y conteos de tópicos basado en el resultado del modelo de Collocation, para lo cual las variables C y Z permiten determinar la ruta (0=tópico; 1= collocation) y WO y DS, contiene los índices de palabra y documentos de entrada del corpus al modelo y T el número de tópicos.

3. RESULTADOS Y DISCUSIÓN

A continuación, se muestran las pruebas de selección de los parámetros utilizables para los modelos LDA y LDA_Col y los resultados arrojados tras la implementación de los modelos de tópicos. Las pruebas que se realizaron para la selección de los parámetros para el modelo son las siguientes:

3.1 Prueba 1: Selección del parámetro *k* (número de tópicos)

Las pruebas para la selección del número de tópicos consisten en la evaluación de las iteraciones y los recursos necesarios para la ejecución de los algoritmos de maximización y minimización.

- Número de iteraciones bajo, con recursos mínimos:

Numero de Cores: 1

Numero de iteraciones [2,3…, 10]

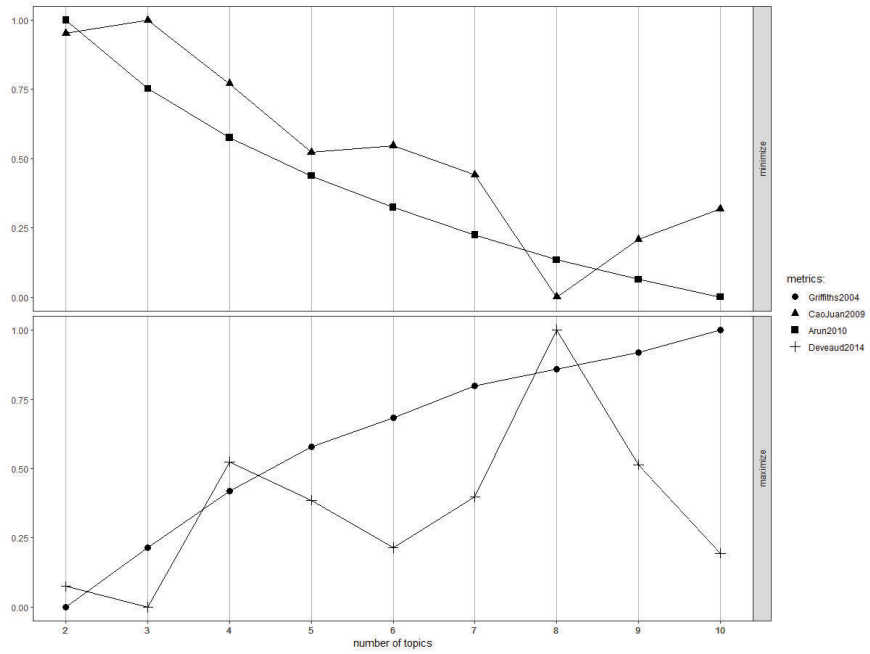


Figura 3.1 Iteraciones [2,3…,10]

La Figura 3.1 especifica las diferentes iteraciones en un rango de 2 a 10, según las técnicas de maximización y minimización, estas no intersecan en un número entero, por lo que el número de tópicos k, está fuera de este rango.

El tiempo que toma las 10 iteraciones asignadas a un solo core, es relativamente bajo, ver la tabla 3.1.

Usuario	Sistema	t [s]
0.47	0.02	38.88

Tabla 3.1 Tiempos de ejecución

- Número de iteraciones alto, con la mayor cantidad de recursos disponibles:

Numero de Cores: 6

Numero de iteraciones [4,5,6…100]

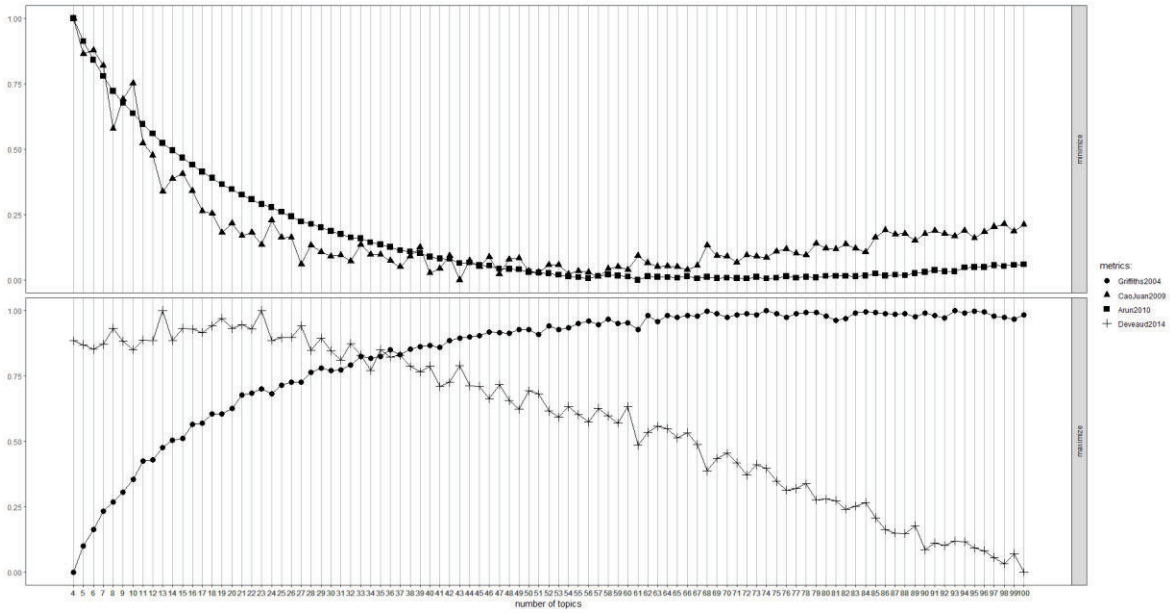


Figura 3.2. Iteraciones [4,5,6… 100]

La Figura 3.2 especifica las diferentes iteraciones en un rango de 4 a 100, según las técnicas de maximización el algoritmo de Griffiths y el de Deveaud la dispersión de los números de tópicos se alejan cada vez mas de su estimación inicial, mientras que según los algoritmos de Arun y de CaoJuan llegan a intersecarse por primera vez en $k = 45$, para luego alejarse de nuevo.

Según la tabla 4.2, el tiempo que toma las 96 iteraciones asignadas a 6 cores, es considerable, ya que el tiempo total transcurrido es de 721.77[s].

Usuario	Sistema	t [s]
6.94	0.39	721.77

Tabla 3.2. Tiempos de ejecución

- Número de iteraciones alto y cambiante, con la mayor cantidad de recursos disponibles:

Numero de Cores: 6

Numero de iteraciones [15:36,38,40,45,50,60,70,80,90,100,120,140]

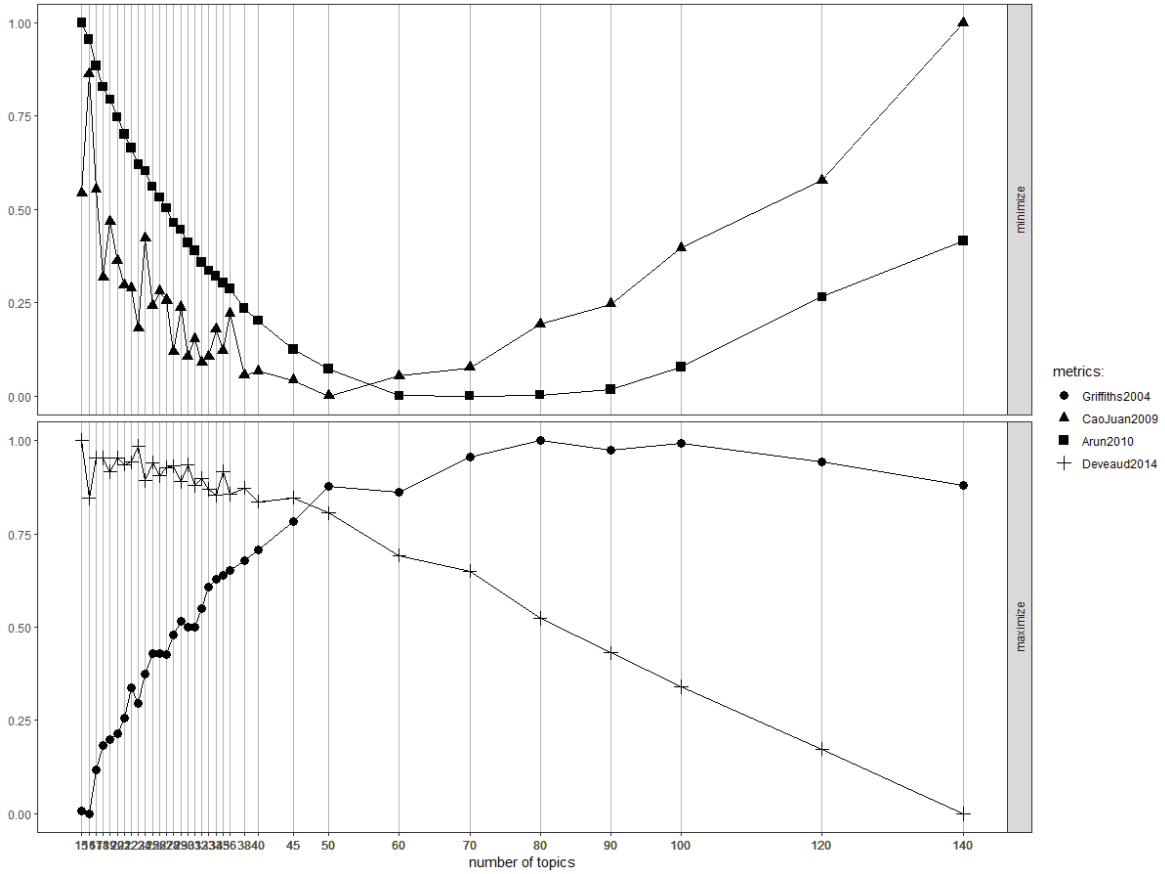


Figura 3.3. Iteraciones [15:36,38,40,45,50,60,70,80,90,100,120,140]

La Figura 3.2 especifica las diferentes iteraciones en un rango alto y cambiante, de lo cual se tiene que, según las técnicas de maximización el algoritmo de Griffiths y el de Deveaud la dispersión de los números de tópicos se alejan cada vez más de su estimación inicial, mientras que según los algoritmos de Arun y de CaoJuan estos llegan a intersecarse por única vez en $k = 55$, para luego alejarse de nuevo.

La tabla 3.3, muestra el tiempo que toma las 96 iteraciones asignadas a 6 cores, este tiempo es considerable, ya que el tiempo total transcurrido es de 598.94 [s].

Usuario	Sistema	t [s]
4.14	0.66	598.94

Tabla 3.3. Tiempos de ejecución

Las pruebas para la estimación del número de tópicos arrojo como resultado, que el caso dos es el que mejor estima el valor de k.

Esta estimación es la más acertada debido a que los algoritmos de minimización llegan a un punto donde la probabilidad tanto del método de Griffiths y el de Deveaud y la dispersión de los números de tópicos se intersecan hasta llegar a alejarse cada vez más de su estimación inicial; mientras que según los algoritmos de Arun y de CaoJuan en el rango en el cual los algoritmos de minimización se intersecaron los de maximización empezaron a alejarse, lo cual permite determinar que dicho rango describe de mejor manera el número de tópicos. Se considera a $k = 45$, ya que es el punto en el cual los algoritmos de minimización intersecan en su totalidad por primera vez.

3.2 Prueba 2: Selección del parámetro Alpha

- Alpha = Número de tópicos / 50

En este caso el valor de alpha será $\alpha = 45/50$ es decir 0.9, la figura 4.4, muestra la distribución de los tópicos dentro de 4 documentos diferentes.

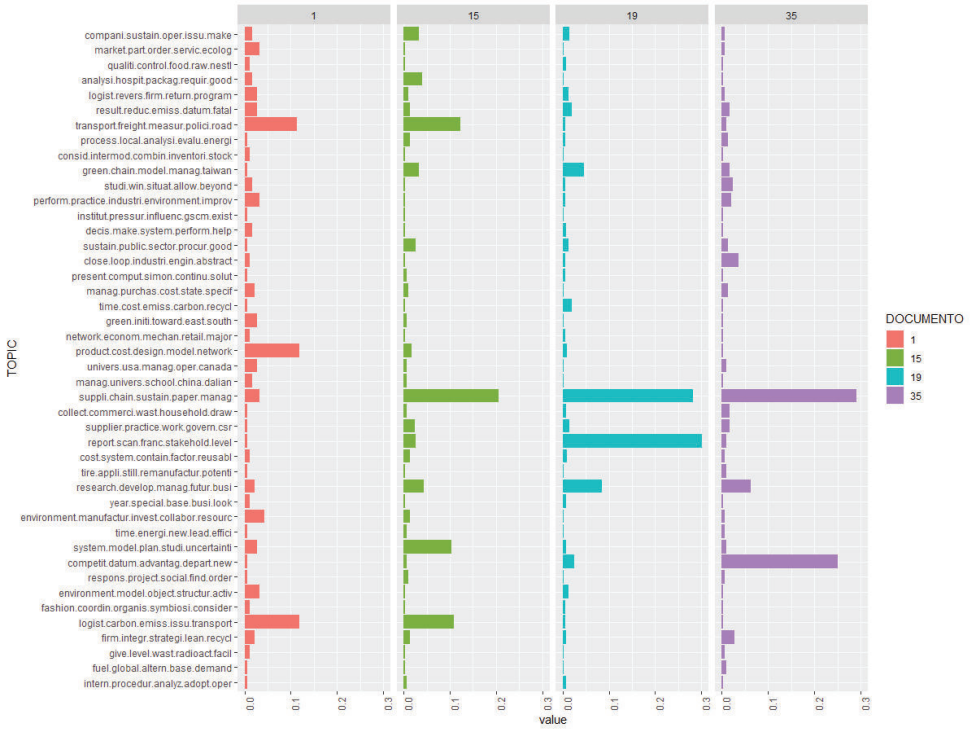


Figura 3.4. Distribución de los tópicos con $\alpha = 0.9$

La figura 3.4 muesta a la mayoria de los tópicos presentes en cada documento, lo cual puede llegar a presentar un problema de ambigüedad.

- Alpha 0.5

La estimación de $\alpha = 0.5$ muestra una mejor distribución de los tópicos, como se evidencia en la Figura 4.5.

Figura 3.5. Distribución de los tópicos con $\alpha = 0.5$

En la figura 3.5 algunos tópicos se muestran en menor probabilidad, pero siguen generarndo cierta ambigüedad al analisis de los documentos y sus respectivos tópicos

- Alpha 0.2

La estimación de $\alpha = 0.2$ muestra una mejor distribución de los tópicos, como se evidencia en la Figura 3.6.

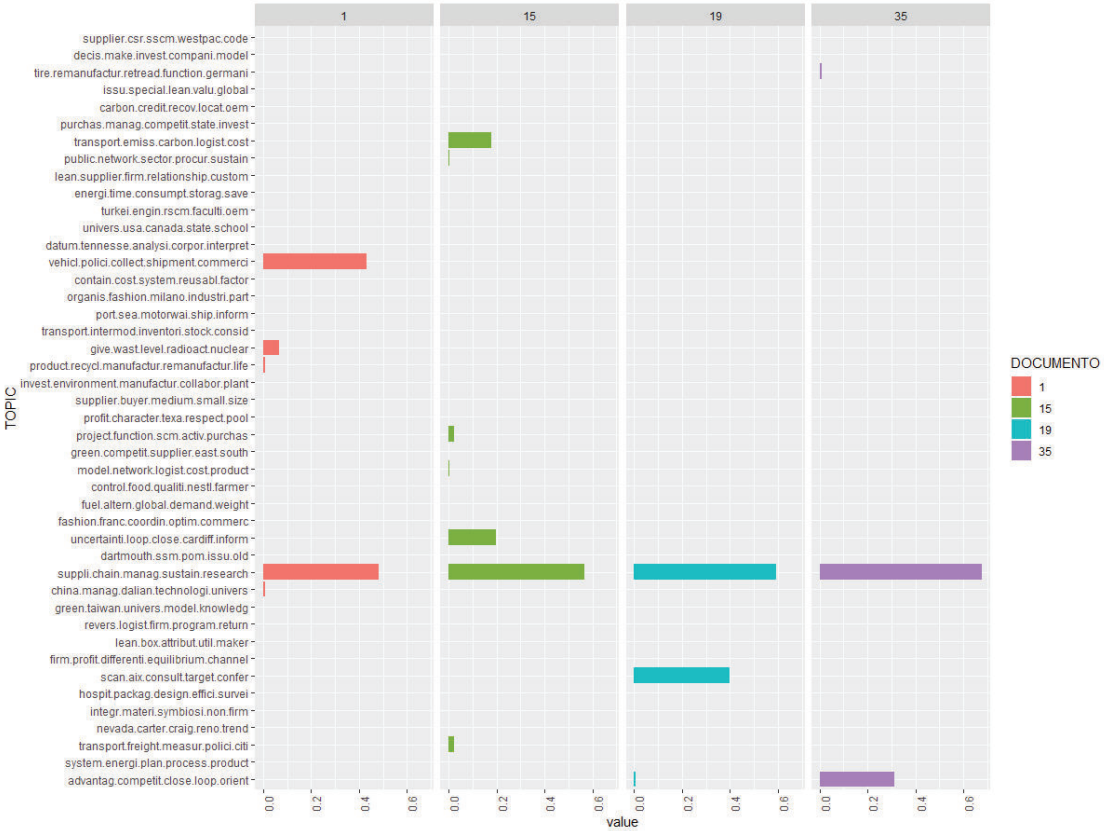


Figura 3.6. Distribución de los tópicos con $\alpha = 0.2$

La figura 3.6 muestra la distribución de los tópicos usando un parámetro de inferencia de probabilidades igual a 0.2 este valor ayuda a que los tópicos se distribuyan de mejor manera dentro de los documentos, y solo se evidencien los más significativos para dichos documentos.

3.3 Resultado del Modelo LDA

Como resultado del modelo de tópicos se tiene dos distribuciones de probabilidad posteriores: θ hace referencia a los k tópicos dentro de cada documento y β es una distribución sobre los términos V dentro de cada tópico, siendo V el vocabulario del corpus.

Partiendo del corpus de 168 documentos, se llega a un modelo topicModel que cuenta con 45 tópicos, de lo cual se tiene las siguientes consideraciones:

```
topicModel:
LDA_Gibbs topic model with 45 topics
Número de términos: 4081
Beta: 45 4081
Theta: 168 45
```

Estos parámetros permiten identificar los diferentes tópicos existentes en el conjunto de documentos, cada tópico tiene su probabilidad dentro de cada documento, al igual que cada palabra tiene su probabilidad dentro de cada tópico, con esto se puede determinar los tópicos existentes, su relevancia y los términos que a estos describen.

Como primer resultado se muestran los diferentes tópicos y una muestra de 5 términos que componen dicho tópico ver la Tabla 3.4

Topic 1	relationship studi relat knowledg share	Topic 23	emiss cost carbon logist consid
Topic 2	supplier manag purchas firm small	Topic 24	univers usa manag oper canada
Topic 3	supplier csr busi work sscm	Topic 25	european dismantl abstract advantag depart
Topic 4	transport measur provid oper result	Topic 26	tire consider examin ecolog kei
Topic 5	lean global green activ strategi	Topic 27	sourc fuel altern global hospit
Topic 6	manag china univers technologi school	Topic 28	organis part form question process
Topic 7	datum corpor trade interpret valid	Topic 29	decis model system make object
Topic 8	professor constraint vari compani dimens	Topic 30	green industri depart univers taiwan
Topic 9	respons public sector govern procur	Topic 31	show intermod box evalu order
Topic 10	busi close loop area valu	Topic 32	project develop analysi countri research
Topic 11	product environment cost model recycl	Topic 33	system model plan energi unit
Topic 12	industri distribut weight end manufactur	Topic 34	consum social assess requir factor
Topic 13	green model structur initi institut	Topic 35	give wast level radioact time
Topic 14	energi lead time storag save	Topic 36	process local itali output ingegneria
Topic 15	invest manufactur collabor plant industri	Topic 37	scan centr report franc view
Topic 16	control qualiti social food studi	Topic 38	logist revers firm research effect
Topic 17	administr procedur concept intern port	Topic 39	environment perform practice improv organ
Topic 18	fashion function optim toward profit	Topic 40	collect commerci southampton joint common
Topic 19	network logist design method sustain	Topic 41	polici freight initi take citi
Topic 20	portfolio suggest object plai specif	Topic 42	issu special focu editor featur
Topic 21	cost system contain factor reusabl	Topic 43	suppli chain sustain research manag
Topic 22	level recoveri product close remanufactur	Topic 44	network collabor mechan chang retail
		Topic 45	compani sustain mani custom issu

Tabla 3.4. Tópicos y los 5 primeros términos

La tabla 3.4 resume el resultado arrojado por el modelo LDA, no obstante, para un análisis más congruente es necesario conocer la probabilidad de cada tópico en el corpus. Analizando la probabilidad de los tópicos latentes en el corpus, se los organiza de mayor a menor.

La Tabla 3.5 indica el top 10 de tópicos y su probabilidad de ocurrencia dentro del corpus.

TOPIC	% Prob
TOPIC_30	2,86%
TOPIC_41	2,80%
TOPIC_28	2,77%
TOPIC_13	2,69%
TOPIC_19	2,49%
TOPIC_42	2,46%
TOPIC_8	2,42%
TOPIC_21	2,41%
TOPIC_43	2,41%
TOPIC_11	2,33%

Tabla 3.5. Tópicos ordenados de mayor a menor probabilidad de ocurrencia

La figura 3.7 esquematiza los tópicos y su porcentaje de ocurrencia:

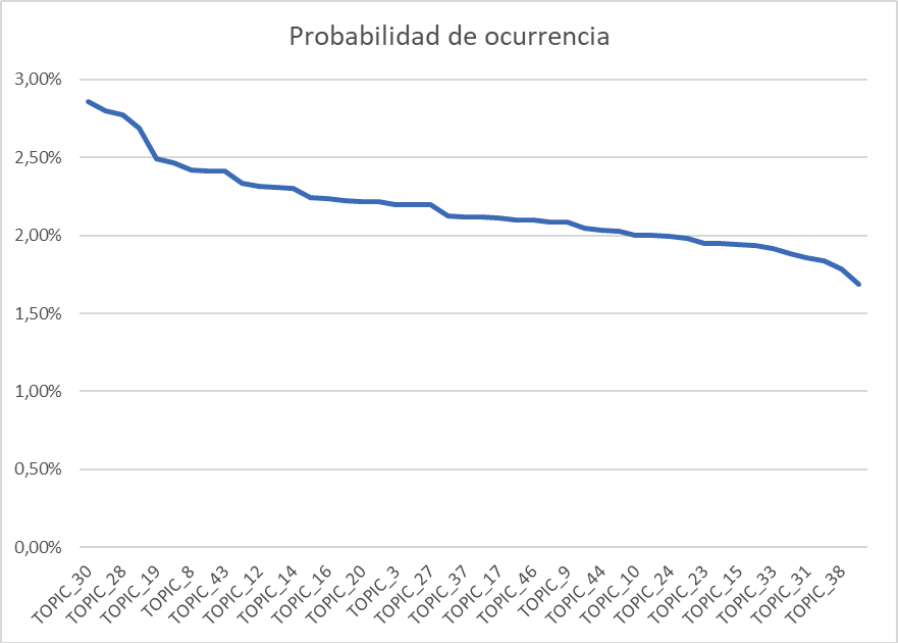


Figura 3.7. Probabilidad de ocurrencia de todos los tópicos

La Tabla 3.6, indica todos los tópicos con su probabilidad de ocurrencia y la probabilidad de cada palabra dentro de cada tópico:

TOPIC_1	0.02226	TOPIC_2	0.02086	TOPIC_3	0.02198	TOPIC_4	0.01886	TOPIC_5	0.01937	TOPIC_6	0.02049
servic	0.06909	green	0.08265	logist	0.05937	becom	0.07907	product	0.08901	technologi	0.07051
area	0.06490	research	0.07595	transport	0.04877	issu	0.05684	consum	0.05534	investig	0.04095
import	0.05653	busi	0.07371	paper	0.04665	limit	0.04696	util	0.03850	research	0.03640
challeng	0.04816	insight	0.05585	green	0.04665	point	0.04696	market	0.03369	recent	0.03185
result	0.04816	keyword	0.03575	chain	0.04241	social	0.04449	materi	0.03129	situat	0.02958
level	0.04397	item	0.03129	taiwan	0.04029	wai	0.03708	respect	0.03129	scienc	0.02958

TOPIC_7	0.02198	TOPIC_8	0.02422	TOPIC_9	0.02086	TOPIC_10	0.02003	TOPIC_11	0.02333	TOPIC_12	0.02315
univers	0.08692	decis	0.06928	pollut	0.03129	includ	0.06517	compani	0.09986	studi	0.10469
supplier	0.08692	case	0.05773	control	0.03129	paper	0.05586	differ	0.06192	term	0.06644
manufactur	0.05301	sustain	0.05004	appendix	0.02682	survei	0.05586	practice	0.05992	need	0.05034
major	0.04241	system	0.04811	canada	0.02682	type	0.04423	provid	0.05793	establish	0.03625
environment	0.03393	final	0.04234	western	0.02682	scale	0.03725	expect	0.03796	risk	0.03424
sector	0.02969	end	0.04042	plant	0.02459	initi	0.03260	determin	0.03596	tradit	0.03223

TOPIC_13	0.02687	TOPIC_14	0.02305	TOPIC_15	0.01942	TOPIC_16	0.02235	TOPIC_17	0.02114	TOPIC_18	0.01998
chang	0.05029	perform	0.06873	abstract	0.12478	busi	0.10423	sustain	0.06833	model	0.08164
cost	0.03989	research	0.06469	purpos	0.07200	resourc	0.06255	larg	0.05511	futur	0.05133
effici	0.03469	empir	0.06469	aim	0.05521	support	0.05629	suggest	0.05070	issu	0.04666
low	0.03469	framework	0.05863	rel	0.03841	nation	0.05212	build	0.04409	contribut	0.04433
emiss	0.03296	author	0.05054	awar	0.03601	colleg	0.04796	mean	0.03308	order	0.03967
transport	0.03122	construct	0.03438	suppli	0.03361	intern	0.04170	impact	0.02867	appli	0.03500

TOPIC_19	0.02492	TOPIC_20	0.02217	TOPIC_21	0.02413	TOPIC_22	0.02217	TOPIC_23	0.01951	TOPIC_24	0.01993
organ	0.06172	consid	0.05677	improv	0.07534	competit	0.07358	measur	0.08598	suppli	0.09586
global	0.06172	result	0.04626	firm	0.05216	futur	0.07148	evalu	0.05494	chain	0.07950

exampl	0.05050	help	0.03995	provid	0.05216	usa	0.04626	account	0.04300	institut	0.07482
relat	0.04676	focus	0.03365	posit	0.04251	understand	0.03575	analysi	0.03106	develop	0.03976
issu	0.03928			process	0.04057	examin	0.03365	standard	0.03106	price	0.03976
polici	0.03742			high	0.03671	base	0.03155	hold	0.02868	sustain	0.03275
conclus	0.03742			multipl	0.03671	enhanc	0.03155	sever	0.02868	success	0.03275
TOPIC_25	0.01835	TOPIC_26	0.02128	TOPIC_27	0.02198	TOPIC_28	0.02771	TOPIC_29	0.02310	TOPIC_30	0.02855
design	0.06605	requir	0.06132	base	0.07844	paper	0.06895	find	0.11700	valu	0.05060
creat	0.05589	discuss	0.05037	influenc	0.04877	methodologi	0.06558	relationship	0.06053	current	0.04408
oper	0.04573	complex	0.04161	method	0.04453	approach	0.05718	increas	0.05246	optim	0.04081
econom	0.04319	strateg	0.03505	comput	0.03605	research	0.04709	effect	0.05044	inform	0.03918
introduc	0.03811	produc	0.03505	gener	0.03393	origin	0.04373	limit	0.04439	effect	0.03592
line	0.03558	tool	0.03067	recent	0.02969	integr	0.04037	explor	0.04036	demand	0.03265
adopt	0.03304	order	0.03067	financi	0.02969	practice	0.03532	case	0.04036	applic	0.03102
TOPIC_31	0.01858	TOPIC_32	0.01951	TOPIC_33	0.01919	TOPIC_34	0.02240	TOPIC_35	0.02100	TOPIC_36	0.01979
question	0.03764	network	0.08121	analysi	0.09473	depart	0.11025	respons	0.10206	custom	0.08241
project	0.03513	field	0.04539	interest	0.08259	model	0.08321	due	0.03995	involv	0.07300
school	0.03262	level	0.04539	present	0.08259	suppli	0.07073	corpor	0.03330	signific	0.06829
specif	0.02761	valu	0.04062	oper	0.07287	engin	0.05826	implic	0.02886	extend	0.04946
basi	0.02510	centr	0.04062	environ	0.03402	revers	0.04578	real	0.02886	materi	0.04710
cultur	0.02259	local	0.03823	product	0.02188	problem	0.04162	describ	0.02886	school	0.03533
complet	0.02259	gener	0.03584	entir	0.01945	process	0.03954	firm	0.02664	suppli	0.02592
TOPIC_37	0.02119	TOPIC_38	0.01784	TOPIC_39	0.01690	TOPIC_40	0.02026	TOPIC_41	0.02799	TOPIC_42	0.02464
industri	0.08137	suppli	0.07839	manag	0.14335	strategi	0.06672	environment	0.06493	introduct	0.11726
consider	0.05059	chain	0.07578	univers	0.09925	environ	0.05522	activ	0.05328	product	0.06621
mani	0.04620	logist	0.06272	environment	0.06893	aspect	0.05062	cost	0.04995	keyword	0.04919
variabl	0.04180	manag	0.04182	region	0.02483	benefit	0.04372	cycl	0.04496	design	0.04540
continu	0.03740	non	0.04182	east	0.02483	variou	0.04372	propos	0.04329	reduc	0.04162

social	0.03300	market	0.03399	govern	0.01932	move	0.03222	manag	0.03497	reduct	0.03595

TOPIC_43	0.02413	TOPIC_44	0.02035	TOPIC_45	0.02119
environment	0.06568	chain	0.10762	manag	0.08797
perspect	0.04251	show	0.06412	univers	0.06818
develop	0.03671	concern	0.05497	develop	0.04400
perform	0.03671	factor	0.05039	studi	0.03740
increasingli	0.03285	keyword	0.04810	joseph	0.02861
manufactur	0.03285	associ	0.04810	evalu	0.02641
address	0.03092	object	0.04123	system	0.02641

Tabla 3.6. Probabilidad de tópicos con sus respectivos términos con probabilidad de ocurrencia

La tabla 3.6 muestra un conjunto extenso de información, la cantidad de términos por tópicos para ser analizados es muy grande, por esta razón se aplica el principio de Pareto, para lo cual se propone que los términos que sumen una probabilidad aproximada al 20% describirán al menos el 80% de los que trata el tópico.

La tabla 3.7 muestra los tópicos ordenados según su probabilidad de ocurrencia, así como también los términos más significativos para cada tópico.

TOPIC_30	Prob	TOPIC_41	Prob	TOPIC_28	Prob	TOPIC_13	Prob
valu	5,06%	environment	6,49%	paper	6,90%	chang	5,03%
current	4,41%	activ	5,33%	methodologi	6,56%	cost	3,99%
optim	4,08%	cost	5,00%	approach	5,72%	effici	3,47%
inform	3,92%	cycl	4,50%	research	4,71%	low	3,47%
effect	3,59%					emiss	3,30%
						transport	3,12%

TOPIC_19	Prob	TOPIC_42	Prob	TOPIC_8	Prob	TOPIC_21	Prob
organ	6,17%	product	6,62%	decis	6,93%	improv	7,53%
global	6,17%	design	4,54%	case	5,77%	firm	5,22%
exampl	5,05%	reduc	4,16%	sustain	5,00%	provid	5,22%
relat	4,68%	unit	3,78%	system	4,81%	posit	4,25%
		reduct	3,60%				

TOPIC_43	Prob	TOPIC_11	Prob	TOPIC_12	Prob	TOPIC_29	Prob
environment	6,57%	compani	9,99%	studi	10,47%	find	11,70%
perspect	4,25%	differ	6,19%	term	6,64%	relationship	6,05%
develop	3,67%	practice	5,99%	need	5,03%	increas	5,25%
perform	3,67%						
increasingli	3,29%						

TOPIC_14	Prob	TOPIC_34	Prob	TOPIC_16	Prob	TOPIC_1	Prob
perform	6,87%	depart	11,03%	busi	10,42%	servic	6,91%
research	6,47%	model	8,32%	resourc	6,26%	area	6,49%
empir	6,47%	suppli	7,07%	support	5,63%	import	5,65%
framework	5,86%					challeng	4,82%

TOPIC_20	Prob	TOPIC_22	Prob	TOPIC_3	Prob	TOPIC_7	Prob
consid	5,68%	competit	7,36%	logist	5,94%	univers	8,69%
result	4,63%	futur	7,15%	transport	4,88%	supplier	8,69%
help	4,00%	usa	4,63%	paper	4,67%	manufactur	5,30%
focus	3,37%	understand	3,58%	green	4,67%		
topic	3,37%			chain	4,24%		

TOPIC_27	Prob	TOPIC_26	Prob	TOPIC_37	Prob	TOPIC_45	Prob
base	7,84%	requir	6,13%	industri	8,14%	manag	8,80%
influenc	4,88%	discuss	5,04%	consider	5,06%	univers	6,82%

method	4,45%	complex	4,16%	mani	4,62%	develop	4,40%
comput	3,61%	strateg	3,51%	variabl	4,18%		
		produc	3,51%				

TOPIC_17	Prob	TOPIC_35	Prob	TOPIC_46	Prob	TOPIC_2	Prob
sustain	6,83%	respons	10,21%	close	5,77%	green	8,27%
larg	5,51%	due	4,00%	logist	5,10%	research	7,60%
suggest	5,07%	corpor	3,33%	examin	4,44%	busi	7,37%
build	4,41%	implic	2,89%	specif	3,55%		
				literatur	3,33%		

TOPIC_9	Prob	TOPIC_6	Prob	TOPIC_44	Prob	TOPIC_40	Prob
pollut	3,13%	technologi	7,05%	chain	10,76%	strategi	6,67%
control	3,13%	investig	4,10%	show	6,41%	environ	5,52%
appendix	2,68%	research	3,64%	concern	5,50%	aspect	5,06%
canada	2,68%	recent	3,19%			benefit	4,37%
western	2,68%	situat	2,96%				
plant	2,46%						

TOPIC_10	Prob	TOPIC_18	Prob	TOPIC_24	Prob	TOPIC_36	Prob
includ	6,52%	model	8,16%	suppli	9,59%	custom	8,24%
paper	5,59%	futur	5,13%	chain	7,95%	involv	7,30%
survei	5,59%	issu	4,67%	institut	7,48%	signific	6,83%
type	4,42%	contribut	4,43%				

TOPIC_23	Prob	TOPIC_32	Prob	TOPIC_15	Prob	TOPIC_5	Prob
measur	8,60%	network	8,12%	abstract	12,48%	product	8,90%
evalu	5,49%	field	4,54%	purpos	7,20%	consum	5,53%
account	4,30%	level	4,54%	aim	5,52%	util	3,85%
analysi	3,11%	valu	4,06%			market	3,37%

TOPIC_33	Prob	TOPIC_4	Prob	TOPIC_31	Prob	TOPIC_25	Prob
analysi	9,47%	becom	7,91%	question	3,76%	design	6,61%
interest	8,26%	issu	5,68%	project	3,51%	creat	5,59%
present	8,26%	limit	4,70%	school	3,26%	oper	4,57%
		point	4,70%	specif	2,76%	econom	4,32%
				basi	2,51%		
				cultur	2,26%		
				complet	2,26%		

TOPIC_38	Prob	TOPIC_39	Prob
suppli	7,84%	manag	14,34%
chain	7,58%	univers	9,93%
logist	6,27%		

Tabla 3.7. Tópicos ordenados y términos más significativos por tópico

La figura 3.8 ilustra un Treemap del top 5 de los tópicos más relevantes con sus términos más significativos.

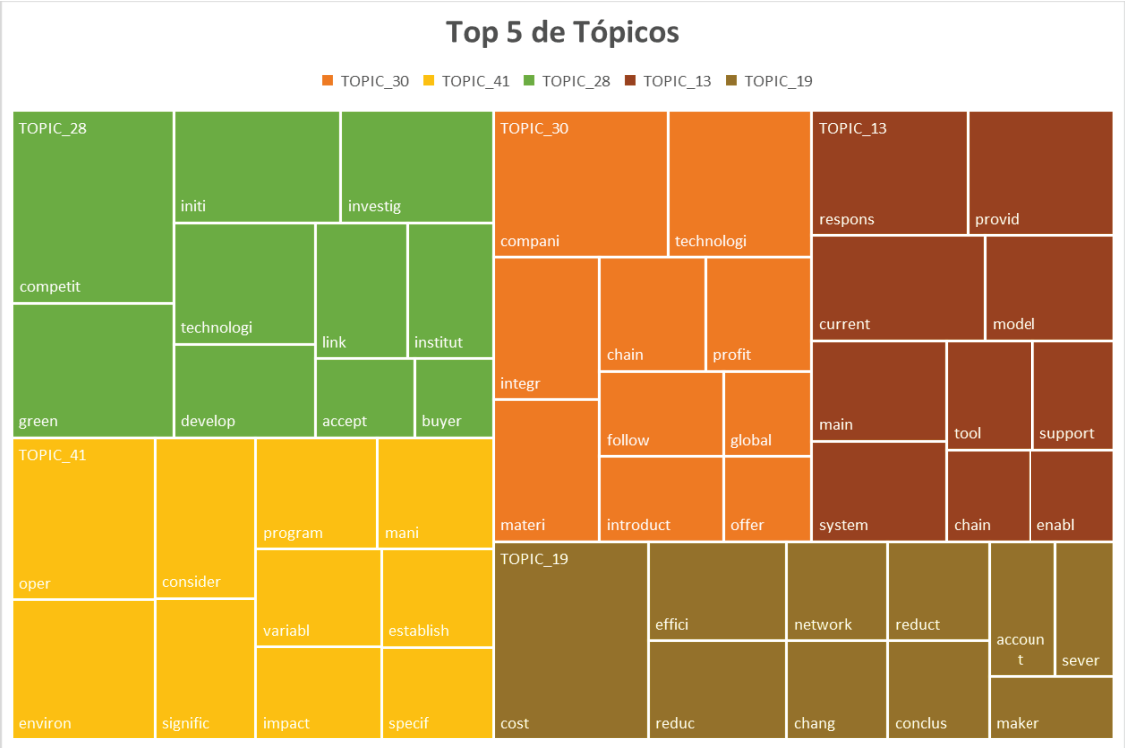


Figura 3.8. Treemap top 5 tópicos

Adicional a esto se analiza los tópicos y el porcentaje que estos ocupan dentro de los diferentes documentos del corpus, se señala que la selección de los tópicos más significativos se lo realizó aplicando el principio de Pareto, para lo cual se considera los tópicos cuya probabilidad sea próxima a 20%, ver la Tabla 3.8.

Documento	topico	% Tópico
1	V43	0,43
2	V43	0,33
5	V43	0,43
6	V5	0,33
7	V43	0,26
8	V45	0,43
9	V4	0,31
11	V43	0,23
12	V4	0,2
13	V43	0,21
15	V43	0,44
18	V17	0,2
18	V38	0,2

Documento	topico	% Tópico
58	V31	0,24
59	V11	0,37
60	V6	0,33
62	V18	0,24
64	V6	0,27
65	V30	0,27
65	V11	0,21
66	V30	0,23
67	V11	0,27
72	V15	0,47
76	V39	0,25
78	V6	0,21
80	V11	0,26

Documento	tópico	% Tópico
118	V2	0,23
120	V15	0,25
121	V42	0,2
123	V44	0,36
125	V29	0,31
127	V23	0,21
103	V22	0,44
133	V10	0,3
133	V43	0,21
139	V43	0,31
140	V30	0,36
142	V1	0,44
143	V3	0,28

26	V32	0,36
27	V40	0,37
28	V4	0,42
29	V27	0,49
31	V17	0,33
33	V41	0,49
36	V13	0,37
39	V13	0,31
39	V43	0,25
41	V6	0,41
42	V11	0,29
42	V33	0,26
43	V43	0,21
44	V43	0,21
47	V19	0,24
49	V11	0,39
52	V11	0,28
53	V23	0,29
54	V23	0,2
55	V36	0,33
56	V26	0,26
57	V19	0,53

83	V38	0,23
84	V29	0,38
85	V11	0,29
86	V29	0,33
90	V1	0,28
90	V39	0,24
91	V39	0,24
91	V15	0,23
93	V22	0,31
94	V12	0,34
95	V38	0,43
96	V38	0,55
98	V21	0,82
100	V24	0,26
102	V24	0,68
103	V6	0,33
104	V43	0,3
105	V7	0,23
107	V39	0,31
108	V43	0,33
111	V43	0,22
113	V43	0,2

145	V2	0,34
146	V16	0,4
146	V43	0,22
147	V43	0,35
147	V10	0,26
148	V9	0,36
148	V43	0,21
149	V43	0,3
149	V34	0,25
150	V9	0,33
150	V43	0,3
151	V2	0,24
153	V3	0,46
154	V10	0,34
155	V33	0,5
158	V6	0,34
161	V1	0,53
163	V35	0,91
164	V6	0,48
168	V33	0,47

Tabla 3.8. Probabilidad de tópico más significativo por documento

La figura 3.9 refleja la ocurrencia de los tópicos más significativos por documento.

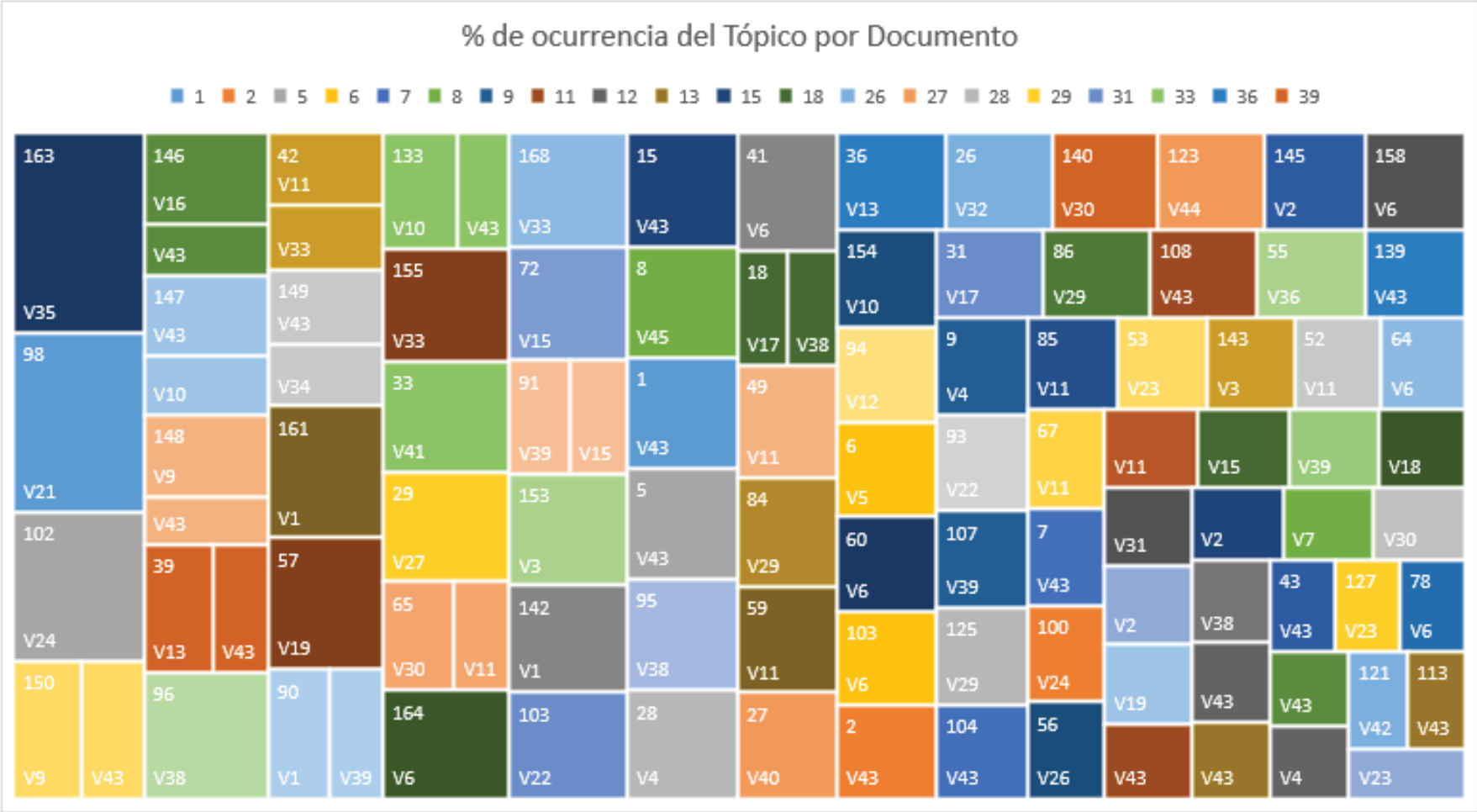


Figura 3.9 Treemap tópico por documento

A manera de resumen el top 10 de tópicos más latentes en el corpus se especifica en la tabla 3.9.

Tópico	Probabilidad
suppli-chain-sustain-manag-research	0,123470029
product-cost-design-remanufactur-increas	0,044038998
green-suppli-manag-chain-taiwan	0,036033705
logist-revers-return-program-effect	0,030245685
perform-practice-industri-improv-environment	0,030188477
environment-invest-manufactur-collabor-resourc	0,029367393
research-issu-perspect-valu-busi	0,029345783
manag-china-school-univers-dalian	0,028557273
emiss-carbon-logist-transport-cost	0,027468409
compani-issu-environment-sustain-custom	0,026987989

Tabla 3.9. Top 10 tópicos

Con esto se tiene que el tópico que se encuentra con mayor frecuencia en el corpus es el que tiene que ver con las palabras (suppli-chain-sustain-manag-research), ver la figura 3.10.

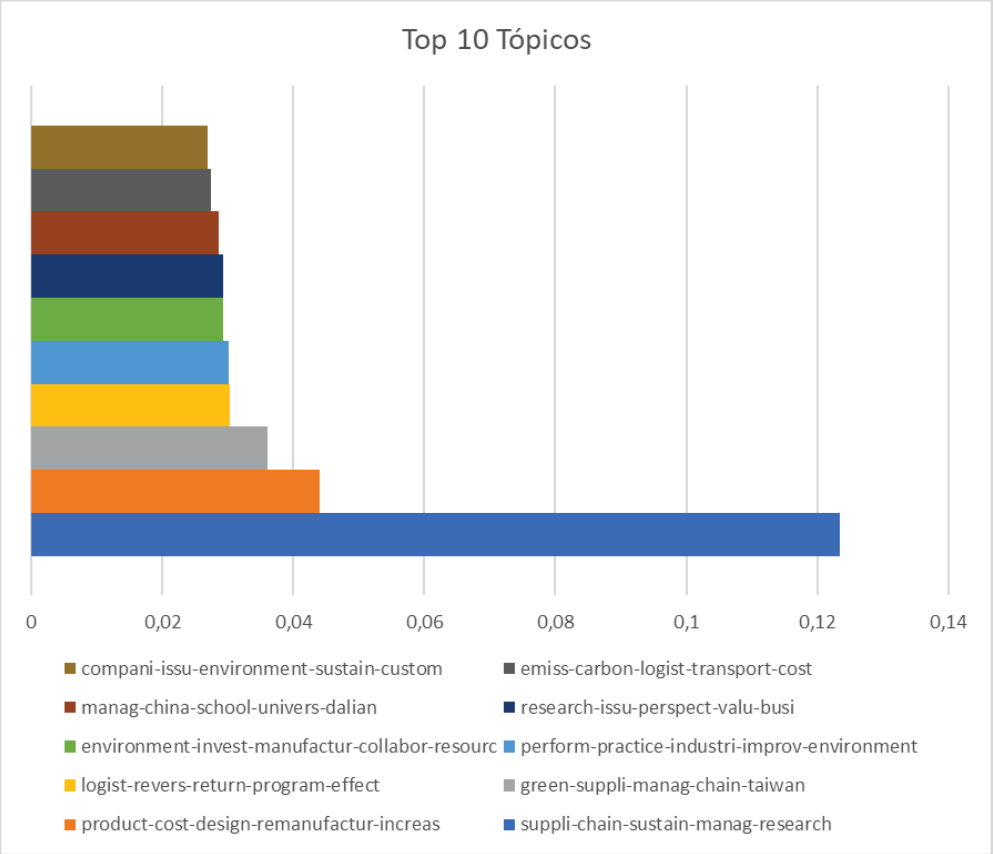


Figura 3.10. Gráfico de barras top 10 tópicos

La frecuencia con la que aparece un tópico como tema principal dentro de un documento, se denomina Rank-1, la tabla 3.10, muestra el Rank-1 de los tópicos más recurrentes dentro del corpus.

Tópico	Frecuencia
suppli-chain-sustain-manag-research	44
green-suppli-manag-chain-taiwan	13
product-cost-design-remanufactur-increas	11
emiss-carbon-logist-transport-cost	8
environment-invest-manufactur-collabor-resourc	8
research-issu-perspect-valu-busi	7
manag-china-school-univers-dalian	7
respons-social-datum-consum-report	5
univers-usa-manag-canada-state	4
transport-freight-measur-effect-road	3
logist-revers-return-program-effect	3
perform-decis-system-measur-oper	3
supplier-csr-area-sscm-practice	3
perform-practice-industri-improv-environment	3

Tabla 3.10. Tópicos principales

La tabla 3.11. muestra los tópicos menos frecuentes dentro del corpus.

Tópico	**Frecuencia**
cost-system-contain-factor-reusabl	1
scan-franc-propos-report-transport	1
close-loop-recoveri-current-econom	1
network-major-chang-assess-retail	1
profit-function-fashion-firm-govern	1
hospit-purchas-packag-survei-critic	1
util-box-properti-lotteri-attribut	1
give-level-wast-radioact-facil	1
environment-model-stage-activ-object	1
purchas-portfolio-polici-recent-definit	0
consider-govern-abstract-datum-mechan	0
competit-advantag-abstract-innov-view	0

Tabla 3.11. Tópicos menos significativos

A la implementación LDA, se adiciona LDA_COL, de lo cual se tiene que el top 10 de los tópicos más frecuentes viene dado por la tabla 3.12.

TOPIC	PROB
TOPIC_23	0,06416
TOPIC_1	0,04495
TOPIC_8	0,03973
TOPIC_42	0,03798
TOPIC_2	0,03768
TOPIC_17	0,03317
TOPIC_33	0,03056
TOPIC_3	0,02892
TOPIC_40	0,02876
TOPIC_13	0,02815

Tabla 3.12. Tópicos más Frecuentes LDA_COL

Una muestra de los tópicos usando LDA_COL se muestran en la tabla 4.13.

TOPIC_23	6,42%	TOPIC_1	4,50%
find	0,80%	analys_analysi_analyz	0,69%
find_firm	0,48%	wide	0,69%
abstract_academ_activ_add_address	0,40%	analys_analysi_analyt_appli_approach	0,57%
approach_appropri_aspect_attempt_barrier	0,40%	attent_australia_automot_base_becom	0,57%
approach_area_argu_avail_base	0,40%	basi_becom_benefit_centr_characteris	0,57%
approach_area_asia_asian_atmospher	0,40%	benchmark_benefit_busi_canada_competit	0,57%
approach_area_assess_assumpt_attract	0,40%	bui_call_carbon_chain_challeng	0,57%
approach_area_avail_base_benefit	0,40%	chain_chemic_china_chine_clark	0,57%
approach_articl_aspect_assess_audit	0,40%	close_clsc_commiss_conveni_cross	0,57%
approach_aspect_attribut_author_awar	0,40%	commun_compani_competit_conceptu_construct	0,57%

TOPIC_8	3,97%	TOPIC_42	3,80%
acknowledg_acquisit_activ_actual_addit	0,65%	need	1,35%
addit_aspect_assist_author_awar	0,65%	view	0,81%
approach_area_auburn_avail_aviat	0,65%	accept_account_analysi_analyt_anoth	0,68%
build_burden_busi_canada_chain	0,65%	account_achiev_addit_address_advantag	0,68%
chang_close_clsc_compani_competit	0,65%	assess_asset_associ_author_awar	0,68%
compani_condit_confront_cost_creat	0,65%	believ_busi_cambridg_capac_captur	0,68%
concern_consid_construct_consum_consumpt	0,65%	benefit_box_busi_carbon_care	0,68%
conflict_connect_consequ_consum_context	0,65%	bia_build_caus_chain_cheng	0,68%
consider_consum_core_corpor_creation	0,65%	busi_capabl_care_case_chain	0,68%
contribut_countri_creation_criterion_custom	0,65%	cluster_competit_counterpart_dalian_depart	0,68%

TOPIC_2	3,77%	TOPIC_17	3,32%
attent_attract_autom_base_canada	0,68%	base_china_clark_compani_condit	0,77%
avail_awar_balanc_bari_base	0,68%	becom_benefit_broad_build_busi	0,77%
awar_becom_benefit_busi_chain	0,68%	bia_compani_consid_countri_deal	0,77%
begin_bottom_build_busi_case	0,68%	brand_busi_case_chain_challeng	0,77%

carbon_carri_center_chaaban_chain	0,68%	certif_chain_chen_christian_chung	0,77%
chain_chang_channel_choic_close	0,68%	commerc_competit_conceptu_conduct_corpor	0,77%
challeng_chang_christian_chung_chungli	0,68%	committe_commun_compani_compar_concurr	0,77%
chang_compet_competit_comput_concern	0,68%	compani_complianc_concern_conclus_configur	0,77%
combin_complex_comput_concern_condit	0,68%	competit_continu_cost_cross_deliveri	0,77%
comput_consequ_consid_consumpt_cycl	0,68%	consum_decid_depart_diffus_equal	0,77%

Tabla 3.13. Tópicos más frecuentes usando LDA_COL

La tabla 3.13 muestra los tópicos más frecuentes tras usar el modelo LDA_COL, existen palabras que no se relacionan con otras, es el caso del tópico 23, 1, 42. Además se tiene los diferentes n-gramas que describen los tópicos, que ayuda a determinar la relación que existe entre las diferentes palabras del tópico.

4. CONCLUSIONES

- Se implementa LDA a un sistema de recuperación de información siguiendo una metodología genérica, sin embargo LDA es un modelo de clasificación de información generativo aplicable a diferentes entornos de minería de texto basados en corpus.

- LDA es modelo escalable que se ajusta al set de documentos que se analicen, ya que si se agregan nuevos archivos el valor de las variables aleatorias solo dependerán del número de tópicos resultado de los algoritmos de maximización y de minimización.

- La implementación de LDA_COL permite determinar la relación existente entre términos dentro de un mismo tópico, esto se consigue gracias a que LDA implementa la distribución Dirichlet para la relación de términos; pudiendo ser estos unigram o n-gramas.

- Se integró herramientas como R y Matlab en las cuales se implementaron técnicas de minería de texto tales como eliminación de stop words, stemming y lemmatización las cuales fueron seleccionadas como resultado de la investigación y prueba de la interacción de estas.

- La implementación del modelo de tópicos sobre el corpus del caso de estudio ayudó a determinar el contenido de los diferentes documentos, así como también la relación existente entre estos y la relevancia de ciertos tópicos.

- El éxito del modelo de tópicos está ligado a la selección de las variables, así como también de la calidad de los datos, por lo cual se debe indagar en diferentes técnicas de pre procesamiento de información tales como data cleaning, data transformation y data reduction.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] D. M. Blei, A. Y. Ng y M. I. Jordan, «Latent Dirichlet Allocation,» *Journal of Machine Learning Research 3,* vol. I, nº 2, pp. 993-1022, 2003.

[2] M. Vallez y R. Pedraza, «El Procesamiento del lenguaje natural en la recuperación de Información Textual y áreas afines,» *Anuario Digital sobre Documentación Digital y Comunicación Interactiva,* vol. V, nº 5, pp. 49-69, Mayo 2007.

[3] C. Montenegro, E. Loza-Aguirre y M. Segura-Marco, «Using probabilistic topic models to study orientation of sustainable supply chain research,» *Trends and Advances in Information Systems and Technologies ,* vol. 745, pp. 576-586, 2018.

[4] A.-h. Tan, «Text Mining: The state of the art and the challenges,» de *Knowledge Disocovery from Advanced Databases*, Singapur, 1999.

[5] R. Feldman y J. Sanger, «Defining Text Minig,» de *The next minig handbook*, New York, Cambridge, 2006, pp. 1-2.

[6] T. Griffiths y M. Steyvers, «Probabilistic Topic Models,» *Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum,* vol. I, pp. 2-15, 26 9 2005.

[7] C. Manning y H. Schütze, Foundations of Statistical Natural Language Processing, Cambridge: MIT, 1999.

[8] A. Riddell, «How to Read 22,198 Journal Articles: Studying the History of German Studies with Topic Models,» *Rochester, New York: Camden House,* pp. 91-114, 22 Julio 2012.

[9] R. Alghamdi y K. Alfaqui, «A Survey of Topic Modeling in Text Mining,» *International Journal of Advanced Computer Science and Applications,* vol. 6, nº 1, pp. 147-153, 2015.

[10] P. Valdiviezo y A. Hernando, «Una comprensiva revisión de los métodos de recomendación basados en técnicas probabilísticas,» *International Journal of Information an Software Enginnering for Big Companies,* vol. I, nº 3, pp. 65-74, 2016.

[11] D. Blei, «Probabilistic Topic Models,» *Comunications of the ACM,* pp. 77-84, Abril 2012.

[12] T. Griffiths y M. Steyvers, «Finding Scientific Topics,» *Topics in Semantic Representation. Psychological Review,* nº 2, pp. 5228-5235, 2004.

[13] X. Wang, A. McCallum y X. Wei, «Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval,» de *Data Mining, 2007. ICDM 2007. Seventh IEEE International* , 2007.

[14] A. McCallum, A. Corrada-Emanuel y X. Wang, «Topic and Role Discovery in Social Networks,» *Artificial Intelligence Research,* vol. 30, nº 30, pp. 249-272, 2007.

[15] V. Ramanathan y H. Wechsler, «Phishing Website Detection Using Latent Dirichlet Allocation and AdaBoost,» de *International Conference on Intelligence and Security*, Piscataway, 2012.

[16] RStudio, «R Studio,» R Studio, [En línea]. Available: https://www.rstudio.com/products/rstudio/. [Último acceso: 15 8 2018].

[17] L. Carvajal, Metodología de la Investgación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.

[18] M. Dalal y M. Zaveri, «Automatic Text Classification: A Thechnical Review,» *International Journal of Computer Applications,* vol. XXVIII, nº 2, pp. 975-8887, 2011.

[19] S. Vijayarani, J. Ilamathi y Nithya, «Preprocessing Techniques for Text Mining - An Overview,» *International Journal of Computer Science & Communication Networks,* vol. 5, nº 1, pp. 7-16, 2014.

[20] M. F. Porter, «An algorithm for suffix stripping,» *Program,* vol. 14, nº 3, pp. 130-137, 1980.

[21] G. J. Anjali, «A Comparative Study of Stemming Algorithms,» *Computer Tech Appl,* vol. 2, nº 6, pp. 1930-1938, 2011.

[22] J. Xu, «NanoNets,» Mefium, 25 Mayo 2016. [En línea]. Available: https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05. [Último acceso: 21 Julio 27].

[23] D. P, L. N y R. B, «Maximum Likelihood from Incomplete Data via the EM Algorithm,» *Royal Statistical Society,* vol. 39, nº 1, pp. 1-38, 1997.

[24] N. Murzintcev, «Cran.R,» R.ORG, 24 Octubre 2016. [En línea]. Available: https://cran.r-project.org/web/packages/ldatuning/vignettes/topics.html. [Último acceso: 26 Julio 2018].