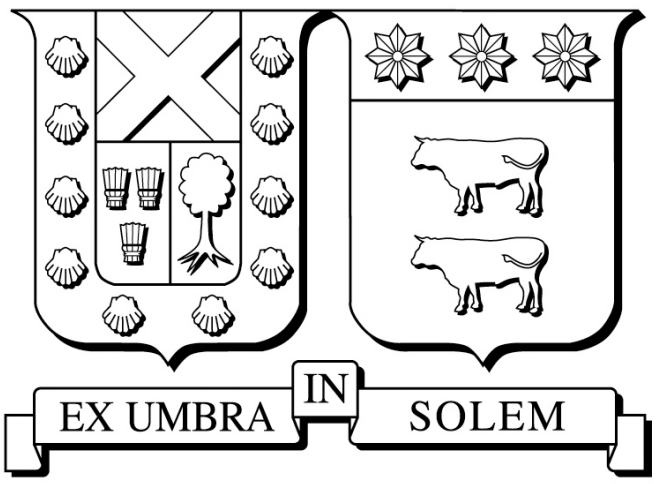
Universidad Técnica Federico Santa María
Departamento de Informática



Topic Models Ensembles

DISERTACIÓN

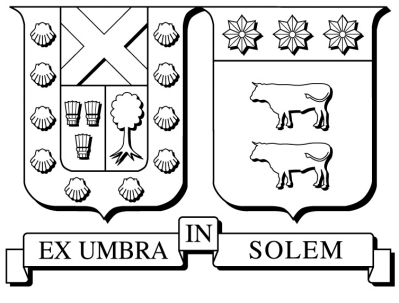
Enviado en cumplimiento parcial de los requisitos
para el grado de

DOCTOR EN INGENIERÍA INFORMÁTICA

por

Pablo Ivan Ormeño Arriagada

Valparaíso, Chile
Enero 2022.



Comité Examinador

Prof. Dr. Marcelo Mendoza Universidad Técnica Federico Santa María Chile	Supervisor de Tesis
Prof. Dr. Carlos Valle Universidad de Playa Ancha de Ciencias de la Educación Chile	Co-Supervisor de Tesis
Prof. Dr. Claudio Torres Universidad Técnica Federico Santa María Chile	Correferente Interno
Prof. Dr. Roberto Gonzalez-Ibañez Universidad de Santiago de Chile Chile	Correferente Externo Nacional
Prof. Dr. Manuel Montes y Gómez Instituto Nacional de Astrofísica, Óptica y Electrónica México	Correferente Externo Internacional
Prof. Dr. Mauricio Solar Universidad Técnica Federico Santa María México	Presidente de la Comisión

Para Florencia...

Resumen

La recuperación de información Adhoc es una tarea desafiante que consiste en hacer ranking de documentos para consultas provenientes desde un enfoque de bolsa de palabras. Los métodos clásicos basados en consultas y documentos de vectores de texto, usan funciones de ponderación de términos para hacer ránking de documentos. Algunos de las limitaciones de estos métodos son que no pueden lidiar con conceptos polisémicos. Además, introducen falsas ortogonalidades entre palabras semánticamente relacionadas. Para superarlas, los enfoques de recuperación de información basados en modelos de temas se pueden explorar. Específicamente, los modelos de temas basados en Latent Dirichlet Allocation (LDA) permiten construir representaciones de documentos de texto en el espacio latente de temas, que modela de mejor manera la polisemia y evitan la generación de representaciones ortogonales entre términos relacionados. Es por esto que se pueden expandir las estrategias de Recuperación basadas en LDA usando estrategias de Aprendizaje de Ensamblado. En este sentido, la selección de modelos obedece a estos paradigmas, por lo que probamos dos enfoques usados exitosamente en el aprendizaje supervisado. Se estudian las técnicas Boosting y Bagging para modelos de temas, usando cada modelo como un experto débil de recuperación. Finalmente, se mezclan las listas de ranking obtenidas de cada modelo usando un enfoque simple pero efectivo de fusión de listas top-k. Se muestra que el enfoque propuesto fortalece los resultados en precisión y en recall, superando a los modelos clásicos de recuperación y las líneas bases de modelos de temas.

Palabras Claves: *Recuperación de Información Ad-hoc, Latent Dirichlet Allocation (LDA), Bagging, Boosting*

Abstrct

Adhoc Information Retrieval is a challenging task that consists in performing document ranking from a bag-of-words approach. The classic methods are based on text vectors of queries and documents, using weighted-term functions to build these rankings. Some of the limitations of these methods are, for example, that they can not deal with polysemic concepts. Besides, they introduce fake originalities between semantically related words. To overcome this, information retrieval approaches based on topic models can be explored. Specifically, topic models based in Latent Dirichlet Allocation (LDA) allow to build documents representations in the topic latent space and improve the polysemic models and avoid orthogonal representations between related terms. Because of this, we can expand the retrieval strategies with LDA using Ensemble Learning. In this way, the model's selection obeys those paradigms and because of this, we try two approaches used in supervised learning. Bagging and Boosting are studied for topic modeling, using every model as a weak retriever. Finally, the documents lists are mixed from every model using a simple but effective fusion of top k lists. It is shown that our approach improves precision and recall.

Key Words: *Adhoc Information Retrieval, Latent Dirichlet Allocation (LDA), Bagging, Boosting*

Índice general

Índice de figuras

Capítulo 0

Aportes de esta Tesis

De este trabajo de tesis se obtuvieron 3 publicaciones:

- **Boosting Text Clustering using Topic Selection**
  Mendoza, M; Ormeno, P; Valle, C.;
  IET Conference Proceedings; **2018**.

- **Ad-hoc Information Retrieval based on Boosted Latent Dirichlet Allocated Topics**
  Mendoza, M; Ormeno, P; Valle, C.;
  37th International Conference of the Chilean; **2018**.

- **Topic Models Ensembles for AD-HOC Information Retrieval**
  Ormeno, P; Mendoza, M; Valle, C.;
  Information; **2021**.

Capítulo 1

Introduction

En esta introducción presentaremos el alcance, los objetivos, la hipótesis y la organización de la tesis.

1.1. Alcance de la investigación

El alcance específico de este trabajo es el diseño de algoritmos nuevos para la tarea de Recuperación de Información. El Aprendizaje con Ensamblados es una subárea del Aprendizaje Automático, en el cuál múltiples métodos se combinan para resolver un problema [**Pol06**]. En otras palabras, los ensamblados combinan un conjunto de modelos para formar un metamodelo mejor.

Por otro lado, la Recuperación de Información es la disciplina que se encarga de obtener recursos que son relevantes para una necesidad de información. En este sentido existen varios métodos que realizan esta tarea [**DB75**]. Un área de la Recuperación de Información, son los Modelos de Temas, los cuales se encargan de encontrar la estructura latente de temas de una colección documental. Luego, utilizando esto y la información a priori del vocabulario que compone esta colección, podemos construir un ranking de documentos que se obtiene a partir de una consulta dada.

1.2. Objetivos de la Investigación

El principal objetivo de esta investigación es la formulación de un método de Ensamblados de Modelos de Temas para la tarea de Recuperación de Información Adhoc. Los objetivos específicos son:

- **O1:** Realizar una revisión bibliográfica de los Ensamblados de Modelos de Temas y de los métodos de Recuperación de Información.
- **O2:** Desarrollar un marco de trabajo teórico para construir modelos de Ensamblado de Temas.
- **O3:** Construir una familia de ensamblados que permitan realizar la tarea de recuperación de información usando modelos de temas.
- **O4:** Aplicar los algoritmos a colecciones documentales reales que se usen en la literatura.

1.3. Hipótesis de Investigación

El aprendizaje con ensamblado tiene como principal característica la forma en la que se crean modelos y estos se van agregando. Lo interesante de esto es que con un diseño apropiado, el desempeño que se espera al combinar estos modelos, debería ser mejor que los modelos individuales. El concepto principal que nos permite afirmar esto es el concepto de diversidad [**BWHY05**] entre conjunto de modelos. Esta diversidad de un ensamblado puede ser logrado entrenando cada máquina con un diferente conjunto de datos o al modificar los parámetros de los modelos o al modificar la forma de fusionar las salidas [**Kun14a**], [**Die00**].

Continuando con lo anterior, en la Recuperación de Información, los datos son colecciones de documentos que son procesados por las técnicas. Dentro de estos métodos, los Modelos de Temas, son los utilizados para algunas de las tareas existentes. Estos modelos [**LTD+16**] [**BB17**] tienen un conjunto de parámetros que hacen que las diferentes salidas dependan de los que usemos como entrada.

Finalmente, con estos modelos de temas, podemos realizar la recuperación de estos documentos dado un conjunto de consultas [**ZL01**]. La principal hipótesis de investigación de este trabajo es:

Es posible mezclar modelos de temas, usando ensamblados, para obtener un mejor desempeño en la tarea de recuperación de información Adhoc, que el que se puede obtener con los modelos individuales.

1.4. Organización de la Tesis

Se pasa a esbozar la organización de esta tesis. En el Capítulo 2 se presentan las bases fundamentales de la *Recuperación de Información*. En el Capítulo 3 se introducen los Modelos de Temas. En el Capítulo 4 presentamos los Métodos de Ensamblados. En el Capítulo 5 se presenta el Ensamblado de Modelos de Temas para la tarea de ranking de documentos, utilizando tres variaciones. En la Sección 5.1 se ve la explicación metodológica de la propuesta,

explicando cada etapa de esta. En el Capítulo 6 se presenta y discute los resultados experimentales obtenidos en varios datasets. Esta tesis concluye en el Capítulo 7 resumiendo las principales conclusiones y contribuciones de esta investigación. Adicionalmente se sugieren algunas direcciones de investigaciones futuras relacionadas.

Capítulo 2

Recuperación de Información

La Recuperación de Información *es la ciencia de la búsqueda por información dentro de bases de datos relacionales, documentos, texto, archivos multimedia, e internet* [**DB75**]. Las aplicaciones de Recuperación de Información son diversas, ellas incluyen la extracción de información desde grandes cantidades de documentos, la búsqueda en librerías digitales, filtros de información, filtros de spam, extracción de objetos desde imágenes, resúmenes automáticos, clasificación de documentos, clustering y búsqueda web.

La idea de buscar información fue mencionada por primera vez por Vannevar Bush en 1945 [**SG01**]. Para 1990 se habían llevado a cabo diferentes técnicas, las cuales procesaban no más de unos miles de documentos [**SG01**]. La irrupción de internet y los motores de búsqueda web han obligado a los científicos y a las empresas a crear sistemas de recuperación a gran escala para seguirle el ritmo al crecimiento exponencial de los datos online.

En este capítulo revisaremos un resumen de las técnicas más importantes de Recuperación de Información. Entre ellas están los Modelos Booleanos, los Modelos de Espacios de Vectores, los Modelos de Lenguaje y los Modelos Probabilísticos.

Luego continuamos con una tarea muy importante, llamada Recuperación AdHoc, en la cual se basa esta investigación. Continuamos, luego con las principales métricas de evaluación en la disciplina.

Finalizamos con la explicación de Ranking de Documentos, el que sirve como método principal, para recuperar documentos ordenados de acuerdo a cierta relevancia (cuánto satisface el documento la necesidad de información del usuario).

2.1. Recuperación de Información

Las estrategias de Recuperación de Información transforman el documento en representaciones adecuadas para recuperarlo eficientemente desde una colección. Cada una de estas

estrategias integra modelos específicos para el proceso de representación [**JR10**]. En la Figura 2.1 puede verse la arquitectura general de un sistema de recuperación de información. El usuario primero realiza la consulta, que se ejecuta sobre el sistema. Luego, se consulta a una base de datos de la colección de documentos o *corpus* y se retorna el documento o los documentos coincidentes.

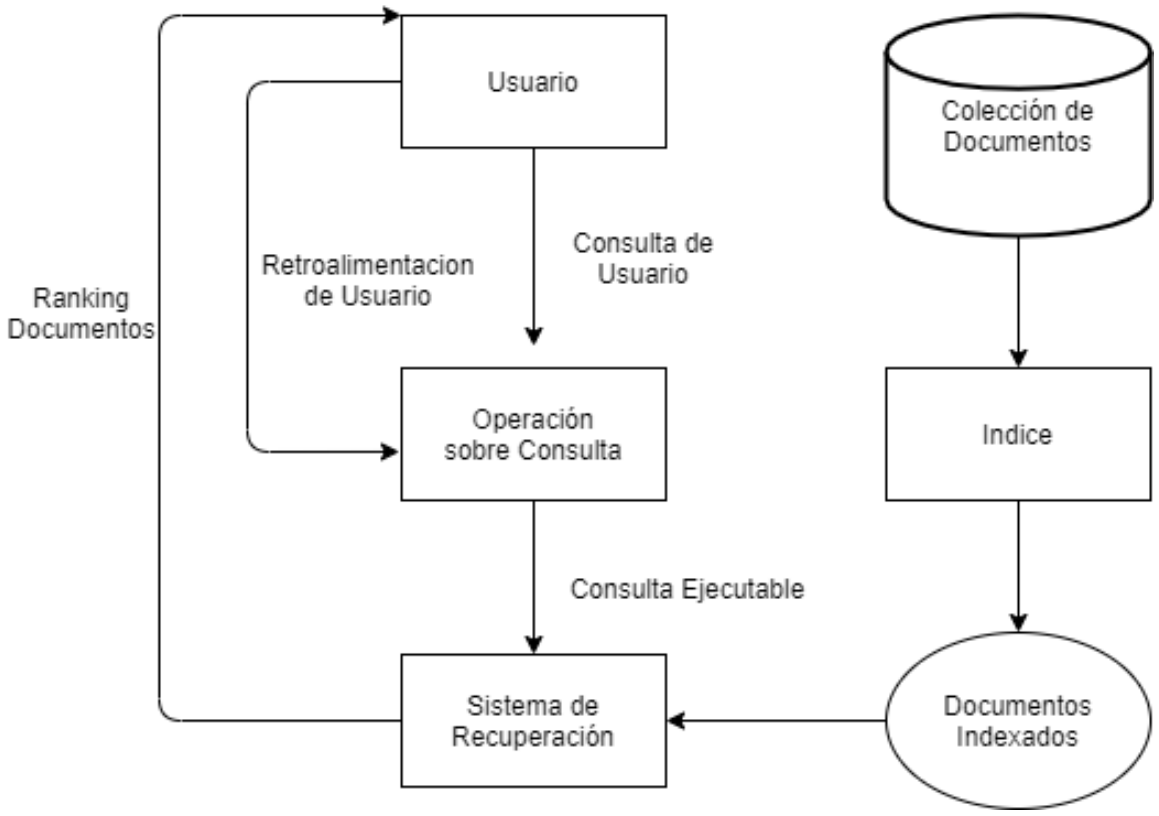


FIGURA 2.1. Arquitectura General Sistema de Recuperación de Información

Podemos ver en [**SVL14**] que se explica el hecho de que los modelos de Recuperación de Información regulan la manera de como un documento y una consulta (en inglés *query*) se representan de forma de saber que tan relevante es este documento para el usuario. Existen cuatro tipos de modelos de Recuperación de Información principales:

- Modelos Booleanos
- Modelos de Espacios de Vectores
- Modelos de Lenguaje
- Modelos Probabilísticos

Los modelos más usados en sistemas de Recuperación de Información y en Sistemas Web son los primeros tres. Aunque estos modelos representan documentos y consultas de manera diferente, ellos utilizan el mismo marco de trabajo. Estos modelos tratan a los documentos y las consultas como una bolsa de palabras o términos. La secuencia de términos y su posición en una secuencia o documento se ignora. De esta forma, un documento se define como un conjunto de términos distintivos. Un término es una palabra cuya semántica ayuda a recordar el tema principal del documento.

2.1.1. Modelos Booleanos Estándar El Modelo Booleano Estándar usa la noción de calce (en inglés *match*) exacto para poder igualar los documentos con la consulta del usuario. Ambos casos, tanto la consulta como la recuperación, se basan en el álgebra booleana. En este modelo, los documentos y las consultas, se representan como un conjunto de términos. Esto quiere decir que cada término se considera sólo presente o ausente en cada documento. De acuerdo a [**BYRnM$^+$99**], este modelo se basa en la lógica booleana y la teoría clásica de conjuntos en la que el documento buscado y la consulta se entienden como un conjunto de términos. La Recuperación se basa en si el documento tiene o no los términos que existen en la consulta. Podemos ver en [**EAA15**] una aplicación de modelos booleanos usados en Web Semántica.

2.1.2. Modelo Booleano Extendido En [**SBYM12**] se discute que los Modelos Booleanos no consideran el peso de los términos de las consultas. Este tipo de dificultades nos llevan al desarrollo de un Modelo Booleano Extendido. El objetivo de este modelo es el de usar el calce parcial y los pesos de los términos, como se hace en el Modelo de Espacio de Vectores [**SHG$^+$14**]. Conceptualmente, este modelo combina también características del álgebra Booleana. Adicionalmente, estos modelos obtienen la similaridad entre consultas y los documentos. De este modo, un documento puede ser ligeramente relevante si calza con alguno de los términos consultados y será retornado como resultado, mientras que el Modelo Booleano Estándar esto no ocurre. En [**SVL14**] se discuten los inconvenientes de este tipo de propuesta, como por ejemplo, que no consideran la relevancia de los documentos.

Existen algunas aproximaciones que trabajan con el concepto de Lógica Difusa aplicada a documentos de texto, como por ejemplo en [**KC17**], [**IO16**], [**BYRN11**].

2.1.3. Modelos de Espacios de Vectores El Modelo de Espacio de Vectores es un modelo algebraico que representa la información textual como un vector [**TP10**]. En estos modelos, luego del preprocesamiento requerido, se extrae un diccionario de términos (vocabulario) de cada documento, el que se compara con todos los documentos del corpus. En este enfoque, los documentos y las consultas se representan como vectores en un espacio multidimensional, donde las dimensiones de este espacio son las palabras usadas para construir el índice que representa los documentos. Podemos ver aplicaciones en esta área relacionadas con Recuperación de Información, Indexación y Rankings de Relevancia, en Motores de Búsqueda Web [**SD12**]

Idealmente, este modelo representa la importancia de una palabra usando la métrica *Term Frecuency Inverse Document Frequency (TFIDF)*. Primero, se calcula *Inverse Document Frecuency*, $idf(t)$, el cual enfatiza que un término que esté presente en casi todo el corpus no es tan bueno [**SD12**]. Finalmente, el producto, es decir, **TF x IDF**, como puede verse en la Ecuación 2.1, es la métrica para el término i en el documento j. Luego se calcula la similaridad entre documentos usando la Similaridad de Coseno.

$$\text{TF-IDF(i,j)} = TF \times IDF. \tag{2.1}$$

$$\text{TF(i,j)} = \frac{numero\ de\ veces\ que\ el\ termino\ i\ esta\ en\ el\ documento\ j}{numero\ total\ de\ terminos\ en\ el\ documento\ j}. \tag{2.2}$$

$$\text{IDF(i)} = \frac{numero\ total\ de\ documentos}{numero\ total\ de\ documentos\ que\ contienen\ el\ termino\ i}. \tag{2.3}$$

En [**KZS+15**] se explica que los métodos en el Modelo de Espacio de Vectores dan buenos resultados cuando se usan con enfoques para hacer ranking de documentos, como Latent Semantic Indexing. Como se explica en [**PGCB13**], el Modelo de Espacio de Vectores extrae el conocimiento automáticamente dado un corpus, por lo que se requiere mucho menos trabajo que en otros enfoques semántico, como los basados en conocimiento codificado a mano o las ontologías. El principal inconveniente del Modelo de Espacio de Vectores, como se nota en [**DCT12**], es el que los documentos están pobremente representados, ya que ellos tienen bajos valores de similaridad, en forma de un pequeño valor escalar y una gran dimensionalidad.

Existen extensiones al modelo de espacio de vectores donde se busca resolver algunos de los inconvenientes que pudiese tener. En [**WES15**] se busca establecer la similaridad entre el documento y la consulta, a través de una modificación en la forma de indexar. Por otro lado, en [**UR15**] se muestran enfoques que mezclan Procesamiento de Lenguaje Natural y Aprendizaje de Máquina. Finalmente, en [**TWJS14**], se menciona que no se aprovecha la ventaja de las relaciones entre los conceptos representados en los documentos.

Por otro lado en [**SLSB12**] se busca enfrentar el problema del spam como fuente de amenazas. Además en [**DWV99**], se muestran algunas soluciones usando Aprendizaje de Máquina, donde los modelos son entrenados con representaciones estadísticas de los términos en los mails. Finalmente, en [**KLMH13**] se explora el uso de detección de semánticas internas en los filtros de spam, usando los Modelos de Espacio de Vectores basado en Temas.

2.1.4. Indexación Semántica Latente En [**JWY+18**] se discute que básicamente la información es recuperada por el calce exacto de términos en los documentos con los de la consulta. Sin embargo, estos métodos de calce a través del vocabulario son imprecisos cuando se emplean para calzar la consulta de un usuario. Esto se debe al hecho de que, generalmente, hay muchas formas para expresar un concepto dado, por el uso de sinónimos. En algunas circunstancias, los términos literales en una consulta de usuario pueden no coincidir con los de los documentos relevantes. Adicionalmente, muchas frases y palabras tienen múltiples significados, lo que se conoce como polisemia. Esto significa que los términos en una consulta de usuario, literalmente, calzan términos en documentos irrelevantes.

De acuerdo a [**Ble11**] un enfoque mejorado permitiría que los usuarios recuperen información en las bases de un tema conceptual o significado de un documento. Es esto donde Indexación Semántica Latente se vuelve útil. Esta es una técnica en Procesamiento en Lenguaje Natural para analizar relaciones entre un conjunto de documentos y los términos que contienen, al producir un conjunto de elementos que los relaciona.

Este enfoque ha probado ser efectivo en categorización de contenido en conceptos predefinidos o temas [**SLM09**]. Idealmente, los conceptos contenidos en los documentos que serán categorizados se comparan con los conceptos contenidos con los items de ejemplo y una

categoría (o categorías) se asignan a los documentos basado en las similaridades entre los conceptos que contienen y los conceptos contenidos en los documentos de ejemplo.

2.1.5. Otras técnicas Existen muchas otras técnicas que se utilizan en Recuperación de Información para resolver los problemas y tareas que existen. Por ejemplo, en [**BCC10**] se habla de los métodos de independencia binaria, donde los documentos se tratan como vectores binarios, es decir hay o no hay término. En este sentido, se sugiere en [**MRS08**] que las consultas se representen de manera similar, donde los términos se consideran independientes. En [**BCC10**] se considera este supuesto un poco restrictivo.

2.1.6. Modelo de Relevancia Probabilística El último objetivo de un modelo de recuperación es medir el grado de relevancia de un documento respecto a una consulta dada. Los modelos probabilísticos se usan ampliamente para medir la verosimilitud de la relevancia de un documento al combinar la frecuencia término y la especificidad de términos en una manera formal como puede verse en [**Pai13**]. Recientes investigaciones muestran que la normalización de la frecuencia de los términos, donde se realza la importancia del término, es un esquema efectivo. Sin embargo, los modelos existentes no permiten utilizar completamente estos componentes de normalización **tf** de una manera óptima. Además, muchos de los modelos del estado del arte ignoran la distribución de un término en la parte de la colección que contiene el término. En [**CPL15**] se introduce un modelo probabilistico donde la relevancia de un documento aumenta con la frecuencia del término de la consulta. El mérito del modelo propuesto fue demostrado en un gran número de colecciones de sitios web. En [**MRS08**] los modelos probabilisticos determinan que la relevancia entre la consulta y el documento requerido se determina mediante la probabilidad de este. Sin embargo, en [**BYRN11**] se señala que existen varios inconvenientes que son inherente en el modelo probabilístico: esto no categoriza los documentos basado en relevancia. Desafortunadamente, la probabilidad de que estos documentos recuperados sean relevantes para la consulta no se calcula, lo que es una debilidad de los modelos probabilísticos. Finalmente, otra debilidad en los modelos probabilísticos de acuerdo a [**KJSR15**] es como estos modelos tienen conjuntos de relevancia de documentos, al asumir los pesos de las relevancias con datos binarios, lo cual no se condice con la frecuencia de los términos indexados, que aparecen en cada documento.

2.1.7. Modelo de Inferencia Estadística Para terminar, existen algunos modelos que permiten trabajar con incerteza. Por ejemplo, en [**WH13**], se trabaja con consultas con forma de afirmaciones sobre los documentos deseados y verificar la veracidad de la consulta, y si lo es, el documento se recupera. En [**Jr.13**], el principal desafío es que el contenido del documento puede no ser suficiente para la consulta, por lo que se utiliza conocimiento adicional y reglas para realizar la inferencia. Hay otros modelos como [**KJSR15**], donde se mezcla Inteligencia Artificial con técnicas de Procesamiento de Lenguaje Natural. En este caso, los modelos de lenguaje tienen varias aplicaciones, como reconocimiento de discurso, generación de texto y

en traducción de máquinas. Podemos entender que un modelo de lenguaje es una secuencia de palabras representadas como distribución de probabilidad.

Por otro lado podemos ver en [**ZJX+15**], [**SNS15**], [**MDK+11**] representaciones modelos basados conteos de palabras u oraciones utilizando Aprendizaje Automático. También se trata el concepto de independencia entre oraciones en [**XZL11**]. Es importante señalar que todos estos modelos buscan satisfacer los requerimientos de información de los usuarios, los que interactuan con sistemas semiautomáticos que contienen datos estructurados y no estructurados.

Podemos ver en [**Zei12**], aplicaciones en Recuperación Contextual, donde se combina conocimiento sobre la consulta y el contexto en que el usuario la aplica, para mejorar la recuperación. Esto se combina con algoritmos de clasificación que usan retroalimentación de los usuarios con evidencia contextual e histórica.

Por último, Latent Dirichlet Allocation es un modelo generativo estadístico que permite que datos observados sean explicados por datos no observados, que conectan los datos para darles una explicación [**Ama09**]. En este enfoque, cada documento, se trata como una mezcla de varios temas. También se asume que hay k temas subyacentes desde donde los documentos son generados y que cada tema se representa como una distribución multinomial sobre V palabras de un vocabulario. Veremos en más detalle este tipo de modelos en el siguiente capítulo.

2.2. Recuperación AD HOC

Esta forma de Recuperación es la tarea de encontrar documentos desde una gran colección, que son relevantes para una consulta de usuario. La consulta puede ser expresada como un conjunto de palabras claves o como una descripción en lenguaje natural. La Figura 2.2 nos muestra una simple consulta de TREC [**HL93**], [**ACD12**], [**Voo04**], [**Jon00**], preguntando sobre Enfermedades sobre Pérdida de Cabello. El nivel de detalle lo dan los campos *title*, *desc*, *narr*, como puede verse a continuación:

```
<num> Number: 508
<title> hair loss is a symptom of what diseases
<desc> Description:
Find diseases for which hair loss is a symptom.
<narr> Narrative:
A document is relevant if it positively connects the loss of
head hair in humans with a specific disease. In this context,
"thinning hair" and "hair loss" are synonymous. Loss of body
and/or facial hair is irrelevant, as is hair loss caused by drug
therapy.
```

FIGURA 2.2. Muestra de una consulta TREC

Esta tarea, donde el sistema retorna un documento completo textual como su salida, se llama técnicamente Recuperación de Documento; es uno de los mejores y más conocidos subtipos de Recuperación de Información. Existen otras tareas como la Recuperación de Speech, de Video, de Música y Recuperación de Pasajes de Texto [**SORN17**], [**Cal94**].

En Recuperación de Información, *Ad hoc* [**KMI18**] significa recuperación de *una vez* (basado en batch) de documentos que son relevantes para una consulta, la que se ejecuta una vez y no puede ser refinada. Esto es opuesto a la Búsqueda Interactiva (donde el usuario puede refinar la consulta iterativamente), o al Filtrado, donde la definición de relevancia puede cambiar con el tiempo, después de que los primeros documentos relevantes han sido vistos. *Recuperación Ad hoc* también asume una colección de documentos fija, en oposición a una recuperación que es dinámica, como la de la web. Es por lo tanto la más simple y más clara definición de la tarea de búsqueda de documentos y también la forma más estudiada de IR.

Existen conjuntos de evaluación modernos, como TREC, donde las consultas son generadas por humanos y formuladas en lenguaje natural; luego un juez decide cuales documentos, dado un conjunto finito de ellos, son relevantes a la consulta, en una tarea llamada Decisión de Relevancia. Estas decisiones se usan como fijas, un *Gold Standard* a priori, en el cual cada resultado del sistema es comparado con el desempeño de evaluaciones de laboratorio que son fáciles de controlar, esto es debido a que los factores alrededor del principal problema de evaluación de recuperación de información, es decir, la subjetividad de la decisión de relevancia, es mantenida constante y así lo mas controlable posible. La colección completa de test consiste solo de tres componentes: **las consultas, las decisiones de relevancia y los conjuntos de documentos**. La ventaja de esta configuración es la repitabilidad de los experimentos bajo ciertas condiciones. Por ejemplo, TREC produce 50 consultas y juicios de relevancia sin tener que recrear las condiciones exactas de jueces humanos y su interacción con un sistema de IR.

Finalmente, esta configuración puede modificar sólo un subconjunto de parámetros a la vez en un sistema de recuperación, los que llamaremos parámetros de sistema. Estos parámetros de prueba son los siguientes: (a) la forma en que se indexan los documentos (asignación de palabras claves que describen mejor el documento); (b) el lenguaje de consulta usado (las

forma en que se combinan estas palabras claves); (c) el algoritmo de recuperación usado (como igualar los términos de la consulta con los indexados y cómo calcular este ajuste).

En la siguiente sección revisaremos las métricas de evaluación más usadas para la recuperación de información adhoc.

2.3. Métricas de Evaluación

Dentro de las métricas más utilizadas en recuperación de información adhoc, están la precision, el recall, accuracy, la medida F y mean Average Precision (mAP), las que revisaremos a continuación.

2.3.1. Recall, Precision y Accuracy Dada una colección de documentos de prueba, las principales métricas usadas para evaluar IR son precision y recall [**AKV16**], y hay algunas métricas de resumen que se derivan de estas métricas punto a punto. Revisemos la Figura 2.3, donde se definen las categorías de precision, recall y accuracy. Los que es relevante y no relevante se decide por un juez humano (o nuestra definición de verdad, la que se llama también Gold Standard) y lo que se recupera y no se recupera lo decide el sistema.

	Relevante	No Relevante	Total
Recuperado	A	B	A+B
No Recuperado	C	D	C+D
Total	A+C	B+D	A+B+C+D

Figura 2.3. Matriz de Confusión Precision Recall

- **Recall:** se define como la proporción de elementos recuperados entre los documentos relevantes:

$$\frac{A}{A+C}. \tag{2.4}$$

- **Precision:** se define como la proporción de elementos relevantes entre todos los elementos que fueron recuperados:

$$\frac{A}{A+B}. \tag{2.5}$$

- **Accuracy:** se define como la proporción de elementos correctamente recuperados, ya sea como relevantes e irrelevantes:

$$\frac{(A+D)}{(A+B+C+D)}. \tag{2.6}$$

Aunque Recuperación de Información en un principio puede verse como una tarea de clasificación (documentos son clasificados como relevantes e irrelevantes), resulta que el accuracy, una de las métricas más utilizadas para tareas de clasificación, no es una buena métrica de IR. Esto es porque combina desempeño de elementos relevantes (A) con desempeño de elementos irrelevantes (D) -los cuales son numerosos, pero menos interesantes para la tarea.

En las Figura 2.4 muestra un ejemplo de cómo la precision y el recall pueden ser usado para juzgar 2 sistemas, uno contra el otro.

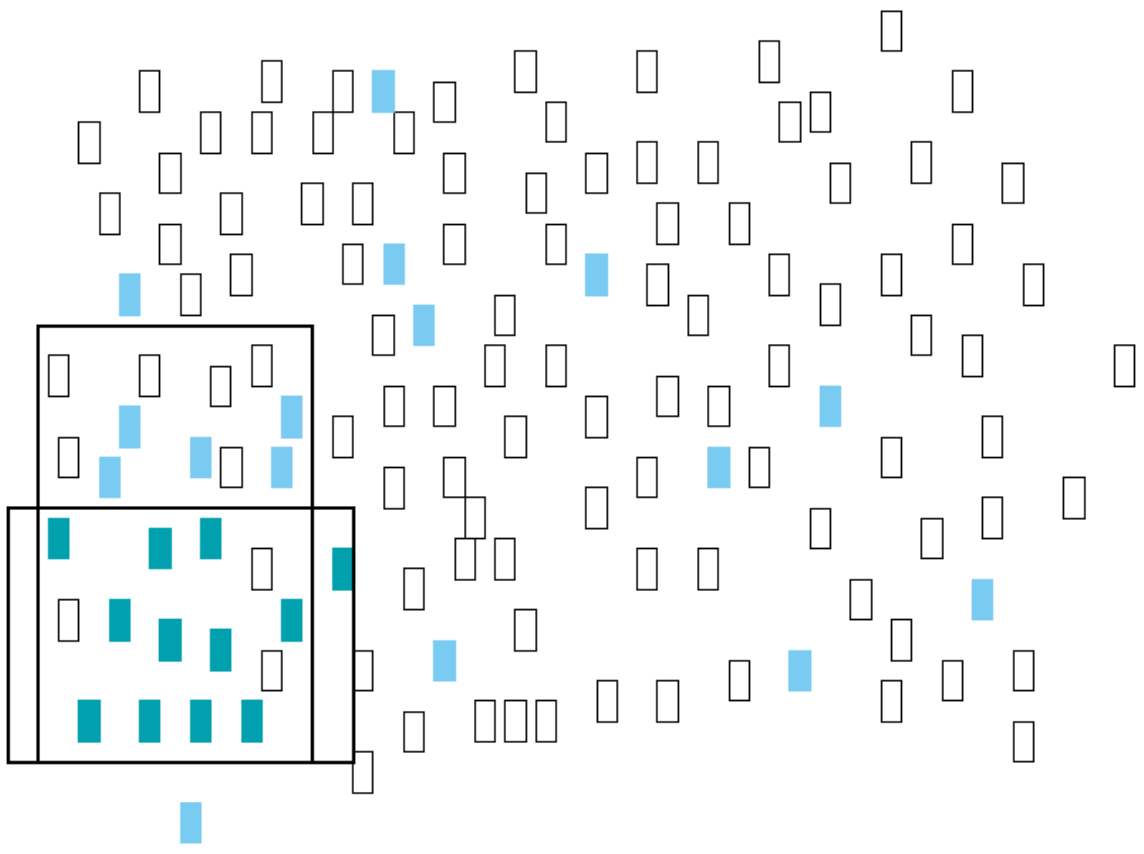


FIGURA 2.4. Ejemplo de uso de Precision y Recall

En este ejemplo, el conjunto completo de documentos es de 130 (A+B+C+D). Entonces, para una consulta dada, hay 28 documentos relevantes (A+C, sombreados). Ahora imaginemos que el sistema ficticio, Sistema 1, recupera 25 elementos dado el rectángulo superior $(A+B)_1$. De estos documentos recuperados, 16 son relevantes (A_1). La precision, el recall y accuracy del Sistema 1 se calculan (en colecciones realistas más grandes, accuracy será cercana a 100 % en todos los sistemas):

$$R_1 = \frac{A_1}{A+C} = \frac{16}{28} = 0{,}57. \tag{2.7}$$

$$P_1 = \frac{A_1}{(A+B)_1} = \frac{16}{25} = 0{,}64. \tag{2.8}$$

$$A_1 = \frac{A_1 + D_1}{A+B+C+D} = \frac{16+93}{130} = 0{,}84. \tag{2.9}$$

Otro sistema, Sistema 2, puede recuperar 15 elementos en el rectángulo más pequeño $(A + B)_2$; en este caso, de los elementos recuperados por el Sistema 2, 12 son relevantes $(A_2 = 12)$ por lo que puede calcularse el desempeño del sistema 2 como:

$$R_2 = \frac{12}{28} = 0{,}43 \tag{2.10}$$

$$P_2 = \frac{12}{15} = 0{,}80 \tag{2.11}$$

$$A_1 = \frac{12 + 99}{130} = 0{,}85 \tag{2.12}$$

El Sistema 2 tienen una precision más alta que el Sistema 1, es decir, este es más *preciso* para recuperar elementos y como resultado, el conjunto retornado contienen una alta proporción de elementos irrelevantes (lo que se mide por la precision), pero se pierde más de los elementos relevantes en las grandes colecciones de documentos (lo que se mide por el recall).

2.3.2. Relación entre Precision y Recall: la medida F En general, hay una relación inversa entre precision y recall, como se ilustra en la Figura 2.5. Aquí la precision y el recall de un sistema ficticio se grafican contra el número de elementos recuperados; mientras más elementos retornan del sistema, más alta es la verosimilitud de que se recuperaren documentos relevantes desde una colección completa. En el caso de que todos los documentos se recuperaran, recall por definición es 1. Esto viene con el costo de recuperar muchos documentos irrelevantes, por lo que mientras más documentos se recuperen, más caerá la Precision.

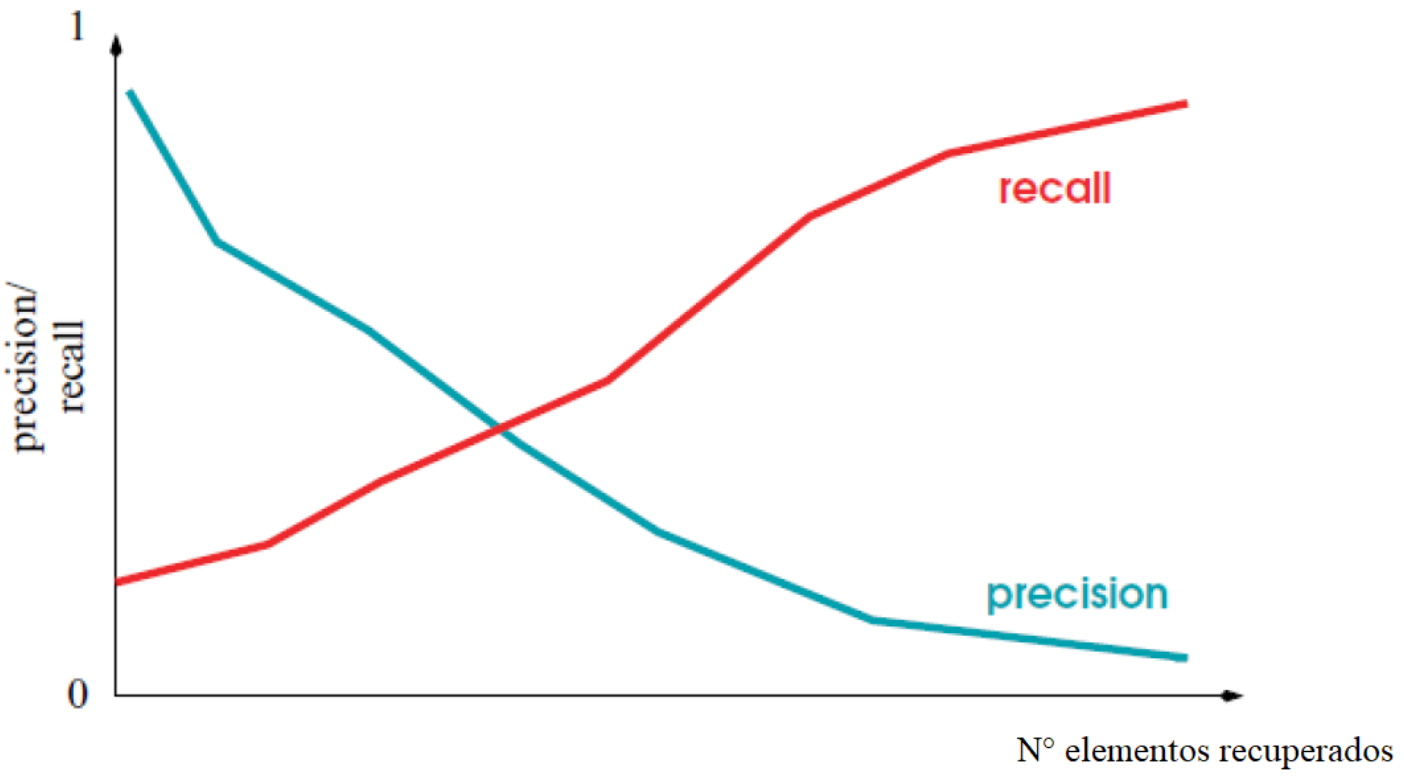


FIGURA 2.5. Relación Inversa entre Precision y Recall

Esta relación inversa entre Precision y Recall fuerza al sistema a comprometer alguno de ellos. Pero hay tareas que particularmente necesitan buena precision y otras que necesitan buen recall. Un ejemplo de esto son las tareas de precision crítica las que son rápidas en Búsqueda Web, donde hay poco tiempo disponible y donde más de un documento relevante existe que pueda responder la información necesaria, debido a la redundancia en la red. Esto significa que se necesita que los documentos relevantes aparezcan en los primeros lugares. Por el contrario, un ejemplo de tarea de recall crítico es la búsqueda de patentes, donde en el peor escenario (con consecuencias costosas) se podría perder alguno de los documentos relevantes; el tiempo no es un problema en este escenario.

En la Figura 2.6 se muestra la relación entre la precision y recall en una manera más estándar, la llamada curva precision-recall. Los datos se ganan manipulando en número de elementos recuperados, como en la Figura 2.5. En sistemas ideales, que combinan alta precision con alto recall, se mostrarán curvas que se estiran tanto como es posible, hacia el rincón superior derecho. El gráfico de precision recall se relaciona con la llamada curva Receiver Operating Characteristic (ROC) conocida desde las ciencias de la vida, en el cual se grafica de la tasa de hits (A en la Figura 2.3) versus la tasa de falsa alarma (B).

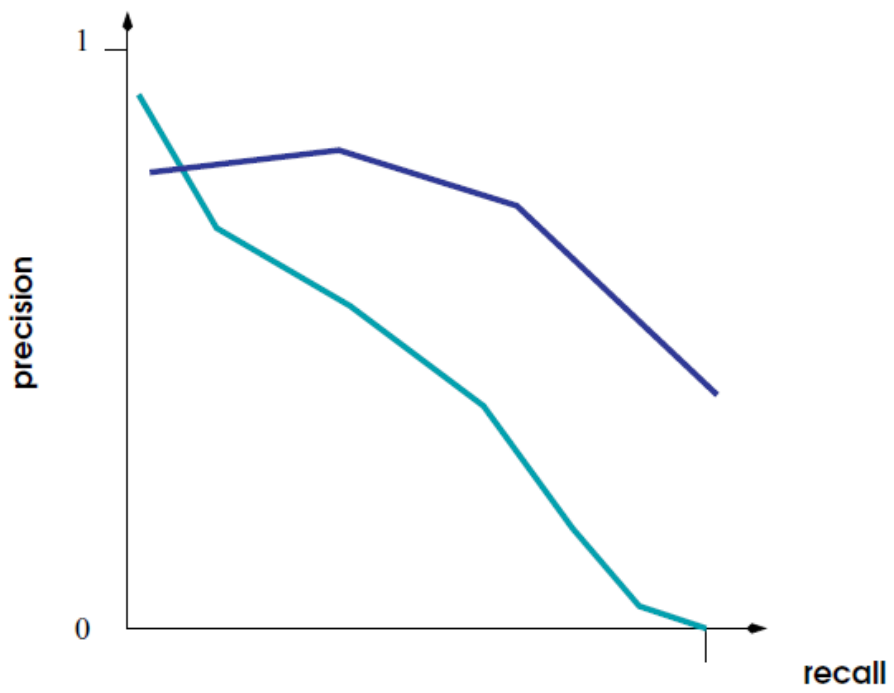


FIGURA 2.6. Dos curvas de Precision y Recall

Debido a la relación inversa entre precision y recall, no es obvio la forma en que el desempeño total de un sistema dado puede ser estimado. Uno podría considerar muchos posibles puntos de precision/recall para cualquier consulta, como también muchas descontinuaciones arbitrarias se pueden usar con los motores de IR de relevancia-ponderada. Esto en oposición con los sistemas Booleanos, que siempre retornan un número fijo de documentos. Una posible respuesta es considerar el área bajo la curva de precision-recall y estimar el desempeño del sistema. Sin embargo, en una configuración práctica, uno no quiere manipular el conjunto de recuperación del sistema, luego graficar la curva precision-recall, y entonces estimar el área

bajo la curva. Dos simples estimaciones permiten determinar empíricamente el cruce entre precision y recall o mejor dicho, calcular la medida F. La medida F [**Rij79**] se define como la media armónica ponderada de precisón y recall:

$$F_\alpha = \frac{PR}{(1-\alpha)P + \alpha R}, \tag{2.13}$$

donde α es el parámetro que permite variar la importancia relativa de recall versus precision (un α grande significa que precision es más importante y viceversa). La medida F es comúnmente usada con $\alpha = 0{,}5$:

$$F_\alpha = \frac{2PR}{P+R}. \tag{2.14}$$

El máximo valor de la medida $F_{0,5}$ para un sistema es una buena indicación de el mejor compromiso entre precision y recall.

2.3.3. Mean Average Precision Existe una medida compuesta y simple que generaliza sobre diferentes consultas, llamada Mean Average Precision (MAP), la cual a veces es referida como Precision Media al observar documentos relevantes. La Precision es calculada en cada punto cuando un nuevo documento relevante se recuperado (usando P = 0 para cada documento relevante que no se recupera). El promedio se determina entonces por cada consulta. Finalmente, un promedio sobre todas ellas se calcula:

$$MAP = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{Q_j}\sum_{i=1}^{Q_j}P(rel = i), \tag{2.15}$$

donde Q_j es el número de documentos relevantes para la consulta j; N el número de consultas y $P(rel = i)$ la precision del i-ésimo documento relevante.

2.4. Ranking de Documentos

La recuperación adhoc tiene que ver principalmente con asignar cierta relevancia entre una consulta y un documento dentro de un corpus. Esto puede ser una actividad muy desafiante ya que los documentos no solo pueden ser relevantes completamente a una consulta, sino que también pueden serlo parcialmente, mientras provean de suficiente información que necesite el usuario.

Esta es una tarea desafiante, ya que por un lado, la información dentro del documento se encuentra no estructurada y por el otro, los documentos abarcan una amplia gama de temas. Es debido a esto, que este tipo de sistemas de recuperación, se enfocan en encontrar elementos en la base de datos, que posiblemente contienen información útil en vez de dar una respuesta específica.

Cada documento se representa como un vector de caracteristicas (identificadores de contenido) junto con pesos asociados. Las características asignadas son las que modelan la información contenida y las consultas tambien se convierten en una representación similar y se espera que un documento útil debe compartir características con la consulta.

La aproximación básica para usar modelos de lenguaje para IR es el modelo de verosimilitud de la consulta, donde cada documento se valoriza por la verosimilitud de este modelo al generar una query Q. Esto puede verse en la Ecuación (2.16),

$$P(Q|D) = \prod_{q \in Q} P(q|D), \tag{2.16}$$

donde D es un modelo de documentos (corpus), Q es una consulta, y q es un término de la query Q. $P(Q|D)$ es la verosimilitud de un modelo de documento que genera los término bajo el supuesto de *bolsa de palabras*, donde los términos son independientes dados los documentos. Esto último, permite asumir el supuesto de que las listas de documentos relevantes pueden ordenarse por $P(q|D)$, lo que supone la probabilidad de la query bajo el modelo de lenguaje, desde donde se deriva el documento d. Finalmente, los documentos se van rankeando por la probabilidad de que la consulta sea observada como una muestra aleatoria desde el modelo de documentos.

Para la recuperación basada en modelos de lenguaje, se trata la generación de consultas como un proceso aleatorio, de la siguiente forma:

1. Inferir el modelo de lenguaje de cada documento.
2. Estimar $P(q|M_D)$, la probabilidad de generar la consulta para cada uno de esos modelos de documentos.
3. Hacer un ranking de acuerdo a esas probabilidades.

En [**AGv04**] se propone una aproximación basado en temas. Este modelo usa información a priori basado en el contenido de temas de un documento, para realizar el ranking. En [**WC06b**] se revisa una aproximación de una forma de hacer recuperación de información construyendo modelos de temas basados usando Latent Dirichlet Allocation (LDA). En este artículo se estudia la efectividad del uso de LDA para mejorar la recuperación adhoc. En [**ZL01**] $P(q_i|D)$ se especifica por el modelo de documentos con suavizado Dirichlet

$$P(w|D) = \frac{N_d}{N_d + \mu} P_{ML}(w|D) + \left(1 - \frac{N_d}{N_d + \mu}\right) P_{ML}(w|coll), \tag{2.17}$$

donde $P(w|D)$ es la máxima verosimilitud estimada de la palabra w en el documento D, y $P(w|coll)$ es la máxima verosimilitud estimada de la palabra w en toda la colección. μ es la prior de Dirichlet.

El modelamiento de documentos (estimar $P(w|D)$) es crucial para la recuperación. Al comparar con modelos de verosimilitud estándar de consultas, LDA ofrece un nuevo marco de trabajo para modelar documentos. Sin embargo, como otros Modelos de Temas, un tema en modelo LDA representa la combinación de palabras. Además LDA se usa con un número

limitado de temas y puede ser muy complejos y dificil de usar como la única representación para IR.

Por lo tanto, en [**WC06a**] se propone una combinación del modelo original de documentos y el modelo obtenido al usar LDA. En este sentido se propone una combinación lineal usando el modelo original de documentos y LDA como se ve en la siguiente ecuación:

$$P(w|D) = \lambda\Big[\frac{N_d}{N_d+\mu}P_{ML}(w|D) + \left(1-\frac{N_d}{N_d+\mu}\right)P_{ML}(w|coll)\Big] + (1-\lambda)P_{lda}(w|D). \tag{2.18}$$

2.4.1. Método de Fusión de Ranking La fusión de rankings es un método usado para combinar diferentes listas en una sola. La idea detrás de este enfoque se desprende del hecho de que algunos métodos de Recuperación de Información (RI) son mejores en algunas consultas específicas que otros, por lo que el conjunto de documentos recuperados desde dos modelos diferentes pueden ser muy distintos y al fusionarlos, es probable que el recall se incremente. Otra observación que se puede hacer, es el hecho de que si un documento se recupera por dos o más modelos de RI, la probabilidad de que el documento sea relevante es muy alta. De hecho, se ha demostrado que combinar diferentes listas de documentos recuperados, mejora el nivel de exactitud del sistema final [**BCB94**].

2.4.1.1. Métodos de Combinación Dentro de los primeros métodos para combinar evidencia desde múltiples modelos, hay algunos desarrollados por Belkin en [**BKFS95**]. Estos enfoques son muy simples y pueden obtener muy buenos resultados. En esta sección se revisarán dos de ellos. La configuración es la siguiente: se tienen n listas de documentos recuperados por n diferentes modelos, cada lista contiene m documentos recuperados para cada consulta. Por lo que, dado una consulta q, la manera más simple de combinar las listas es usando el método *CombSUM*. El valor de un documento en la lista final se obtiene simplemente sumando todos los valores de los documentos de cada modelo:

$$\text{score}_{\text{CombSUM}}(d) = \sum_{m\in D_m} \text{score}(d), \tag{2.19}$$

donde D_m son todos los métodos usados y d es el documento actual. Otro de los métodos, llamado *CombANZ* toma el valor de obtenido por *CombSUM* y lo divide por el número de modelos en el cual el documento aparece en el top 1000 de la consulta dada:

$$\text{score}_{\text{CombANZ}}(d) = \frac{1}{\sum_{m\in D_m : d\in \text{top}_m(1000)1}} \sum_{m\in D_m} \text{score}(d). \tag{2.20}$$

Finalmente, *CombMNZ* multiplica la suma de los valores por el número de los modelos donde el documento aparece en el top-1000:

$$\text{score}_{\text{CombMNZ}}(d) = \sum_{m \in D_m : d \in \text{top}_m(1000)} 1 \sum_{m \in D_m} \text{score}(d). \tag{2.21}$$

Como puede verse en este capítulo, existen varios enfoques en relación a Recuperación de Información. Se sugieren enfoques basados en unigamas y en n-gramas. Los que tiene que ver con unigramas o palabras independientes, requieren que las palabras sean similares a la de las consultas y a los documentos que son relevantes. Estos desempeños deben ser evaluados usando métricas estadísticas como precision, recall y medida F. Una de las tareas en las cuales nos hemos enfocado tiene que ver con la recuperación adhoc y en el ranking de documentos de acuerdo a criterios de relevancia.

Como se presentó en este capítulos los Modelos de Temas son una buena alternativa para poder modelar documentos y lenguajes, además presentan una buena alternativa para realizar recuperación y ranking de documentos. En el siguiente capítulo desarrollamos los Modelos de Temas, específicamente, Latent Dirichlet Allocation, que es uno de los más utilizados.

Capítulo 3

Topic Models

En Aprendizaje Automático y en Procesamiento de Lenguaje Natural [**EL07**], existen modelos estadísticos como los Modelos de Temas que sirven para descubrir los temas que existen en una colección documental [**LTD+16**] [**BB17**]. El modelamiento de temas se usa frecuentemente en Minería de Datos para encontrar las estructuras semánticas latentes en un cuerpo de texto [**JBJ16**]. Intuitivamente, dado que un documento se trata de cierto tema en particular, este debiese aparecer en el texto con más frecuencia y sus palabras relacionadas también. Por ejemplo *perro* y *hueso* aparecerán más frecuentemente en un documento sobre perros, *gato* y *miau* lo harán en un documento sobre gatos, y *los* y *son* aparecerán de igual forma en ambos, ya que son palabras muy comunes en el español. Un documento, por lo general, tiene muchos temas divididos en diferentes proporciones, por lo que en un documento que es 10 % sobre gatos y 90 % sobre perros, habrá probablemente 9 veces más palabras sobre perros que palabras sobre gatos. Los temas producidos por las técnicas de modelamiento de temas son *clusters* de palabras similares. Un Modelo de Temas captura esta intuición en un marco de trabajo matemático, el que permite analizar el documento y descubrir, basado en la estadística de cada palabra, que temas pueden estar relacionados y cual es su proporción en el documento.

En este capítulo haremos una descripción de los modelos gráficos [**HWWG13**]. Estos modelos probabilísticos muestran las relaciones que existen entre las variables aleatorias a través del uso de grafos. Estos modelos han sido aplicados en un gran número de campos, por ejemplo: bioinformática, ciencias sociales, teoría de control, procesamiento de imágenes, análisis de márketing entre otros [**BB01**]. Luego describiremos *Latent Dirichlet Allocation* (LDA) [**JWY+18**], que es una de las técnicas más utilizadas para realizar Modelamiento de Temas [**Wu18**]. La idea básica de esta técnica es que los documentos se representan como una mezcla aleatoria sobre los temas (variables latentes) que lo componen, donde los temas se caracterizan como una distribución sobre las palabras. Finalmente describiremos el concepto de Coherencia de Temas [**MWT+**]. El objetivo de esta métrica es medir el grado de similaridad semántica entre las palabras con más alto puntaje en el tema. Esta medida ayuda a distinguir

entre temas que son semánticamente interpretables y temas que son solo objetos salidos de la inferencia estadística.

3.1. Modelos Gráficos

Dentro del área de Aprendizaje Automático, los Modelos Gráficos Probabilísticos han sido un tema ampliamente usados en investigaciones científicas [**Mur12**]. Estos modelos se dibujan mediante un grafo y representan una estructura de condición de dependencia entre variables aleatorias. Se usan comúnmente en teoría de la probabilidad, tanto en estadísticas como en Aprendizaje de Máquinas. De esta manera, se puede entender visualmente la relación que existe entre estas variables y así, realizar tareas de inferencia estadística con la información observada. Estos modelos representan la distribución de probabilidad conjunta sobre cierto número de variables que se encuentran factorizadas, lo cual supone una independencia entre ellas. En la Figura 3.1 se ven los ejemplos de varias representaciones de modelos gráficos, entre los que tenemos; a) unigramas, b) *n-gramas* donde hay relación de probabilidad entre las palabras o c) *Indexación Probabilística Semántica Latente (o pLSI)* donde encontramos estructuras latentes ocultas a partir de las palabras. Lo anterior puede ser entendido a través de la distribución de probabilidad que existe sobre un *corpus* (colección de textos escritos sobre un tema en particular). En estos modelos, los círculos representan variables aleatorias, las cuales son parámetros de las distribuciones de probabilidad. Por otro lado, la notación de *placas* (cuadros alrededor de un subconjunto de círculos) se utiliza para representar una repetición de parámetros. Finalmente, las *flechas* muestran una dependencia entre las variables.

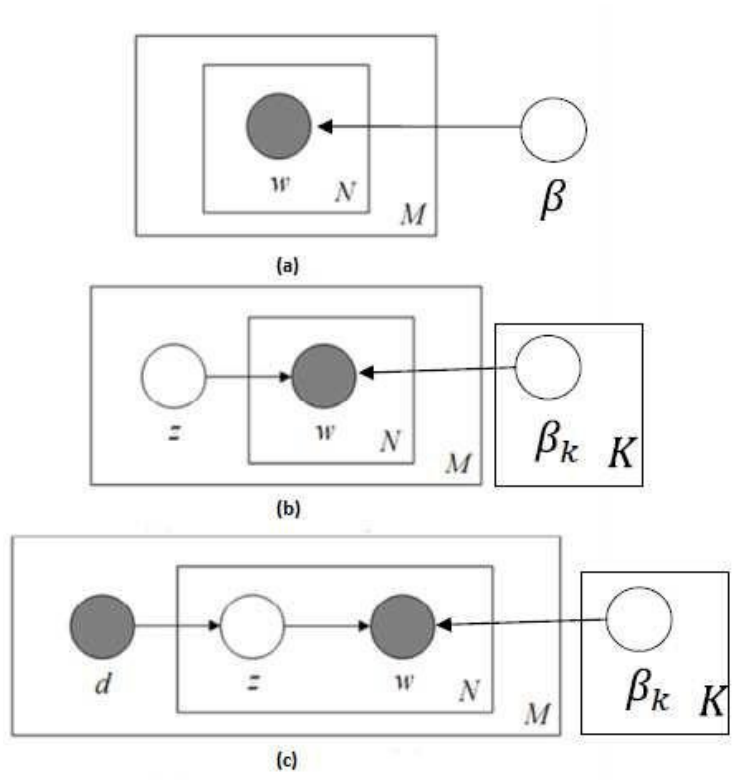


FIGURA 3.1. Ejemplos de Modelos Graficos (a)unigrama, (b) Mezcla de unigramas, (c) Indexación Probabilística Semántica Latente

En el área de modelado de texto, las palabras se consideran como las características de los documentos y sus frecuencias dentro de estos como los valores de estas características. Como el tamaño del vocabulario es demasiado grande, estas técnicas sufren de la **maldición de la dimensionalidad** (el aumento del volumen de datos y el volumen del espacio aumenta exponencialemente) [**Don00**], y es debido a esto que se necesitan métodos para encontrar formas de reducir la dimensionalidad del problema. Algunas veces, esto se logra con una simple selección de características [**GE03**], por ejemplo, eliminando las palabras vacias (*stopwords*) o truncar el vocabulario basado en una frecuencia mínima. Lamentablemente, en algunos casos, es necesario encontrar una representación dimensional menor para que estos documentos codifiquen sus propiedades semánticas y esto se soluciona en parte con los Modelos de Temas. A continuación revisaremos LDA, que es una de las técnicas más usadas en la literatura.

3.2. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [**BNJ03**] es un modelo jerárquico bayesiano que captura la información relacionado con los temas de una colección de documentos usando las probabilidades a priori de Dirichlet y encontrando una mezcla de ellos. Estos temas representan la distribución sobre las palabras, y el modelo aprende considerando la información de las palabras a nivel de cada uno de los documentos y a nivel del conjunto completo.

Podemos señalar una **palabra w** como una unidad de dato discreto que se define como un objeto dentro de un vocabulario indexado por $\{1, 2, ..., V\}$. Además, un **documento** es una

secuencia de **N** palabras denotado por $d = \{w_1, w_2, ..., w_N\}$, donde w_N es la n-ésima palabra de la secuencia. Un corpus es una colección de M documentos denotados por $D = \{d_1, d_2, ..., d_M\}$.

El objetivo final es encontrar un modelo probabilístico que no sólo asigna alta probabilidad a los miembros de un corpus, sino que asigna alta probabilidad a otros documentos similares.

Es importante agregar que LDA es un modelo probabilístico generativo. La idea básica consiste en que los documentos están representados como una mezcla aleatoria sobre temas latentes donde cada tema se caracteriza como una distribución de probabilidad sobre las palabras.

En la Figura 3.2, se ve como LDA asume el siguiente proceso generativo para cada documento en el corpus:

- Por cada tema **t**:
  - Obtener la Multinomial $\phi_t \sim \text{Dir}(\beta)$
- Por cada documento **d**:
  - Obtener la multinomial $\theta_d \sim \text{Dir}(\alpha_1, ..., \alpha_T)$
  - Por cada palabra **i**:
    - Obtener $z_i \sim \text{Mult}(\theta_d)$
    - Obtener $w_i \sim \text{Mult}(\phi_{z_i})$

FIGURA 3.2. Proceso Generativo de LDA

Sobre el modelo anterior se hacen ciertos supuestos: primero la dimensionalidad **t** de la distribución de Dirichlet (y en consecuencia la dimensionalidad de la variable los temas z) se asume conocido y fijo. Segundo, las probabilidades de las palabras se parameterizan por una matriz β de k x V. Notar que N es independiente de todos los otros datos (θ o z).

LDA usa principalmente la distribución de Dirichlet, que corresponde a una familia de distribuciones sobre un simplex de vectores positivos que suman 1. Esta distribución posee la siguiente función de densidad:

$$P(\theta|\vec{a}) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\Pi_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} ... \theta_k^{\alpha_k - 1}, \tag{3.1}$$

donde el parámetro α es un vector k dimensional con componentes $a_i > 0$ y donde $\Gamma(x)$ es la funcion Gamma.

Dado los parametros α y β, la mezcla de temas θ, además de un conjunto de K temas y un conjunto de N palabras w, la distribución conjunta de estas variables y se representa en la Ecuación 3.2:

$$P(\theta, z, w|\alpha, \beta) = P(\theta|\alpha) \Pi_{n=1}^{N} P(z_n|\theta) P(w_n|z_n, \beta). \tag{3.2}$$

El problema computacional de los algoritmos de topic model como LDA es la aproximación de la probabilidad conjunta posterior de las variables latentes del modelo dados los términos, en la Ecuación 3.3:

$$P(\phi_{1:K}, \theta_{1:D}, z_{1:D} \mid w_{1:D}) = \frac{P(\phi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{P(w_{1:D})}, \tag{3.3}$$

Esta distribución es la clave para usar LDA en tareas cuantitativas o cualitativas, como predicción o generalización de documentos. Algunas técnicas de aproximación que han sido usadas por LDA son: *mean field variational inference* [**BNJ03**], *collapsed variational inference* [**BNJ03**], *expectation propagation* [**ML02**] y *gibbs sampling* [**Ble12**]. Cada una presenta sus ventajas y desventajas, escogiendo un algoritmo de inferencia basado en velocidad, complejidad, precisión y simplicidad.

El modelo LDA se representa como un modelo gráfico probabilístico como se ve en la Figura 3.3. En ella hay tres niveles de representación de LDA. Los parámetros α y β son parámetros a nivel de corpus, se asume que se muestrean una vez cuando se genera el corpus. Las variables θ_d son variables a nivel de documento, muestreadas de cada uno de ellos. Finalmente, las variables z_{dn} y w_{dn} son variables a nivel de palabras y son muestreadas una vez por cada palabra en cada documento.

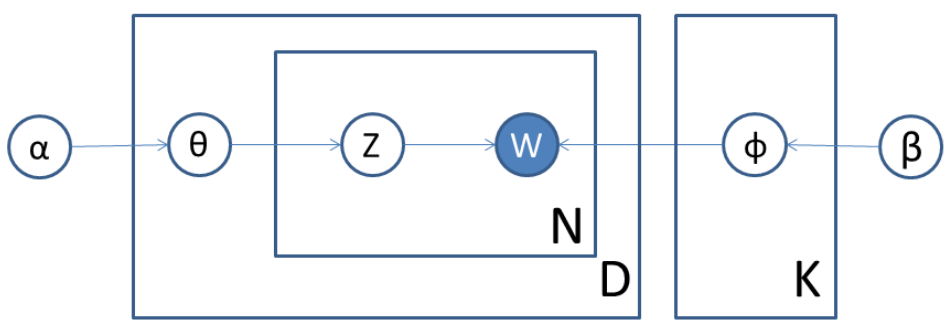


FIGURA 3.3. Modelo Grafico de LDA

El Cuadro 1 detalla las variables del modelo gráfico.

Nombre	Visibildad	Descripción
W	Observada	Palabras en el documento D.
Z	Oculta	Asignación de palabras por temas.
θ	Oculta	Proporción de temas por documentos.
ϕ	Oculta	Distribución de temas por corpus.
α	Parámetro de Dirichlet.	
β	Parámetro de Dirichlet.	

CUADRO 1. Variables LDA

Por último, en la Figura 3.4 se muestra el esquema de cada uno de los pasos de LDA, desde que ingresa el corpus completo, hasta que se obtienen las matrices que representan el modelo:

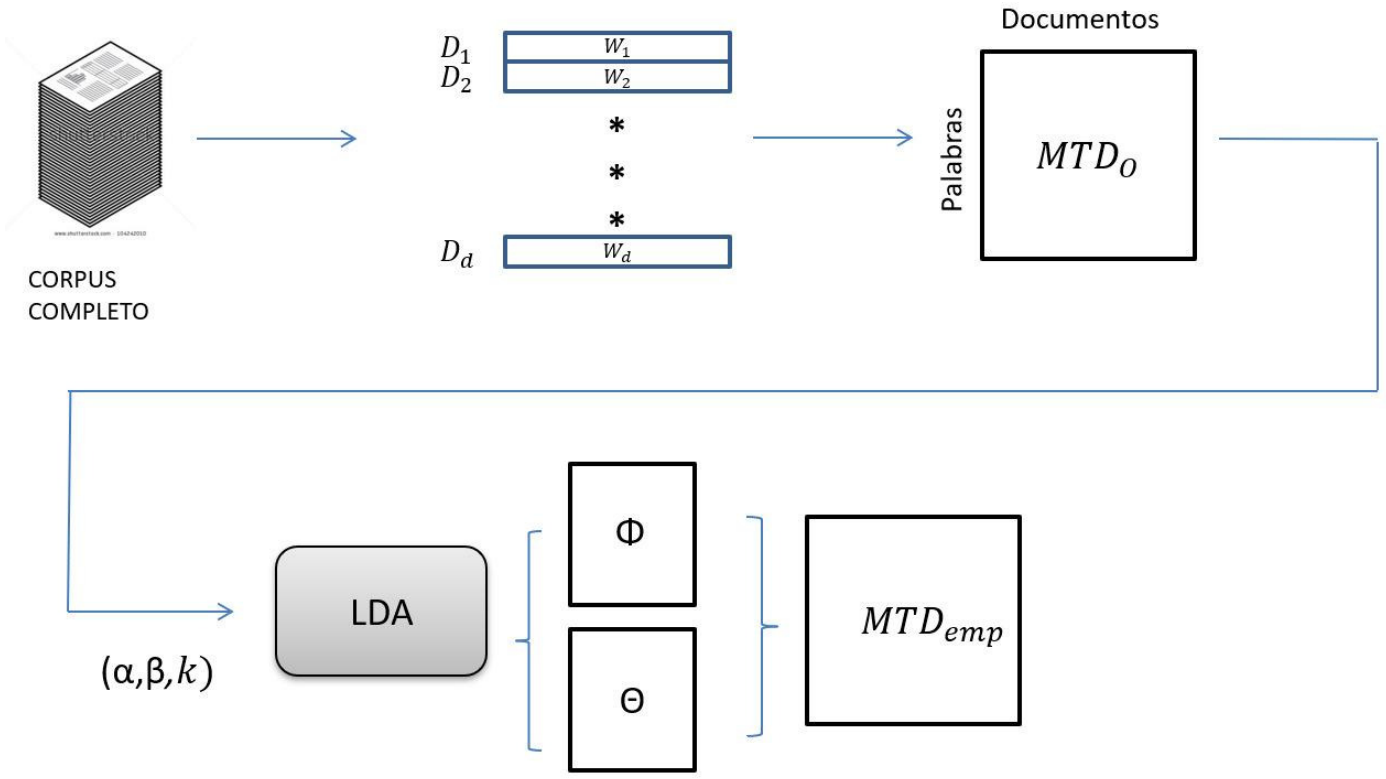


FIGURA 3.4. Flujo de LDA

3.3. Divergencia Kullback-Lieber

Para medir la diferencia entre dos distribuciones de probabilidad sobre la misma variable x, la divergencia Kullback-Leiber, o simplemente, la divergencia KL, ha sido popularmente usada en la literatura de Mineria de Datos [**QWCZ14**], [**PKVP06**]. El concepto se originó en la teoría de la probabilidad y la teoría de la información. La divergencia KL, que está relacionada con la entropía relativa, la divergencia de información y la información por discriminación, es una medida no simétrica de la diferencia entre dos distribuciones de probabilidad $P(x)$ y $q(x)$. Específicamente, la divergencia KL de $q(x)$ desde $P(x)$, denotada como $D_{KL}(P(x), q(x))$, es una medida de la información perdida cuando $q(x)$ se usa para aproximar $P(x)$. Sea P(x) y q(x) dos distribuciones de probabilidad de una variable aleatoria discreta x. Esto es, ambas $P(x)$ y $q(x)$ suman 1, y $P(x) > 0$ y $q(x) > 0$ para cualquier x en X. $D_{KL}(P(x), q(x))$, se define en la Ecuacion 3.4:

$$D_{KL}(P(x)||q(x)) = \sum_{x \in X} P(x) \ln \frac{P(x)}{q(x)}. \tag{3.4}$$

Las divergencia KL mide los números esperados de los bits extras requeridos para codificar muestras desde $P(x)$ cuando se usa un código basado en $q(x)$, en vez de usar el código basado

en $P(x)$. Típicamente $P(x)$ representa la distribución *verdad* de los datos, u observaciones o distribución calculada teóricamente. La medida de $q(x)$ representa típicamente una teoría, modelo, descripción o aproximación de $P(x)$. La versión continua de la divergencia KL es:

$$D_{KL}(P(x)||q(x)) = \int_{x \in X} P(x) \ln \frac{P(x)}{q(x)} dx. \tag{3.5}$$

Aunque la divergencia KL mide la distancia entre dos distribuciones, esta no es una medida de distancia. Esto se debe a que la divergencia KL no es una medida métrica. No es simétrica: la KL de $P(x)$ a $q(x)$ no es la misma generalmente que la KL de $q(x)$ a $P(x)$. Además, no necesita satisfacer la desigualdad triangular. A pesar de eso, $D_{KL}(P||Q)$ no es una medida negativa. $D_{KL}(P||Q)) \geq 0$ y $D_{KL}(P||Q)) = 0$ si y solo si $P = Q$.

Los modelos de temas tienen un importante rol en ciencias de la computación para minería de texto o en el procesamiento de lenguaje natural. En el modelamiento de temas, un tema es una lista de palabras que se obtiene de métodos estadísticamente significativos. Un texto puede ser un email, un capítulo de libro, un post, un artículo de revista y cualquier tipo de texto sin estructura. Estos métodos no pueden entender el significado y los conceptos de las palabras. Por el contrario, se supone que cualquier parte del texto se combina al seleccionar las palabras de una bolsa, donde cada bolsa corresponde a un tema en particular. Esta herramienta realiza este proceso una y otra vez hasta que se queda con la distribución de palabras más probable dentro de la bolsa a la que llamamos tema. El modelamiento de temas entrega un punto de vista útil de una gran colección documental en término de la colección como un todo (corpus), los documentos individuales y la relación entre documentos. En LDA existe una alta dependencia de los parámetros de entrada del método. Estos parámetros son k que es la cantidad de temas que se cree tiene la colección documental y los supuestos distribucionales α y β. Esto le permite a LDA realizar ciertos supuestos. Un documento puede tener múltiples temas (y es debido a esto que se utiliza la distribución Dirichlet) y existe una distribución que modela esta relación. Por otro lado, las palabras también pertenecen a múltiples temas, cuando se consideran que existen fuera de un documento en particular.

Es decir, dependiendo de los parámetros que usemos, serán los supuestos que tengamos y, así también será la calidad del resultado que obtengamos. Por lo que necesitamos medidas que nos permitan saber que tan buena es la calidad de la solución, como por ejemplo la divergencia Kullback-Lieber.

En el siguiente capítulo revisaremos los Ensamblados de Máquinas [**Kun14a**] que son técnicas que nos permiten solucionar, en parte, el problema de la variabilidad de estas soluciones.

Capítulo 4

Ensamblando Modelos de Temas

El aprendizaje de ensamblado es el proceso por el cual múltiples clasificadores o expertos, se generan de manera estratégica y se combinan para resolver un problema de inteligencia computacional. El aprendizaje de ensamblado se utiliza de manera principal para mejorar el desempeño de un modelo.

Existen diversos trabajo en los cuales se muestran las bondades de este tipo de metodología [**Kun14a**], [**Die00**], pero todas se encargan de resolver un problema fundamental del aprendizaje de ensamblado: cómo entrenar el modelo base (algoritmos de ensamblado), cómo combinar las salidas que se obtienen de estos modelos (métodos de combinación) y cuál es el factor fundamental para determinar el éxito de estas propuestas (diversidad).

Por otro lado, como se vio en el capítulo anterior, los Modelos de Temas son técnicas que permiten encontrar la distribución de temas sobre el vocabulario de un conjunto de documentos y corpus.

Se ha propuesto diferentes metodologías para ensamblar este tipo de modelos y así obtener mejores resultados en relación al desempeño [**RC13**], [**SLYS10**].

En este capítulo repasaremos conceptos fundamentales de aprendizaje de ensamblados y veremos los trabajos relacionados con Ensamblados de Modelos de Temas. Finalizamos proponiendo una taxonomía para agrupar este tipo de modelos.

4.1. Aprendizaje de Ensamblados

Los métodos de ensamblados de máquinas tienen como objetivo fundamental mejorar el desempeño y la precisión de los modelos de aprendizaje estadístico, usando modelos que en su definición parecen simples. El principio fundamental de los ensamblados se basa en construir una combinación lineal de un conjunto de modelos base en lugar de usar un único modelo global.

Inicialmente se usa un conjunto de modelos de entrenamiento para construir un clasificador. Dicho clasificador es una hipótesis sobre la verdadera función f, es decir, dado un nuevo x, trata de encontrar el correspondiente valor de y. Un método de ensamblado de clasificadores, es un conjunto de estos modelos o hipótesis $h_1, ..., h_l$ para los cuales las decisiones individuales se combinan para obtener un nuevo dato, por ejemplo, mediante votación o promedio ponderado. La principal característica de los métodos de ensamblado es que son una forma flexible de aproximar la función f, debido a que podemos garantizar una mejor capacidad de generalización.

Una condición necesaria para que el clasificador general posea mejor precisión y capacidad de generalización que el promedio de los miembros, es que estos modelos deben ser débiles y diversos. Un clasificador *débil* se define como aquel que está ligeramente correlacionado con la verdadera clasificación (tiene un desempeño de clasificación ligeramente superior a adivinar aleatoriamente, cuya probabilidad de error es de 0,5 en el caso de la clasificación binaria). Por otro lado, dos clasificadores son *diversos* si ellos cometen diferentes errores sobre un nuevo dato.

Si bien esta representación formal del problema es interesante, en la práctica no es suficiente. Sin embargo, existen 3 razones fundamentales que permiten afirmar que se pueden construir buenos ensamblados [**Kun14a**], [**Die00**], las que están resumidas en la Figura 4.1.

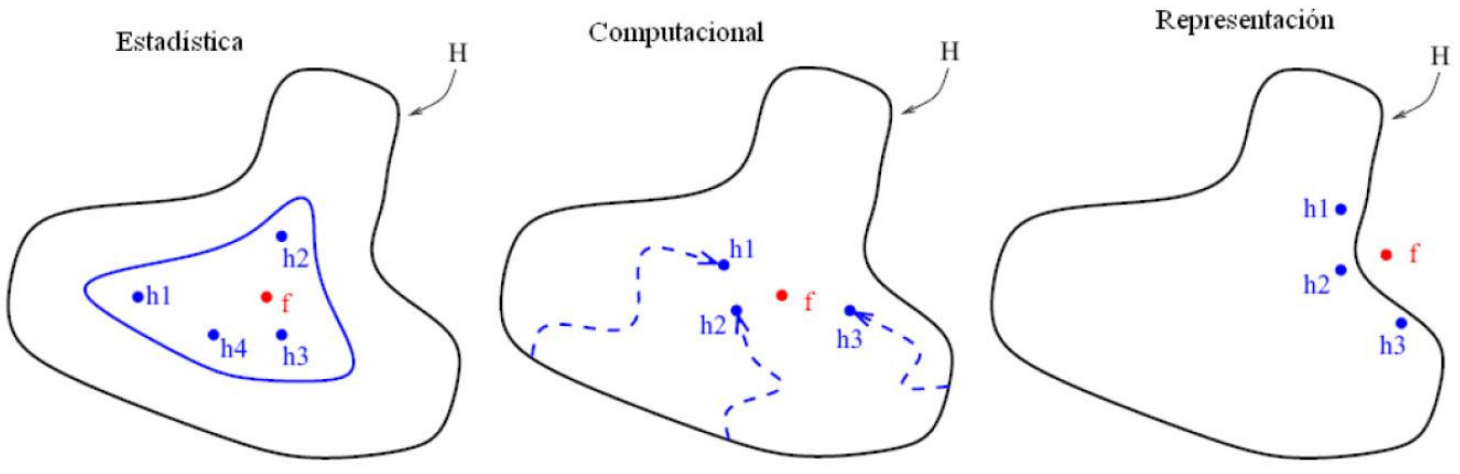


FIGURA 4.1. Las tres razones fundamentales de Ensamblados.

1. **Razón Estadística:** un algoritmo de aprendizaje puede ser visto como una búsqueda en el espacio H para identificar la mejor hipótesis existente. El problema radica en que la cantidad de datos de entrenamiento disponibles es demasiado pequeña comparada con el tamaño del espacio. Sin suficientes datos, el algoritmo puede aprender diferentes hipótesis en H, a tasas de error similares. Al construir un ensamblado de todos estos clasificadores, el algoritmo puede ponderar los modelos y reducir el riesgo de entregar un resultado equivocado.
2. **Razón Computacional:** muchos algoritmos de aprendizaje pueden quedarse atrapados en óptimos locales, ya que realizan búsquedas en lugares específicos dentro del

espacio de hipótesis. Un ensamblado se construye realizando un remuestreo del conjunto de datos, permitiendo realizar búsquedas en diferentes sectores de este espacio, por lo que es capaz de entregar una mejor aproximación de la función desconocida.

3. **Razón de Representación:** en la mayoría de las aplicaciones de máquinas de aprendizaje la verdadera función f puede no ser representada por alguna hipótesis en H. Al formular una suma de hipótesis obtenidas desde H, es posible expandir el espacio de funciones representadas.

Los métodos de ensamblado entrenan múltiples máquinas que resuelven el mismo tipo de problema. Al contrario de los enfoques que tratan de construir máquinas en base a los datos de entrenamiento, los métodos de ensamblado intentan construir un modelo a partir de una serie máquinas que resuelven una situación en particular y las combinan para obtener una mejor solución.

Las habilidades de generalización de un ensamblado son mucho más fuertes que el de sus componentes. El aprendizaje con ensamblados es interesante principalmente porque puede mejorar el desempeño de los modelos débiles un poco mejor que si estos estuvieran adivinando y los hace más fuertes, los cuales pueden realizar una predicción más precisa. Un ensamblado puede construirse en dos pasos. Primero, se genera un número de componentes. Luego, en una segunda etapa, se decide cómo los componentes son usados para realizar la predicción. Generalmente, para obtener buenos ensamblados, los componentes deben ser tan precisos y diversos como sea posible.

Dentro de los métodos de ensamblados más populares se encuentran Boosting [**Sch90**] y Adaboost [**FS97**]. Estas técnicas fueron creadas inicialmente por Nilsson [**Nil65**], quien agrupó un conjunto de perceptrones en una capa de entrada y combinó las salidas en una segunda capa de votación. Posteriormente se propusieron estrategias de agregación más avanzadas como Bagging y Boosting.

4.1.1. Bootstrap Aggregating (Bagging)

En Bagging (Bootstrap Aggregation Learning) [**BB96**] se genera aleatoriamente una muestra bootstrap para cada uno de los miembros del ensamblado desde una distribución uniforme, con el objetivo de generar diversidad en las muestras generadas. Al hacer esto, se logra que cada máquina se entrene con un subconjunto diferente de datos, el que se construye a partir del conjunto de entrenamiento original.

Dado un conjunto de tamaño N, se eligen N objetos aleatorios uniformemente distribuidos con reposición, con los cuales se entrena el nuevo miembro del ensamblado. Este proceso se repite para cada uno de los elementos (ver Algoritmo 2).

Algoritmo 2 Bagging

1: Sea M el número de predictores necesarios.
2: Sea $d = \{(x_1, y_1), ..., (x_N, y_N)\}$ el conjunto de datos.
3: **para** t= 1,...,M **hacer**
4: Generar una nueva muestra d_{bag} escogiendo N muestras desde d con reemplazo.
5: Entrenar un estimador f_i con la muestra d_{bag} y agregarlo al ensamblado.
6: **fin para**
7: La hipótesis final:

$$H(x) = \sum_{t=1}^{T} h_t(x). \tag{4.1}$$

4.1.2. Adaptive Boosting (Adaboost) El método de boosting [**Sch90**] es una estrategia general de aprendizaje de modelos por medio de la combinación de estos. La idea de boosting es utilizar un clasificador débil (cualquier clasificador que sea un poco mejor que el azar) para construir un ensamblado de estos, y de este modo aumentar su desempeño. Esta mejora se realiza por medio de un promedio de salidas de esta colección de clasificadores. El algoritmo más popular de boosting es Adaboost [**FS95**]. En esta técnica se entrena el primer modelo con un conjunto de muestra que se crea a partir de datos uniformemente distribuidos. Esto quiere decir que $D_1(i) = \frac{1}{i}$ para todo i. Al final de la iteración, a los patrones mal clasificados se les asigna una probabilidad mayor en el conjunto inicial, que se utiliza para crear el siguiente miembro del ensamblado. Una vez que se crean todos estos miembros, la salida se genera por votación ponderada basada en el error de entrenamiento. La clasificación que entrega Adaboost es una combinación lineal de los parámetros de confianza de los clasificadores que dependen de este error ponderado:

$$\epsilon_t = P_{r_i \sim D_t}[h_t(x_i) \neq y_i] = \sum i : h_t(x_i) \neq y_i D_t(i).$$

El Algoritmo 4 muestra el procedimiento

Algoritmo 4 Adaboost

1: Dados $\{(x_1, y_1), ..., (x_m, y_m)\}$ donde $x_i \in X$, $y_i \in Y = \{-1, +1\}$.
2: Inicializar la distribución de la muestra $D_1(i) = \frac{1}{n}, \forall i = 1, ..., n$.
3: **para** t= 1,...,T **hacer**
4: Entrenar los modelos débiles usando la distribucion D_t.
5: Tomar las hipótesis débiles $h_t : X \to \{-1, +1\}$

$$\epsilon_t = P_{r_i \sim D_t}[h_t(x_i) \neq y_i]. \tag{4.2}$$

6: Escoger $\alpha = \frac{1}{2} ln \frac{1-\epsilon_t}{\epsilon_t}$
7: Actualizar

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \begin{cases} e^{-\alpha_t} & \text{si } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{si } h_t(x_i) \neq y_i \end{cases} . \tag{4.3}$$
$$= \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

donde Z_t es un factor de normalización.
8: **fin para**
9: La hipótesis final:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right). \tag{4.4}$$

Estudios en métodos de ensamblado [**Mac97**],[**OM97**] muestran que Adaboost tiene un mejor desempeño que Bagging, aunque es sensible a los outliers (o datos atípicos) y es fácil sobreajustar los datos.

4.2. Ensamblados de Modelos de Temas

En esta sección revisaremos los trabajos relacionados con los Ensamblados de Modelos de Temas, y sus aplicaciones tanto en tareas de Recuperación de Información, como en tareas de Clasificación y Regresión.

4.2.1. Trabajos Relacionados En primer lugar tenemos que en [**RC13**] se usan mezclas de procesos de Dirichlet y Aprendizaje de Ensamblados con el objetivo de incrementar el desempeño a través de diferentes fuentes de datos (diferentes contextos). En este caso en particular, se utilizan datos de atencíon médica, y trabajan con datos de diferentes dominios (genéticos, de raza, nivel socioeconómico). Para este objetivo se utilizan datos de diferentes niveles de características.

Este modelo permite construir una representación de red de datos. Además, la relación entre enfermedades no es evidente al examinar las frecuencias por separado. El uso de una

representación relacional de los datos (la co-ocurrencia de las enfermedades) permite obtener una información más específica.

Con el objetivo de utilizar datos obtenidos de diferentes fuentes, es que se trabaja con el concepto de Ensamblado. Para esto, los datos se reúnen para así crear un modelo global. El artículo indica que los modelos son promediados, y estos se van agregando.

Si bien presenta un marco donde se usa ensamblados, no plantea la forma en el cual los parámetros de los Procesos de Dirichlet son inicializados, ya que debido a la dispar naturaleza de esto, se puede pensar que no deban usarse las mismas suposiciones a priori.

En [**SLYS10**] se presenta un marco de trabajo, como una solución que combina Modelos de Temas sobre la partición del corpus completo. Esto puede verse de dos maneras. Primero, esto se puede usar en Modelos distribuidos para grandes corpus, y segundo, puede usarse en Modelos incrementales para corpus que crecen rápidamente.

Suponga que cada documento corresponde a sólo un subcorpus, se puede denotar c_d. Entonces se puede aprender un modelo de temas separado de cada subcorpus. En la Ecuación (4.5) podemos ver la representación de esto.

$$P(w, d|c) = P(d|c)P(w|d, c) = P(d|c) \sum_{z \in Z_{c_d}} P(z|d)P(w|z), \tag{4.5}$$

con z $\in Z_c$ el modelo base aprendido localmente.

El marco de trabajo es dividido en 3 parte por los autores:

- **Fase 1: Topic Model Base,** se aprende el tópico base $P(w|z, c)$ de cada uno de los sub-corpus c, como se muestra en la imagen.
- **Fase 2: Modelando el Ensamblado de Temas,** se aprende el ensamblado de temas $P(w|y)$ sobre la co-ocurrencia de z y w en todo el corpus, como se ve en la figura. Acá, z in cualquier tema in la unión de todos los temas bases.
- **Fase 3: Inferencia,** Tomar $P(w|y)$ como el modelo de temas resultante y inferencia $P(y|d)$ para cada documento d. El ensamblado de temas, modela una solucón para el corpus completo.

En [**STGCL12**] se propone un Aprendizaje de Ensamblado con Modelos de Temas usando LDA debido a las grandes ventajas al explotar la co-ocurrencia de las características locales o palabras visuales, con el objetivo de descubrir estructuras comunes o intrínsecas de los datos.

Primero, se usa LDA generativos para encontrar la estructura oculta de temas de grandes volúmenes datos visuales, y entonces realiza esta categorización al agrupar datos de entrenamiento a gran escala con gran variación dentro de muchos pequeños temas locales. Segundo, realizan discriminaciones correctamente al entrenar cada tema localmente para generar múltiples clasificadores con pequeña efectividad. Así, los datos dentro de un tema parecieran ser similares, en parte, con respecto a la estructura oculta. El modelo individual correspondiente toma mejores decisiones sobre este tema.

La solución propuesta está motivada por una visión desde los estudios psicofísicos de que los humanos pueden categorizar los objetos visuales más fácil y rápidamente. Como las imágenes de la misma categoría tienen partes en común, por ejemplo, los autos tienen ruedas o los

animales tienen ojos y piernas, la propuesta del artículo es un aprendizaje de ensamblado con topic models LDA debido a si gran ventaja en explorar la co-ocurrencia de las características locales o palabras visuales para descubrir estructuras latentes de los datos, similares o comunes.

En [**LQQQ**] se propone un algoritmo de clasificación basado en boosting llamado LDA-Boost, y LDA es usado para modelar el espacio de características en el marco de la tarea de Categorización de Texto. En este caso, en vez de usar palabras o frases, LDABoost usa los temas latentes como características. Es a causa de esto, que la dimensión de las características se reduce significativamente.

La reducción de dimensionalidad son las bases de LDABoost. Este algoritmo usa LDA para modelar documentos. Gibbs Sampling se usa para estimar los parámetros de LDA, los temas se estiman dentro de estos parámetros. Los más representativos se evalúan con la distancia de Mahalanobis para crear un conjunto de características. Ellos usan una Naive Bayes multinivel como clasificador y Mutual Information como medida de desempeño.

El objetivo de usar LDA, es el de reducir la dimensionalidad y para extraer características. Como clasificador débil se utiliza un algoritmo Naive Bayes Multinivel.

En [**RW12**] se propone un método para detectar sitios que son usados para el fraude electrónico. Este método usa LDA para realizar análisis semántico, y Adaboost se usa para la clasificación.

LDA construye un modelo de temas desde el contenido que tiene de ambos tipos de sitios, de phishing y legítimos. El modelo usa Gibbs Sampling para los parámetros de estimación. Y, el clasificador se construye usando la distribución de temas de LDA como entrada.

En este paper se utiliza Perplexity como métrica de generalización de la forma:

$$\text{Perplexity} = exp\left(-\frac{\sum_d logP(w_d)}{\sum_d N_d}\right). \tag{4.6}$$

En el paper [**GLJG16**] se utiliza AdaboostMH [**SS00**] para realizar clasificación de documentos utilizando LDA como una forma de extraer las características del corpus. La idea es integrar estas dos metodologías de la siguiente forma. Se divide el conjunto completo en uno de entrenamiento y otro de test. Las entradas son preprocesadas (se obtienen los tokens, se obtienen las raíces, se eliminan las stopwords) y se ejecuta LDA sobre la colección documental y se obtienen las respectivas θ_D, la distribución documento-tópico y ϕ_t, la distribución tópico-palabra.

Se utiliza la distribución θ_D para modelar los documentos. Finalmente, cada documento se presenta como un conjunto de tópicos, en vez de una bolsa de palabras.

En el paper [**WG14**] se tiene un algoritmo que propone una forma básica de combinación de modelos con diferentes parámetros a través de una estrategia de Boosting dentro del ámbito de la categorización de texto. Se emplean Naive Bayes y Support Vector Machine como clasificadores débiles.

Cada modelo entrega información útil para la decisión final, por lo que el método posee una buena capacidad de generalización. El método funciona de la misma forma que el anterior.

Existen M diferentes modelos LDA con $K_1, K_2, ..., K_M$ por lo que cada muestra del conjunto de datos tiene $\sum_{m=1}^{M} K_M$ dimensiones.

Este método es efectivo debido su buena habilidad de generalización por la introducción de los métodos de Modelamiento de Temas, y se hace uso de los clasificadores débiles a través de un marco de trabajo con Boosting.

En [**ASANA17**] tambien se utilizan Topic Models y Ensamblados para clasificar texto. Como mostramos anteriormente en [**RW12**] se usan para detectar sitios web de phishing resolviendo un problema de clasificación binaria. Esta solución obtiene haciendo selección de características para reducir el espacio de manera eficiente y Adaboosh.MH acelerado para la categorización de texto multietiqueta.

En [**ASAN15**] se trabaja usando la Bolsa de Palabras para la representación de texto. Este método aumenta dramáticamente el tiempo computacional del aprendizaje con Adaboost.MH, especialmente con datasets a gran escala. En el artículo se muestra cómo mejorar la eficiencia y la efectividad de Adaboost.MH usando temas latentes, en vez de palabras. Se usa LDA como un método de modelamiento, para estimar los temas latentes en el corpus que sirven como características en Adaboost.MH.

En [**WLC16**] se tiene una contribución que se divide en dos partes. En primer lugar, se presenta un método automático de indexación llamado Indexación con Latent Dirichlet Allocation (LDI). Para mejorar el desempeño de este algoritmo se introduce una nueva definición de vectores de probabilidad de documento basado en LDA. Además, se propone un Modelo de Ensamblado para recuperación de documentos. El Modelo de Ensamblado combina modelos básicos de indexación al asignar diferentes pesos e intenta descubrir los pesos óptimos al maximizar el Mean Average Precision (MAP).

En [**Ona18**] se realiza un trabajo de Text Mining, donde se propone un enfoque eficiente con múltiples clasificadores para categorización de texto basado en swarm-optimized topic model. Se utiliza la capacidad que tiene LDA para superar el problema de alta dimensionalidad del espacio de vectores del modelo. El enfoque de optimización swarn permite optimizar los parámetros de LDA. El enfoque híbrido, utiliza poda de ensamblado basado en medidas de diversidad combinadas y también enfoques de clustering, para obtener un sistema clasificador múltiple con un desempeño altamente predictivo y de mejor diversidad.

En [**KTMA16**] se busca categorizar becas de investigación para darles estructura al portafolio, para poder analizar, planificar estratégicamente y para tomar decisiones. Se utilizan 5 modelos de clasificación para clasificar un conjunto de BBSRC becas de investigación en 21 temas usando unigramas, términos técnicos y modelos LDA. Para mejorar la precisión se investigan métodos para combinar sus predicciones en 5 clasificadores agregados.

En [**BBM20**] se presenta un método de agregación para construir un modelo de temas combinados que está compuesto de temas de mayor coherencia en relación a los modelos individuales. En este estudio se investiga el proceso de agregar múltiples topic models que se han generado usando diferentes parámetros con el foco en la combinación general y en los temas específicos que pueda ser capaz de incrementar la coherencia. Se emplea la similaridad de

coseno y la divergencia Jensen-Shannon para calcular la similitud entre temas y combinándolos en un modelo agregado cuando este parámetro excede un umbral predefinido.

En [**ZZZ15**] la clasificación de escenas ha sido una método efectivo para la interpretación semántica remota de imágenes de alta resolución espacial (HSR). Los modelos de temas probabilísticos (PTM) han sido aplicados exitosamente a escenas naturales al utilizar una única característica (por ejemplo, característica espectral); sin embargo, son inadecuados para imágenes HSR debido a la estructura compleja de las clases que cubren el terreno. Aunque algunos estudios han investigado técnicas que combinan múltiples características, las diferentes características están usualmente cuantificadas después de una simple concatenación de estas.

En [**POO18**] se propone un enfoque de documentos usando modelos de temas del estado del arte y fusión de métodos, para enriquecer los documentos de una colección con el objetivo de mejorar la calidad del clustering de texto y su etiquetado. Se propone un espacio de modelo bi-vector en el cual cada documento del corpus se representa con 2 vectores: uno se genera basado en fusión de topic modeling, y el otro es el tradicional modelo de vectores. Los experimentos en varios datasets muestran que usar una combinación de modelamiento de temas y fusión de métodos para crear vectores de documentos, puede mejorar la calidad de los resultados en el clústering de estos.

Con el objetivo de mantener la estabilidad de los resultados de salida de los Modelos de Temas, en [**BMNG16**] se propone un algoritmo que aporta tanto en la estabilidad como en la precisión. Se propone un nuevo método ensamblado de Modelos de Temas basado en Matriz de Factorización no-negativa (NMF), el que combina una colección de Modelos de Temas que son inestables para producir una única salida. Este ensamblado combina múltiples factorizaciones de matrices para producir un único modelo de ensamblados.

El concepto del manejo de la inestabilidad también se revisa en [**BMNG17**]. En este artículo se trabaja con la inestabilidad inherente de métodos populares de Modelos de Temas, y se usa un número de medidas para obtener esta inestabilidad. Se trabaja en este problema en el contexto de la factorización de matrices para Modelamiento de Temas usando las estrategias de Aprendizaje de Ensamblado.

En [**LMP$^+$18**] se propone un Modelo de Temas supervisado utilizando LDA con mezcla de softmax. La estimación de parámetros se hace basado en una Variational Expectation Maximization (EM). Se propone un método de aproximación para clasificar datos no vistos y analizar la convergencia de la estimación de parámetros. En este método se combina tanto la semántica latente, como la clasificación con ensamblados. Corresponde al concepto de ensamblado de modelo de temas supervisados.

En [**BHA$^+$19**] se propone Latent Topic Ensemble Learning que usa un ensamblado de modelo específicos de temas para extraer data de múltiples hospitales, como un algoritmo de análisis de datos claves para predecir readmisión en hospitales. Este algoritmo se derivan temas latentes desde diferentes fuentes de información. Se utiliza para alinear datos a través de diferentes hospitales.

En [**EAGS16**] se propone un método de clasificación multiclase usando ensamblado para categorización de texto basado en 4 ideas clave:

1. Ejecutar LSI para reducir la dimensionalidad.
2. Dividir el vocabulario de manera aleatoria.
3. Hacer Bootstrap de los documentos
4. El uso de BoosTexter como learner base multietiqueta para dar soporte a la diversidad y a precisiones individuales en el comité.

La combinación del método de ensamblado y la proyección LSI muestran tener mejoras significativas en términos de Average Precision, Coverage, Ranking Loss y One Error.

En [**ZSCH16**] se propone un método integrado para ensamblado de subespacio de clustering de datos de texto sparse de alta dimensionalidad. Este método usa una representación de características de dos niveles con datos de texto (palabras y temas) para generar un clúster de subespacios. Por otro lado, se ensamblan los clústeres para hacerlos un poco más robustos. Este método depende del modelamiento de temas para lograr una representación del texto y para generar diferentes componentes de ensamblado.

En [**BBM16**] se introduce un nuevo método para agregar múltiples Modelos de Temas para producir una agregación que mejore la coherencia de los modelos individuales. Para generar un modelo de temas se debe especificar un número de parámetros. Dependiendo de los parámetros escogidos el resultado puede ser o general o muy específico. En este artículo se investiga el proceso de agregar múltiples Modelos de Temas usando diferentes parámetros. La agregación de modelos se hace usando la similaridad de coseno y la divergencia Jensen-Shannon para combina temas sobre un umbral de coherencia. El modelo se evalúa usando métodos de evaluación para calcular la coherencia de temas en el modelo base, contra esos métodos agregados.

En [**QLYL18**] se propone un nuevo método que persigue el objetivo de ensamblar múltiples NMF (non negative matrix factorization) sin ningún costo adicional de entrenamiento. Se entrena un único algoritmo de NMF con un plan de tasa de aprendizaje cíclico, el cual puede converger en algunos mínimos locales a lo largo del camino de optimización. Se guarda el resultado del ensamblado cuando este converge, y se reinicia la optimización con una tasa de aprendizaje más grande que ayude a escapar de éstos mínimos locales.

4.3. Taxonomía de Construcción de Modelos de Ensamblado

En los capítulos anteriores ya hemos revisado los conceptos y metodologías que utilizaremos para crear nuestros modelos. Ahora, mostraremos lo que se ha hecho respecto a este tema y organizaremos lo que existe de acuerdo a la taxonomía similar a la presentada por Kuncheva en [**Kun14b**]. Usando este criterio, revisaremos la literatura existente respecto a ensamblado y topic models en general.

4.3.1. Construcción de Modelos de Ensamblado Dado un conjunto de datos de n ejemplos y m características, $D = (x_i, y_i)(|D| = n, x_i \in R^m, y_i \in R)$ es un modelo de

ensamblado ϕ que usa una función de agregación G, que agrega K inductores, $f_1, f_2, ..., f_k$ y con esto predice una salida única de la siguiente forma, como se ve en la Ecuación 4.7:

$$\hat{y}_i = \phi(x_i) = G(f_1, f_2, ..., f_k), \tag{4.7}$$

donde $\hat{y}_i \in R$ se usa para problemas de regresión e $\hat{y}_i \in Z$ se usa para problemas de clasificación. Dado que esto corresponde a un marco general, el construir una modelo de ensamblado involucra, tanto la selección de metodologías de entrenamiento para los modelos participantes, como el escoger el proceso que mejor se ajuste a las salidas de los inductores.

Tal como se ve en [**Rok09**], y [**Kun14b**] los ensamblados pueden clasificarse de acuerdo a los siguiente criterios

- ***Nivel de Datos y Nivel de Características:*** esta es una buena forma de introducir diversidad en un ensamblado. Hay diferentes formas de manipular los datos:
  - **Remuestreo:** la diversidad implícita puede ser introducida usando una muestra aleatoria para entrenar cada inductor n de acuerdo a una distribución D, la cual puede ser uniforme como en *Bagging* o que puede ser creados usando una distribución sobre los ejemplos del conjunto de entrenamiento. Existen también los enfoques como *Adaboost*, que calcula la distribución de las instancias de cada inductor de acuerdo a una métrica que depende de los ensamblados previos.
  - **Particionamiento Horizontal:** el conjunto de entrenamiento se divide en algunas submuestras las cuales mantienen las mismas características que el conjunto original de ejemplos.
  - **Particionamiento Vertical (conjunto de características):** a diferencia de la subcategoria previa, cada submuestra mantiene los conjuntos de entrenamiento, sin embargo cada uno de ellos contiene un subconjunto del conjunto original de características.
- ***Nivel de Modelos o de manipulación del inductor base:*** para introducir diferencias entre los inductores, se manipulan los parámetros de estos, para agregar diversidad. Existen diferentes formas de aplicarlas.
  - **Punto de inicio del espacio de hipótesis:** dependiendo del modelo de optimización, la diversidad puede ser estimulada en el ensamblado al comenzar en el espacio de hipótesis desde diferentes puntos.
  - **Manipulación de los parámetros del inductor:** el modificar los parámetros del modelo base puede inducirse también diversidad.
  - **Combinar diferentes inductores base:** en esta categoría se induce diversidad implícita al usar diferentes inductores. La esencia de esta idea es el combinar lo mejor de cada uno de ellos.
- ***Nivel de Combinación:*** en este caso encontramos una variedad de métodos para combinar las salidas individuales de los inductores, los más comunes en la literatura son:
  - **Votos:** en la configuración de clasificación, la mayoría de votos es ampliamente estudiada para calcular la salida del ensamblado.

- **Combinación lineal:** la decisión conjunta se obtiene a veces como una combinación lineal de las salidas individuales.

4.3.2. Entrenamiento de los modelos Hay muchas formas de entrenar los modelos para alcanzar la salida deseada. Sin embargo, hay algunos principios claves que se deben considerar cuando se genera un ensamblado:

1. **Diversidad:** el gran desempeño que muestran los modelos de ensamblado se alcanza principalmente debido al uso de varios sesgos inductivos [**DRTV13**]. Así los inductores participantes deben ser lo suficientemente diversos para obtener el desempeño predictivo deseado.
2. **Desempeño Predictivo:** el desempeño predictivo individual del inductor debiese ser tan alto como es posible y al menos tan bueno como un modelo aleatorio.

Estos dos principios parecen contradecirse uno con el otro a primera vista. Además, los ensamblados con inductores diversos no siempre mejoran el desempeño predictivo [**Bi12**]. La principal idea es combinar inductores predictivos con errores no correlacionados en un ensamblado, ya que ha sido teórica y empíricamente demostrado que el desempeño predictivo de un ensamblado completo tiene correlación positiva con el grado de error cometido por inductores individuales, cuando estos son no correlacionados [**Ali96**]. Existen metodologías que se usan para alcanzar la meta de incluir inductores diversos:

- **Manipulación de Entradas y Particionamiento (Nivel de Datos y Características)**: en este enfoque cada modelo base se ajusta usando un diferente conjunto de entrenamiento, por lo que se usa una variedad de entradas en los diferentes modelos base. Este método ha mostrado ser efectivo en casos donde pequeños cambios del conjunto de entrenamiento pueden resultar en modelos completamente diferentes. En la implementación más simple de esto, cada modelo se entrena usando una muestra ligeramente diferente. La distribución de las clases entre los diferentes inductores puede ser aleatoria o determinada de acuerdo a la distribución de clases del conjunto de datos completo [**CS95**].

  La diversidad se puede alcanzar dividiendo el conjunto de datos original en pequeños subconjuntos y usar cada subconjunto para entrenar diferentes inductores. En la partición horizontal, se divide el conjunto de datos original en subconjuntos que incluyen todo el conjunto de características para que el inductor solo pueda diversificar todas sus instancias [**CCH$^+$04**]. En cambio, el particionamiento vertical funciona de la manera opuesta ya que cada inductor usa las mismas instancias pero con diferentes características [**Rok08**]. El método de agregación de subespacio de características [**TWT$^+$11**] divide el espacio de características en regiones locales mutuamente excluyentes que se definen sobre un numero fijos de atributos. En [**BSS16**] se subdivide el espacio de atributos en regiones locales. Este método prioriza las particiones que resultan en regiones consistentes.

Por otro lado, existen algunas configuraciones dentro del contexto de Topic Modeling y Ensamblado. En [**RC13**], por ejemplo, se utilizan diferentes fuentes de datos médicas que son particionadas y entregadas a cada uno de los Inductores. En [**SLYS10**] se hace una separación de los documentos en conjuntos disjuntos y cada uno de ellos le son entregados a los inductores. Por el contrario, lo que se hace en [**STGCL12**] cada conjunto de datos es separados en las diferentes características y cada uno de los inductores se especializa en ellas. En el caso de [**RW12**] en vez de usar los documentos, se utilizan los conjuntos de temas y sus distribuciones correspondientes como entrada del ensamblado. En [**GLJG16**] se utiliza el concepto de *Conjunto de Datos Textual* y esto sirve como entrada del ensamblado. En [**ASANA17**] se hace algo algo diferente, ya que las palabras, que son las características del texto, se seleccionan de acuerdo a un umbral de peso que tiene dentro del conjunto de documentos. En [**ASAN15**] se hace una selección doble en relación a los datos. Primero, para cada documento se seleccionan los temas con mayor ponderación dentro del conjunto y luego se seleccionan los mayores temas por categoría. En [**RW12**], [**GLJG16**], [**WG14**], [**ASANA17**] se utiliza un criterio de muestreo del conjunto de documentos original, muy al estilo Adaboost, donde se le dan ponderaciones al conjunto de documentos. En [**WLC16**] [**POO18**] se desarrolla un esquema de muestro donde se da una distribución de probabilidad a los documentos. En cada iteración los valores de esta distribución se actualizan. En [**ZZZ15**] se adopta una estrategia diferente y las imágenes que sirven como datos de entrada para el modelo se dividen y se muestrean y a cada una de estas imágenes se les obtienen características particulares. Finalmente podemos ver como en [18] se obtienen datos de múltiples fuentes a partir de las muestras (bases de datos de hospitales).

- **Manipular el Algoritmo de Aprendizaje (Nivel de Modelos):** en este enfoque, se modifica el uso de cada modelo base. Una manera de hacer esto es manipular la manera en la cual el modelo base se mueve por el espacio de hipótesis. Esto se hace llevando el modelo base a diferentes caminos de convergencia [**BWT05**]. Por ejemplo, cuando se construye un ensamblado de *árboles de decisión* se puede inyectar aleatoriedad sacando uno de los k mejores atributos de cada separación. Por otro lado, el distribuir los vecinos [**MRGO09**] es un método que expande el espacio de características al generar diferentes combinaciones de las que existen en el conjunto original. Otra forma de crear diversidad de esta manera es entrenar el modelo base con variados valores de hiperparámetros [**LC12**]. Por ejemplo , el entrenar una red neuronal con diferentes tasas de aprendizaje, diferentes número de capas o atributos.

  En relación a Modelos de Temas y Ensamblado, en [**RC13**] los modelos son creados a partir de un ajuste que se le hace a los parámetros. En [**SLYS10**] se crean los Modelos de Temas con diferentes parámetros y con diferentes técnicas de iniciación. En [**LQQQ**], [**RW12**],[**GLJG16**],[**WG14**],[**ASANA17**], [**ASAN15**], se utiliza LDA como técnica de Modelamiento de Temas, estos se entrenan con diferentes parámetros de este para obtener diferentes conjuntos de temas.

Por otro lado, en [**WLC16**] se utiliza Indexación usando LDA como modelo de temas. En [**Ona18**] se optimizan los parámetros de LDA usando un enfoque heurístico de colmena con diferentes métodos hasta encontrar la adecuada combinación de parámetros. En [**ZZZ15**] se utiliza los modelos LDA y PLSA para capturar la información semántica de las imágenes. En [**BMNG16**] se crea un conjunto base de modelos de temas corriendo múltiples NMF la que se aplica a la misma matriz término documento. En [**LMP$^+$18**] se utiliza LDA, SLDA y Softmax (multinomial). En [**EAGS16**] se utiliza una combinación de una construcción de ensamblado basado en rotación y una proyección de Indexación semántica latente. En [**ZSCH16**] se aprenden clusteres de cada matriz de topic models.

- **Manipulación de las Salidas (Nivel de Combinación):** esto se refiere a la técnica que combina numerosos clasificadores binarios en un único clasificador multiclase. *Error-correcting output codes* (ECOC) es un ejemplo de esta aproximación [**DB95**]. En este método, cada clase se codifica como un palabra de L bits donde L es el número de clasificadores participantes en el ensamblado. El propósito de cada clasificador es predecir un bit L del código. A los clasificadores se les aplica entonces una nueva instancia para generar el strings de L bits que representa la predicción. La clase escogida, que se va a predecir por la instancia dada, es la clase en la cual su código es el más cercano a la cadena de instancia. La cercanía puede ser medida usando diferentes métodos, como las distancias Euclidania y la distancia de Hamming. El ECOC Adaptivo (AECOC) extiende ECOC al agregar procedimientos de reducción convencionales cuando se entrenan los diferentes clasificadores binarios [**ZC13**]. Un esquema de codificación de corrección de error N-aria es un método desarrollado recientemente que divide el clasificador multiclase en subproblemas más simples [**ZThM16**].

  En relación a Topic Modeling y Ensamblados, en [**RC13**] se utiliza la estrategia de combinar todas las salidas de los modelos base en una única matriz Phi. En [**SLYS10**] se usa la unión para combinar temas base que se desprenden de los modelos. En [**STGCL12**] y [**LQQQ**] se usa la estrategia del voto de cada uno de los clasificadores débiles para obtener un modelo final. También en [**Ona18**] se utiliza mayoría de votos. En [**WG14**],[**ASANA17**], [**ASAN15**] se utiliza la estrategia de combinación lineal ponderada donde cada uno de los modelos aporta con un peso de acuerdo a que tan bien ajustados están los datos. En [**WLC16**] se utiliza MAP como métrica de desempeño y para poder ponderar las salidas de los modelos. En [**POO18**] se realiza una combinación de clusters y se obtienen los mejores elementos. En [**BMNG16**] se produce un único topic model al mezclar los obtenido por las NMF individuales. En [**LMP$^+$18**] el modelo final se forma a partir de una disjunción de modelos Softmax. En [**BHA$^+$19**] se utiliza Soft Majority Voting.

En este capítulo revisamos los elementos del Aprendizaje de Ensamblado, los fundamentos que sustentan esta metodología y las principales alternativas y formas de combinar que existen actualmente, como Bagging y Boosting. Estas metodologías pueden mezclarse con diferentes formas de Modelos de Temas, con el objetivo de resolver diferentes tareas. Entre ellas están Ranking de Documentos y la Recuperación Adhoc. Pero estas tareas no se encuentras muy exploradas en los que se relaciona con Ensamblado de Modelos de Tema ni tampoco podemos encontrar un marco de trabajo que formalice estas técnicas de manera general. En el siguiente capítulo se proponen metodologías trabajan sobre estas tareas en particular y se busca presentar una familia de modelos para este fin.

Capítulo 5

Propuesta

En los capítulos anteriores hemos revisamos los conceptos y metodologías que nos servirán para crear nuestra propuesta de Ensamblado de Modelos de Temas para Recuperación de Información Adhoc.

En primer lugar, hemos visto las técnicas que nos permiten recuperar documentos desde una colección documental, el que está modelado como una distribución de probabilidad (Modelo de Lenguaje), al cual se le aplican unas consultas, que supondremos, provienen de esta misma distribución y se realiza una recuperación de documentos importantes para esta consulta. Luego, se tienen los modelos de temas, los cuales nos permiten encontrar la estructura subyacente de temas de una colección documental. Con estos métodos podemos disminuir la complejidad de un conjunto documental y modelar todo a través de variables latentes. Estos modelos son otra forma de representar el modelo de lenguaje de nuestra colección documental y que agrega información adicional a este (estructura de temas subyacente).

Continuamos con el aprendizaje por ensamblados que ha demostrado ser una estrategia que mejora el desempeño de los modelos, ya que permite la disminución del error de generalización. La disminución de este error tiene que ver con la reducción del sesgo y la varianza que poseen las máquinas. Esta compensación de sesgo/varianza puede explicarse en términos de la inestabilidad que tienen los algoritmos. Al presentar inestabilidad, un modelo, puede registrar mayores variaciones de desempeño en relación al conjunto de aprendizaje. En este sentido una máquina es inestable cuando se sobreajusta [**PBS16**]. El aprendizaje con ensamblados aprovecha la inestabilidad de los modelos. En vez de evitarla se realiza una combinación de todos estos learners base que son inestables y obteniendo uno mejor [**VM02**].

Como método base para nuestros ensamblados usaremos LDA, el cual, como vimos en el capítulo 3, necesita como parámetros, los supuestos distribucionales y la cantidad de temas, para poder crear los modelos y dependiendo de cuales escojamos, obtendremos el modelo de salida (diferentes representaciones del corpus). Esto lo vuelve un learner inestable y entrega diferente información respecto al conjunto de documentos, tanto a nivel de vocabulario como a nivel de temas.

En este sentido, como vimos en el capítulo 4 de Ensamblado, también tenemos muchas formas de combinar estos modelos. En primer lugar, podemos dividir el corpus, en conjuntos disjuntos y entregar cada porción a un learner. Una vez obtenidas las representaciones de estos subcorpus, podemos crear una nueva representación del corpus general [**SLYS10**]. Por otro lado, existen otras formas vistas en el capítulo 4 de Ensamblados. Primero, tenemos Bagging [**BB96**] el cual produce un conjunto de muestras a partir de conjunto de datos original, induciendo variabilidad a partir de la aleatoridad usando boostraping. Entonces, el mismo learner se aplica a cada versión del conjunto de dato, produciendo un enasmblado homogeneo. Finalmente, para Boosting [**Sch90**], se genera un proceso de aprendizaje secuencial, ejecutando una ponderación de los elementos de acuerdo al error de modelamiento del conjunto de datos generado. Mientras menos similar al elemento original, mayor ponderación tendrá dentro del conjunto.

En cada una de estas configuraciones de ensamblado, donde los modelos de temas se realizan con diferentes muestras, se crearan modelos de lenguaje probabilísticos distintos y con estos modelos realizaremos consultas, utilizando el modelo de verosimilitud de estas para ver cuales documentos son relevantes a esta y realizaremos la recuperación de documentos usando la ecuación de suavizado de Dirichlet vista en [**WC06b**], basada en LDA, usando la información del modelo original, la información el modelo del vocabulario y el modelo obtenido con los temas. Estos serán ordenados de acuerdo a su importancia y realizaremos un ranking de documentos.

Una vez obtenidos todos los rankings del ensamblado, los integraremos en un ranking de consenso usando el método CombMNZ [**WBM07**] visto anteriormente.

En las siguientes secciones presentamos los algoritmos antes mencionados, pero utilizando LDA como learner base.

5.1. Ensamblados de Modelos de Temas para Ranking de Documentos

Este esquema propone ensamblar diferentes modelos de temas que luego serán utilizados para realizar ranking de documentos. En la Figura 5.1 podemos ver las diferentes etapas de manera general de nuestra propuesta. Primero, se comienza con el corpus completo con todos los documentos de la colección. Luego, se construyen muestras de estos documentos, los que son entregados a cada uno de los Modelos de Temas. Esta muestra se transforma en una *Matriz Término Documento*. Esta matriz es la que sirve como entrada y es modelo teórico de conjunto de documentos. Teniendo todo lo anterior en cuenta, creamos el modelo LDA usando los parámetros de entrada k, α, y β.

Luego de crear el modelo, se obtienen las dos matrices, ϕ y θ, correspondiente a la *distribución de las palabras dados los temas* y la *distribución de los temas dados los documentos* de la muestra. Estas dos matrices luego se componen y se obtiene una *Matriz Término Documento* la que corresponde al modelo empírico. Lo descrito anteriormente, tiene como supuesto de que el Modelo de Temas va a generar un modelo de lenguaje de la colección documental,

como se explicó en el capítulo de Recuperación, y en este caso cada uno de los métodos del ensamblado generará un modelo de lenguaje distinto que dependerá de la muestra con la que se esté trabajando.

Con el modelo obtenido anteriormente, y con el conjunto de consultas y la lista de documentos relevantes a esta consulta, podemos realizar la tarea de ranking. Esto es posible, también bajo el supuesto descrito anteriormente, donde la probabilidad de generar la consulta depende del modelo de lenguaje que estemos usando, entendiendo que la consulta se genera del mismo modelo.

Finalmente, fusionamos cada una de las listas de ranking obtenidas a partir de todos los modelos y obtenemos una lista de consenso.

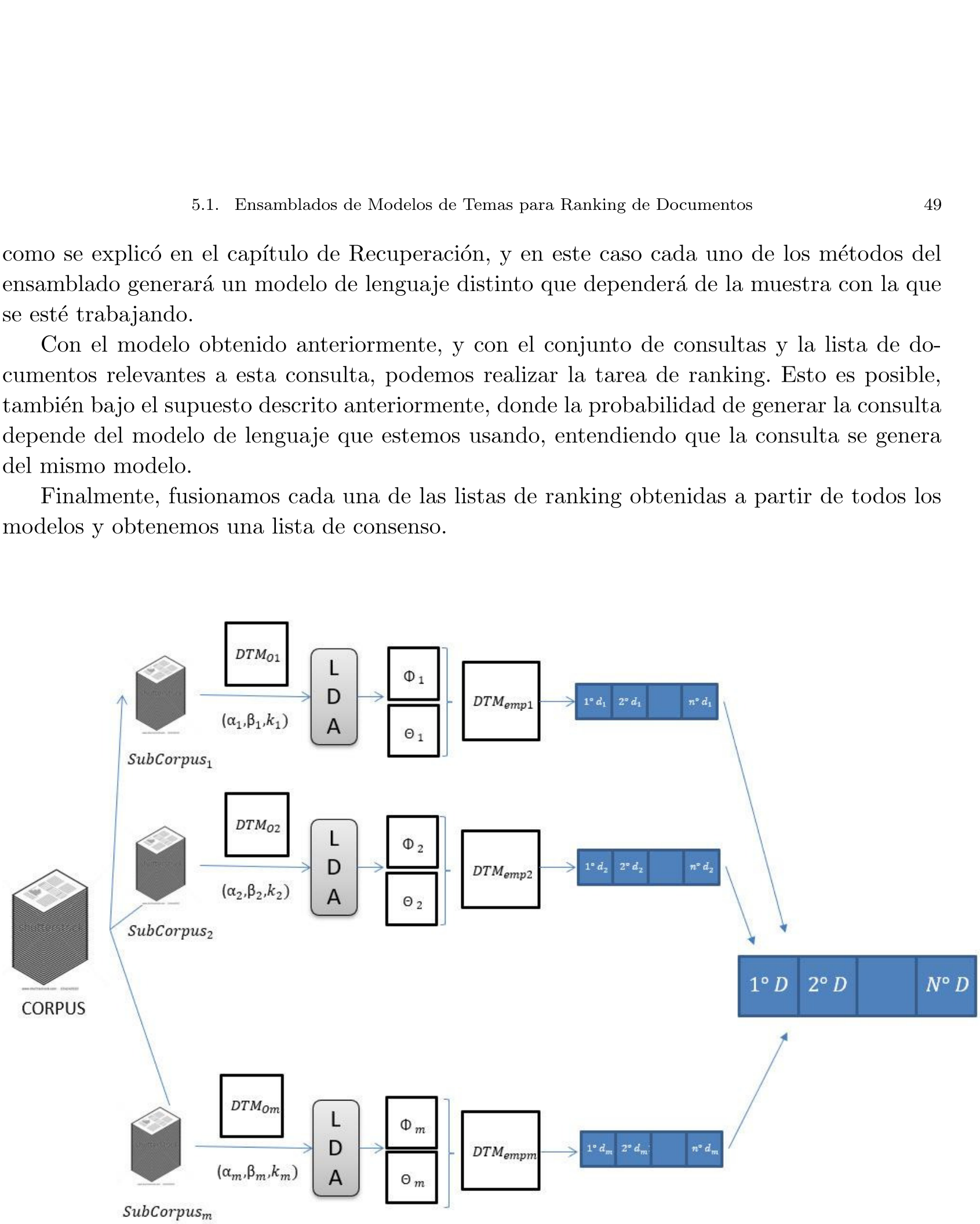


FIGURA 5.1. Esquema Propuesto de Ensamblado de Modelos de Temas para Recuperación de Información Adhoc.

En las siguientes secciones mostraremos las variantes propuestas para estos ensamblados. Estas propuestas se diferencias en alguna de las etapas del ensamblado, como vimos en el Capítulo 4. Realizamos variaciones en el conjunto de entrada de cada modelo, cambiando la forma en que creamos las muestras. Luego realizamos modificaciones en la creación de modelos, realizando modificaciones en los parámetros de entrada. Una vez obtenidos los modelos de

temas, realizamos variaciones de estos al utilizar el parámetro de coherencia y así filtrar algunos de los temas obtenido. Luego, realizamos una de las tareas de ranking sobre el conjunto de documentos y a partir de las consultas y las listas de relevancia, creamos las listas ordenadas de los documentos. Finalmente las listas se fusionan para obtener el resultado final.

5.2. Ensamblado sin Muestreo

En primer lugar usaremos un ensamblado que no utiliza muestreo para crear los modelos. A este algoritmo lo llamaremos **Ensamblado de LDA sin Muestreo del Corpus**. Este método divide el corpus de cada modelo de tema a partir del cual se crean los rankeadores, algo similar a lo que se hace en [**SLYS10**]. Esta forma de ensamblar tiene varias aplicaciones, entre las que se encuentran trabajar con corporas muy grandes, como por ejemplo páginas web o datos incrementales como las noticias.

Este método comienza dividiendo el conjunto original de datos en segmentos distribuídos, cada uno correspondiente a los datos del subcorpus. Entonces para cada segmento aprendemos los temas base, usando LDA, con un número k determinado, para la muestra de temas.

El modelo de temas obtenido se compone de las matrices ϕ y θ para cada uno de los subcorpus. Luego, vamos a componer cada una de estas matrices, y así obtener la matriz término documento, $P(w/d)$, para cada uno de los subconjuntos. Finalmente utilizando la matriz término documento original de los datos, $P(w|d)_O$, la matriz término documento obtenida por la composición, $P(w|d)_{LDA}$ y el conjunto de consultas asociados al corpus Q y realizaremos el ranking correspondiente. Podemos ver la descripción paso a paso en el Algoritmo 5.

Algoritmo 5 Ensamblado LDA con conjuntos disjuntos para Ranking de Documentos

1: Dado $\{d_1, d_2, ..., d_D\}$ donde $d_i \in D$, con D como el corpus entero.
2: **para** t= 1,...,T **hacer**
3: Obtener un conjunto de muestra s_t de manera que no se repita ningún elemento de los conjuntos de muestra anteriores.
4: Crear un modelo LDA_t, para un número de temas k determinado, con el conjunto de documentos s_t.
5: Obtener las matrices ϕ_t y θ_t del modelo LDA_t y componer una nueva matriz término documento DTM_t.
6: A partir de la matriz término documento obtenida y la matriz término dcumento original del conjunto s_t, generar un ranking Rk_t a partir del conjunto de Querys.
7: **fin para**
8: Finalmente generar un ranking final uniendo los rankings obtenidos por el ensamblado y obteniendo un ranking final RK:

$$RK = \cup_{t=1}^{T} Rk_t \tag{5.1}$$

Esta forma de ensamblar se justifica por varias razones. Primero el conjunto de datos puede ser demasiado grande y los topic models a gran escala pueden volverse intratables. Por otro lado, los corpora, como por ejemplo, los artículos y las noticias, crecen rápidamente con el tiempo y los modelos de temas necesitan tener acceso a todo el conjunto de documentos, incluidos datos viejos y nuevos.

En este caso las muestras se crean sin ningún supuesto distribucional sobre los documentos. Cada modelo de lenguaje se crea usando un subconjunto del corpus original y se usa como supuesto desde donde se crean las consultas que permiten realizar el ranking de documentos de acuerdo a cierta relevancia.

5.3. LDA Bagging: Ensamblado de LDA con muestro uniforme

Como vimos en el capítulo 4 de Ensamblados, Bagging es un método de que fue creado para la tarea de clasificación. Este método permite crear muestras boostrap, de manera aleatoria con reemplazo a partir de una distribución uniforme, del conjunto original de datos y a partir de estas muestras, se pueden entrenar los modelos base, los que serán fusionados mediante alguna técnica y realizar la tarea de mejor forma. La idea es trabajar con un conjunto de tamaño similar al corpus o una porción de este, pero que el modelo tenga acceso a la mayor cantidad de documentos y así darle diversidad que necesitan.

En nuestra propuesta, el objetivo principal es poder realizar recuperación de información utilizando esta misma configuración, pero aplicada a esta tarea. Más precisamente, buscamos realizar un *metaranking* de documentos, fusionando cada una de las listas de ranking que se obtienen de los modelos del ensamblado. Supongamos que tenemos una colección documental $D = \{d_1, d_2, ..., d_D\}$ donde D es un conjunto de documentos. Se puede ejecutar un modelo de temas, tantas veces como queramos, es este caso, modificando los documentos de entrada. Estas muestras tienen propiedades estadísticas que justifican su uso: primero, estas se obtienen directamente del corpus inicial del cual se desconoce la distribución de temas y estas muestras son independientes unas de otras. Estas muestras se consideran representativas del conjunto documental. Siguiendo la idea del Bagging original, crearemos muestras desde D, las que llamaremos, s_t. Para obtener estas muestras, asumiremos que cada uno de los documentos tiene la misma probabilidad de ser escogido. Una vez obtenida la muestra usaremos este subcorpus para construir un modelo de temas, LDA_t, usando un número k_t y con esto obtener la distribución de los temas sobre el subvocabulario, ϕ_t, y sobre el subconjunto de documentos, θ_t.

Con estas matrices podemos componer una nueva matriz término documento, DTM_t, de la distribución del subvocabulario sobre el subcorpus. Entonces, con la distribución original, $P(w|d)_o$ y la distribución empírica $P(w|d)_{LDA_t}$ podemos realizar una lista de documentos que hemos obtenido a partir del conjunto de consulta. Finalmente, con cada una de las lista de ranking, R_{k_t}, haremos una fusión de ellos y obtendremos el ranking final de consenso R_K. La descripción del algoritmo podemos verla en el Algoritmo 6.

Algoritmo 6 Bagging para Ranking de Documento

1: Dado $\{d_1, d_2, ..., d_D\}$ donde $d_i \in D$, con D como el corpus entero.
2: **para** t= 1,...,T **hacer**
3: Obtener un conjunto de muestra s_t, el que se obtiene del conjunto D, con reemplazo a partir de un corpus uniforme.
4: Crear un modelo LDA_t, para un número de temas k determinado, con el conjunto de documentos s_t.
5: Obtener las matrices ϕ_t y θ_t del modelo LDA_t y componer una nueva matriz término documento DTM_t.
6: A partir de la matriz término documento obtenida y la matriz término dcumento original del conjunto s_t, generar un ranking Rk_t a partir del conjunto de Querys.
7: **fin para**
8: Finalmente generar un ranking final uniendo los rankings obtenidos por el ensamblado y obteniendo un ranking final RK:

$$RK = \cup_{t=1}^{T} Rk_t \tag{5.2}$$

El objetivo principal de esta propuesta es la de encontrar una mejor lista de ranking de documentos, a partir de ciertos criterios de relevancia, que los que podemos obtener con los métodos individuales además de reducir la varianza de los modelos debido a las modificaciones en el corpus de entrada (cambios en el corpus, producen modificaciones en el conjunto de temas). Cuando utilizamos LDA, para cierto número de temas sobre un corpus, se vuelve complicado poder escoger la adecuada. Esto significa que para ciertas aplicaciones, LDA es un algoritmo inestable y sensible a los parámetros. Recordemos que los parámetros de LDA definen ciertos supuestos distribucionales y su característica estocástica lo vuelve impredecible. Nuestra propuesta busca obtener diferentes listas de documentos, a partir de muestras del conjunto documental, y el modelo de temas asociado a este conjunto, para mejorar la recuperación del conjunto original, en relación a la métrica propuesta.

Como las muestras se crean bajo el supuesto de que los documentos se distribuyen aleatoriamente vamos a suponer que los modelos de lenguajes probabilísticos son parecidos entre las muestras y supondremos también que las consultas provienen de esta misma distribución de probabilidad.

5.4. LDA Adaboost: Ensamblado de LDA de muestreo adaptativo

Al igual que en la sección anterior y como se presentó en el capítulo 4 de Ensamblados, Adaboost es la segunda implementación que usaremos. En este caso, se realizan modificaciones a la distribución del conjunto de documentos. Este método es un modelo de ensamblado secuencial donde los pesos dependen de los modelos previos y se ha demostrado empíricamente que este algoritmo tiene buen desempeño en términos de clasificación.

Como su nombre lo indica, este algoritmo se va a adaptando a medida que las iteraciones van pasando y los nuevos métodos se van creando. En este sentido, las instancias mal clasificadas obtienen grandes pesos. Con esto se busca reducir el sesgo y la varianza en el sentido del aprendizaje supervisado. En nuestra propuesta usaremos una configuración similar. En el algoritmo 7, podemos ver la propuesta. Este método también fue creado para la tarea de recuperación de información. Para lograr esto, se propone una modificación de la distribución de los documentos usando la Divergencia Kulback Leiber como medida de desempeño del modelo y poder hacer comparaciones con los documentos originales. Como sabemos, esta métrica compara 2 distribuciones de probabilidad y mientras más cercano a 0 sea el valor, más se parecen ambas.

Cuando hablamos de documentos, nos referimos a la distribución de probabilidad de las palabras en este documento. En este sentido, cada documento del corpus es una distribución y cuando ejecutamos el modelo de temas, se obtienen dos distribuciones desde él. Una de las distribuciones es la de los temas sobre los documentos y la otra, la distribución de las palabras dados los temas. Estos conceptos están representados mediante las matrices, ϕ y θ respectivamente. Como ya hemos dicho, si componemos ambas matrices, tenemos la matriz término documento que se obtiene desde este modelo. En ella se encuentran modelados los mismos documentos del corpus original, por lo que podemos hacer la comparación entre lo entregado por el modelo y éste.

Entendiendo lo anterior, describiremos el algoritmo como sigue. Dado un corpus D, se le asigna a estos documentos la misma probabilidad inicial $\frac{1}{D}$. A partir de esta distribución, se crea una muestra de documentos, s_t, desde el corpus original, el cual se modela como la matriz término documento original. Con esta muestra, usando LDA, se crea un modelo de temas para un k determinado. Con esto obtenemos las matrices ϕ y θ, las que una vez compuestas, permiten obtener la nueva matriz término documento empírica.

Estas dos matrices son comparadas y con ello podemos saber cuáles documentos fueron mejor modelados por el método, usando la divergencia Kullback-Leiber. Con este valor podemos recalcular la distribución de probabilidad de los documentos, dando mayor probabilidad a los que fueron mal modelados por los temas y este número de iteraciones se va repitiendo, hasta alcanzar cierto criterio. Con cada uno de los modelos obtenidos, se realiza un ranking usando la ecuación de Dirichlet propuesta en [**ZL01**]. Finalmente, estos ranking se fusionan para obtener un metaranking final.

Algoritmo 7 Adaboost para Ranking de Documento

1: Dado $\{d_1, d_2, ..., d_d\}$ donde $d_i \in D$, con D el corpus.
2: Inicializar la distribución $D_1 = \frac{1}{D}, \forall j = 1, ..., d.$
3: **para** t= 1,...,T **hacer**
4: El conjunto de muestra s_t se obtiene desde el corpus con distribución D_t.
5: Crear un modelo LDA_t para un número específico k, con el conjunto de documentos de muestra s_t.
6: Obtener ϕ_t y θ_t desde el modelo LDA_t y componer la matriz término documento DTM_t.
7: **para** d=1,...D **hacer**
8: Comparar el documento original d_o con el documento obtenido por el modelo d_{emp}. Ambas distribuciones sobre las palabras del documento se comparan usando la Divergencia Kullback-Leiber KLD_d, como parámetro de confianza.
9: Actualizar la distribución de los documento en el corpus como

$$D_{t+1}(i) \quad = D_t(i)e^{KLD_d} \tag{5.3}$$

10: **fin para**
11: **fin para**
12: Finalmente, se crea un ranking final fusionando los rankings obtenidos por el ensamblado.

$$RK = \cup_{t=1}^{T} Rk_t \tag{5.4}$$

Los métodos de boosting trabajan de la misma forma que Bagging. Se construye una familia de modelos que se van agregando para obtener un modelo más robusto que tenga mejor desempeño. Esto lo logramos desarrollando un algoritmo que se vaya ajustando de una forma adaptativa. Al contrario del caso anterior, esperamos obtener modelos menos sesgados debido a los muchos modelos que podemos obtener con los diferentes subcorpus y con los diferentes parámetros de entrada. En cada iteración los documentos que menos se asemejan al documento original, utilizando la divergencia Kullback Leiber, obtienen una mayor ponderación en las siguientes iteraciones. Obtenemos un parámetros de confianza a partir de una comparación hecha del subcorpus original versus el subcorpus obtenido por el modelo, con subcorpus y vocabularios que no fueron bien modelados en las iteraciones anteriores. Esto permite que estos documentos tengas más opción de ser recuperados por los métodos de ranking.

El supuesto aquí es diferente. En cada iteración las distribuciones de los documentos va cambiando, y las muestras se van creando a partir de estas distribuciones. A medida que agregamos modelos al ensamblado, los documentos mal modelados van apareciendo más frecuentemente, por lo que los modelos de lenguaje se hacen a base de estos documentos. Al ejecutar un modelo de temas creamos distribuciones, que en conjunto con el modelo original, nos permitirán hacer recuperación de documentos con las consultas.

En este capítulo se presentaron las diferentes técnicas propuestas, cada una con sus diferencias. En este sentido cada una tiene sus ventajas y desventajas al tratar el ensamblado, ya sea con los datos de entrada, o la generación de diversidad, o la forma de combinar las salidas. Debido a esto se plantearon diferentes alternativas para generar distintas configuraciones, las que serán contrastadas con el método base que ofrece la literatura, LDA. En el siguiente capítulo, veremos el desempeño de estos algoritmos con diferentes conjuntos de datos de la literatura, para ver como se comportan los métodos de acuerdo a los diferentes escenarios.

Capítulo 6

Experimentos

En esta sección vamos a medir la calidad de los algoritmos propuestos, al presentar los diferentes experimentos diseñados para evaluar las características de ellos.

En primer lugar nos enfocaremos en comparar estas técnicas con los modelos base LDA y TFIDF. Buscaremos el mejor desempeño para un número de temas determinado y usando ese valor como base. Para ver el desempeño de los ensamblados, variaremos su tamaño (número de modelos). Estos 5 métodos serán usados en la tarea de recuperación de información adhoc y revisaremos sus desempeños.

Para todos los algoritmos de ensamblado, el método base para realizar los modelos de temas será LDA. Como ya dijimos, esta técnica usa 3 parámetros: k, el supuesto número de temas; α, que es el supuesto distribucional de dirichlet de los temas, y β, que es el supuesto distribucional del vocabulario. Usaremos los valores propuestos en [**RZGSS04**], para $\alpha = \frac{50}{k}$ y $\beta = 0{,}01$.

Es importante notar que el objetivo no es demostrar que los algoritmos propuestos tienen el mejor desempeño siempre, sino que es el hacer una comparación entre las técnicas y mostrar el desempeño de estas en diferentes escenarios. Además buscamos demostrar si es que los supuestos distribucionales hechos sobre los modelos de documentos y las consultas usadas pueden aplicarse al momento de ensamblar.

Para evaluar los métodos se realizarán experimentos en 4 colecciones estándar de documentos (CACM, CISI,CRAN,MED) [**Col19**] para los cuales están disponibles las consultas y los juicios de relevancia. Se aplican todos los métodos (LDA, TFIDF, y los 3 ensamblados) y compararemos el desempeño de recuperación en cada colección.

Los documentos de cada base de datos están indexados con los términos que existen en el título y en el abstract, pero que no existen en la lista de palabras comunes. Las palabras de las consultas están escritas en lenguaje natural y estos términos se usarán sólo si no aparecen en la lista de términos comunes y aparecen en al menos 1 documento. Los términos se lematizarán para disminuir el tamaño del vocabulario.

En la Tabla 1 se pueden ver las características de cada colección:

Datos	Documentos	Consultas	NÂ° Términos
MED	1.033	30	5.775
CRAN	1.400	225	8213
CISI	1.460	112	10.170
CACM	3.204	64	9.961

Cuadro 1. Características de los Datos

Como medidas de evaluación usaremos precisión y recall para evaluar la relevancia de los documentos obtenidos. Recordemos que estos se definen de la siguiente forma:

$$Precision = \frac{|positivos|}{|recuperados|}, \tag{6.1}$$

$$Recall = \frac{|positivos|}{|relevantes|}, \tag{6.2}$$

donde *relevante* representa el conjunto de documentos que son relevantes para la consulta y *recuperados* representa el conjunto de documentos que son recuperados y los positivos pueden entenderse como $relevantes \cap recuperados$. La precisión es una medida de exactitud o calidad, mientras que recall es una medida de completitud o cantidad. Se busca tener un buen sistema de recuperación que tenga un buen balance entre calidad y cantidad entre las respuestas, donde hay una compensación entre precisión y recall en la recuperación de documentos.

Para poder resolver este problema es que usamos la medida F1, que como se definió en el capitulo 2, es una forma de combinar los valores de precisión y de recall, y se define como la media armónica entre estos valores de desempeño del modelo. Usaremos la fórmula estándar de F1 definida en la siguiente fórmula:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}, \tag{6.3}$$

Finalmente usaremos mean Average Precision (Map) para medir el desempeño de los modelos. Entonces, para un conjunto de consultas tenemos:

$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}, \tag{6.4}$$

donde Q es el número de consultas en el conjunto y AveP(q) es la precisión promedio para cada una de las consultas dadas.

6.1. Resultados

En esta sección describiremos el estudio empírico relacionado con el diseño de experimentos trabajado anteriormente. En este sentido los gráficos y tablas reportados son el resultado de los experimentos realizados. Se puede revisar el detalle de estos en el Anexo A.

En relación al **mAP** podemos decir que se obtienen buenos resultados, especialmente en los ensamblados donde se realiza remuestreo. Notamos que el ensamblado funciona muy bien en relación a esta métrica cuando lo evaluamos con listas pequeñas, por lo que se puede observar que para listas de esta forma, la precisión promedio en general supera al algoritmo base, como por ejemplo en la Figura 6.1 donde los mejores resultados los obtienen el método de muestreo uniforme o en la Figura 6.2 donde el mejor resultado lo obtiene el método de muestreo adaptativo. Por el contrario, cuando tratamos de recuperar listas más grandes (@15, @20), LDA supera a los algoritmos propuestos. En el caso del dataset CACM, se obtuvo un muy buen resultado en la primera iteración. Finalmente el método de ensamblado supera en precisión promedio a TFIDF en todas las listas de recuperación.

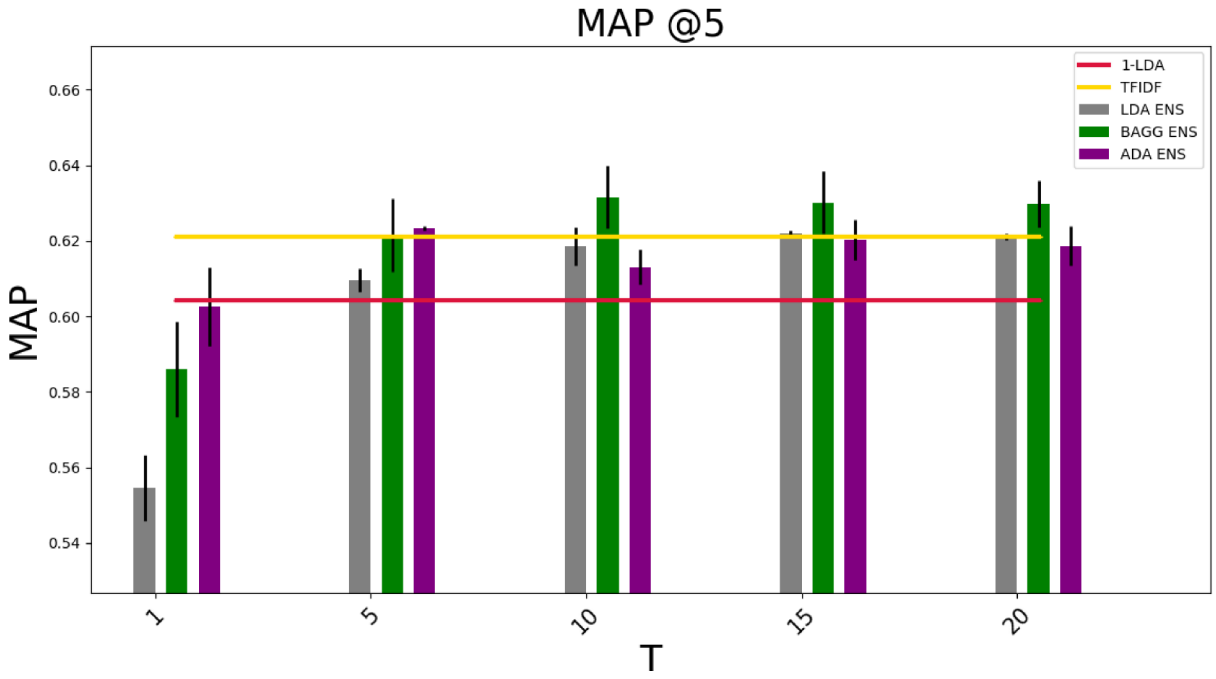


Figura 6.1. CRAN MAP 5

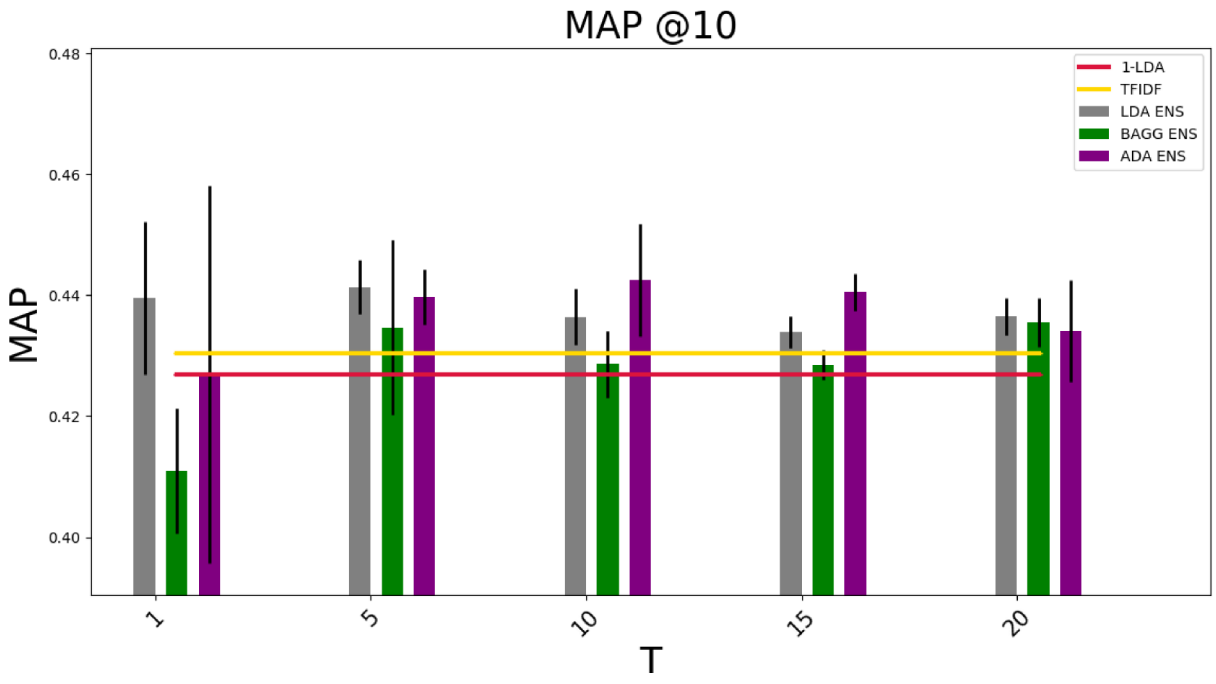


FIGURA 6.2. CRAN MAP 5

En relación a **Precision** se puede ver en las imágenes del anexo que en todos las colecciones documentales, para el caso @5, se supera a los métodos base. En el caso de CISI para lista de 5 elementos, se ve que el mejor valor se obtiene en T=10, y luego la métrica empeora, como puede verse en la Figura 6.3. Se aprecia que en CACM se supera los métodos base para @15 y @20. Para listas de ranking pequeñas vemos que el ensamblado tiene buenos resultados, lo que significa que en general, del conjunto de documentos recuperados los ensamblados obtienen más documentos que son relevantes para la consulta. Cuando comparamos los ensamblados podemos ver que el método de muestreo uniforme (LDABagging) casi al final del ensamblado, para T=20, supera a los métodos base, a excepción de CACM, que obtiene su mejor resultado para @5 en T = 5.

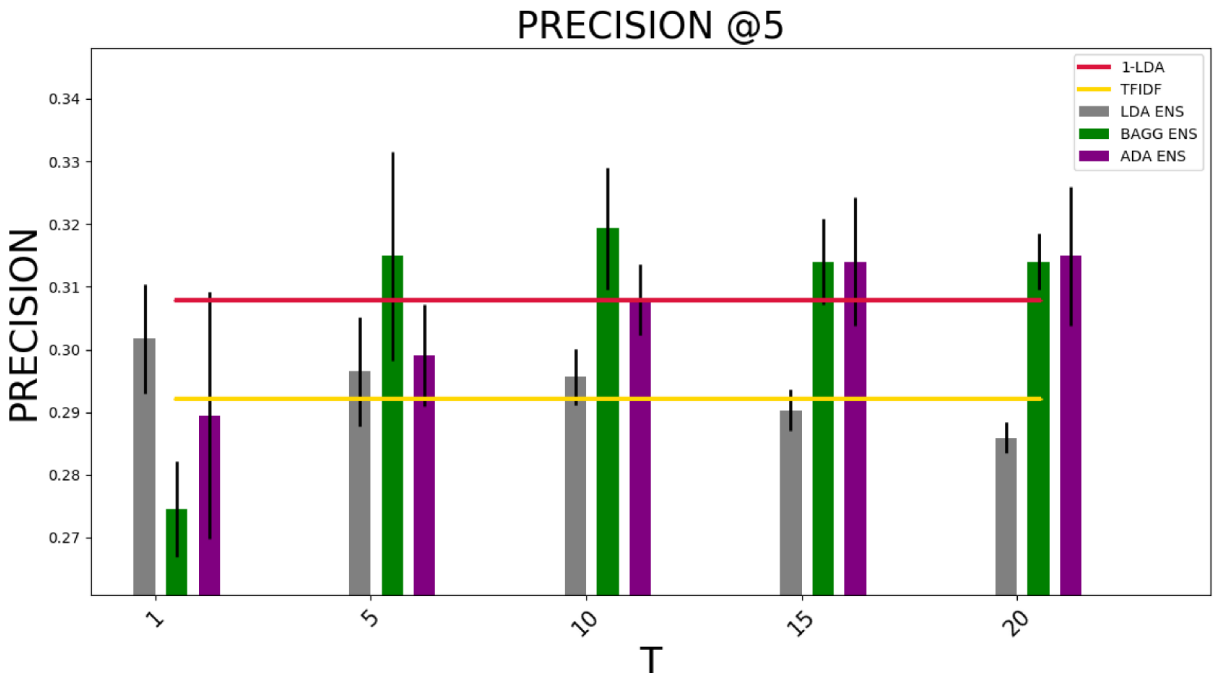


FIGURA 6.3. CISI PRE 5

En relación a **Recall**, vemos que CACM obtiene mejores resultados para @15 y @20, como puede verse en la Figura 6.4 . En el caso de MED, CISI y CRAN se obtienen buenos resultados para @5. Se pueden ver que en algunos casos se supera a TFIDF, pero no a LDA, como por ejemplo en la Figura 6.5. Excepcionalmente, se puede observar que en el caso de CISI se supera a LDA pero no a TFIDF. Estos resultados son importantes ya que nos indican que tan bueno es el algoritmo para recuperar documentos que son considerados relevantes, en relación con los demás.

En relación a los métodos de ensamblado podemos ver que para Recall, el método de muestreo aleatorio uniforme supera a los demás en la mayoría de las ocasiones, a excepción de LDA-Adaboost, en dataset CISI, que supera a los demás métodos en @5.

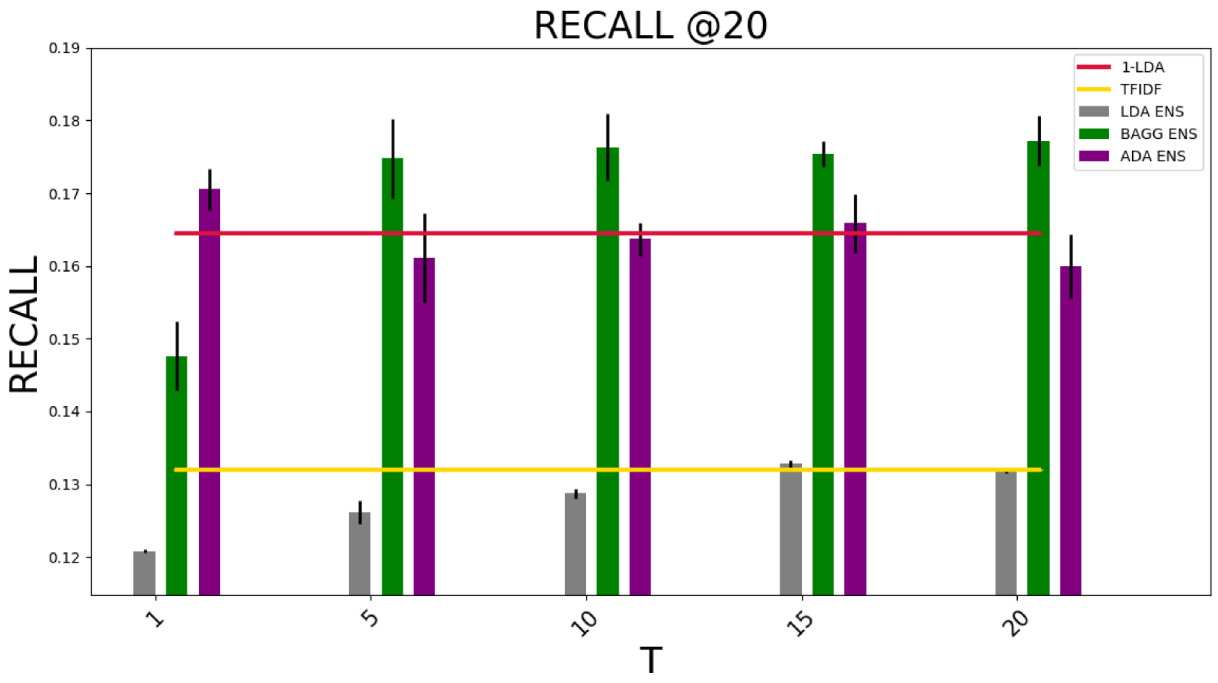


FIGURA 6.4. CACM REC 20

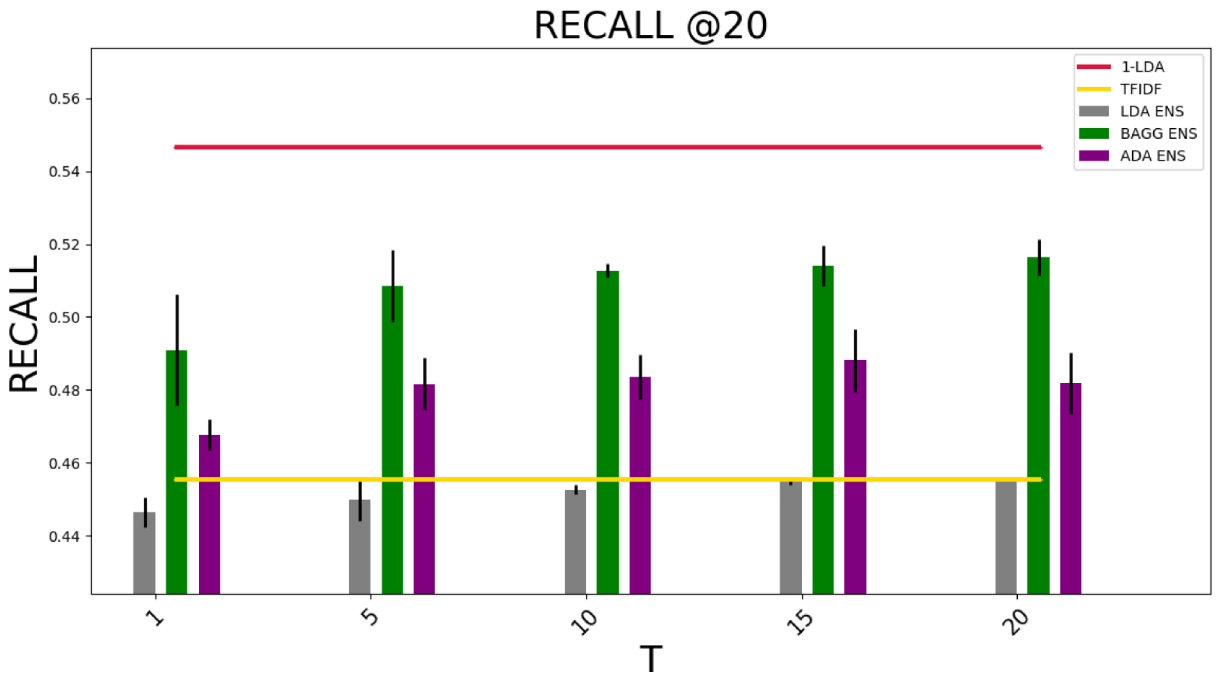


FIGURA 6.5. MED REC 20

En relación a **F1** se observa que para listas de ranking pequeñas se obtienen buenos resultados, ya que se supera a los métodos base en todos los datasets, como puede verse por ejemplo en @5 para CRAN en la Figura 6.6. En otras, para listas más grandes ni siquiera se supera el modelo base LDA como en la Figura 6.7. La media armónica entre precisión y recall obtiene buenos resultados en ese sentido. En relación a los métodos de ensamblado, tanto LDA-Bagging como LDA-Adaboost tienen buenos resultados para listas pequeñas de ranking.

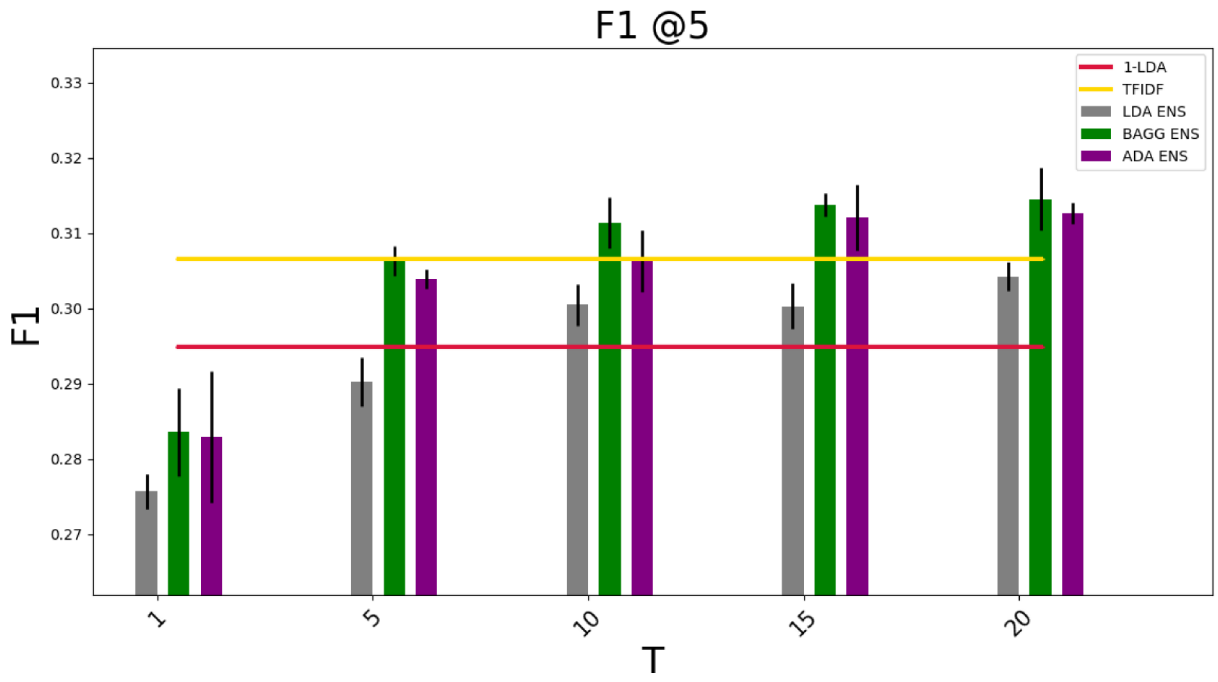


FIGURA 6.6. CRAN F1 5

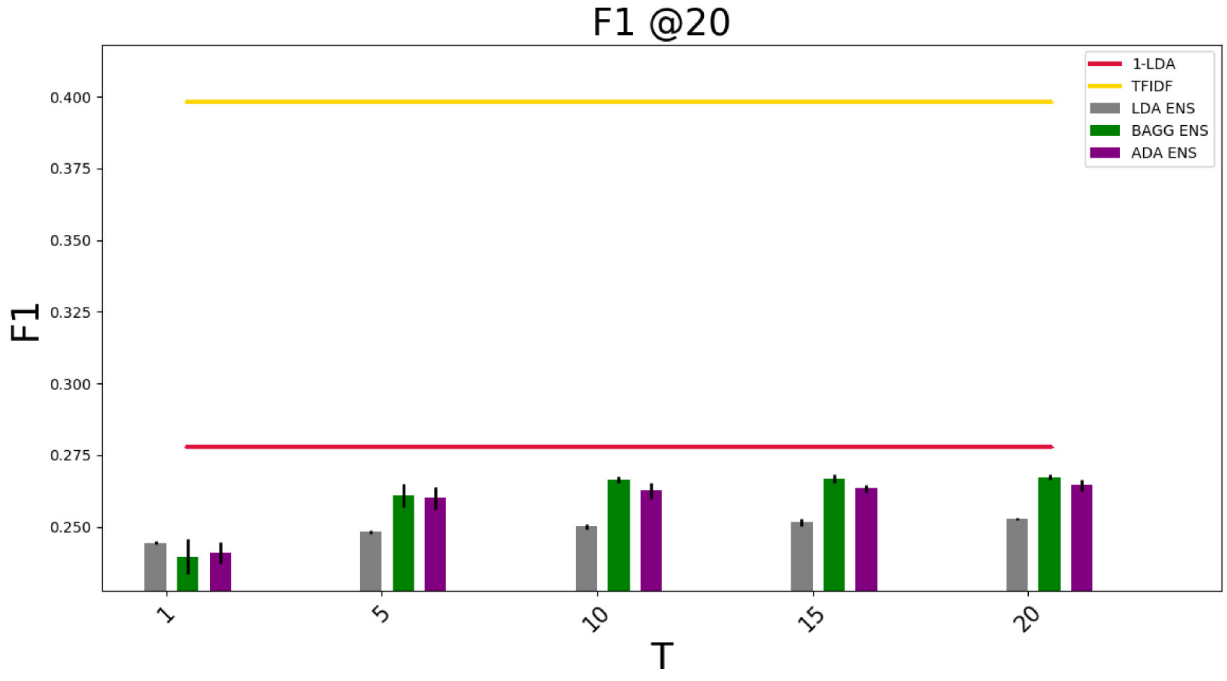


FIGURA 6.7. CRAN F1 20

En sentido general, se puede apreciar que los supuestos distribucionales se cumplen para lista de 5 o 10 valores. En estos casos podemos ver para Bagging, que en general obtiene muy buenos resultados. Esto puede significar que es una buena idea suponer que las muestras y las consultas provienen de la misma distribución de probabilidad. Por otro lado, y de manera excepcional, podemos ver que para el dataset CISI@5, Adaboost obtiene buenos resultados, tanto en Recall como en F1. En este caso el supuesto de que se puede dar más ponderación a los documentos que no fueron tan bien modelados, en este caso las consultas podrían ser modeladas por estas distribuciones de probabilidad. En general, la propuesta de ensamblado de

conjuntos disjuntos no tiene buenos resultados, a excepción de CISI donde se ve en Recall@10 o en MAP@20, que obtiene el mejor resultados de todos. Podemos decir que dividir el corpus en conjuntos para poder arma un ensamblado no ofrece buenos resultados a la hora de hacer recuperación.

Finalmente, en los anexos también se muestran las curvas de precision y recall de los métodos testeados para los 4 datasets y los métodos utilizados. Al comparar el método base con los métodos de ensamblado vemos que en general los ensamblados tienen mejor desempeño. Salvo algunas excepciones, como por ejemplo CISI@20 como vemos en la Figura 6.8, donde algunos tramos gana el método base, en MED@15, como se ve en la Figura 6.9.

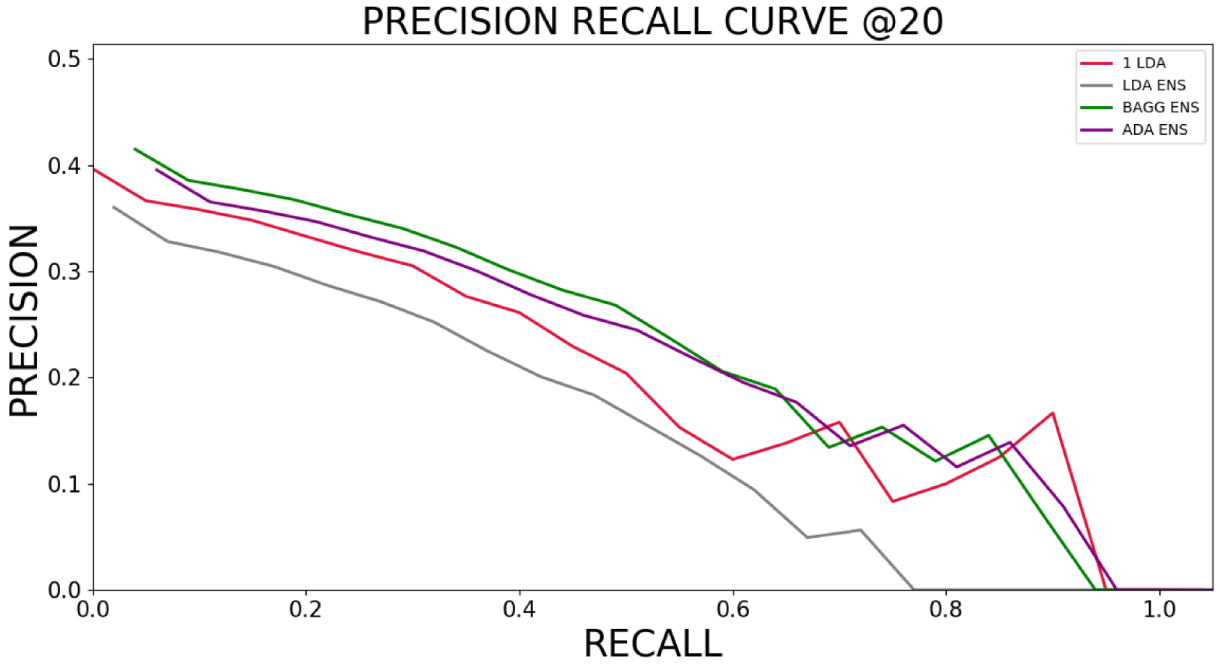


FIGURA 6.8. CISI CURVA 20

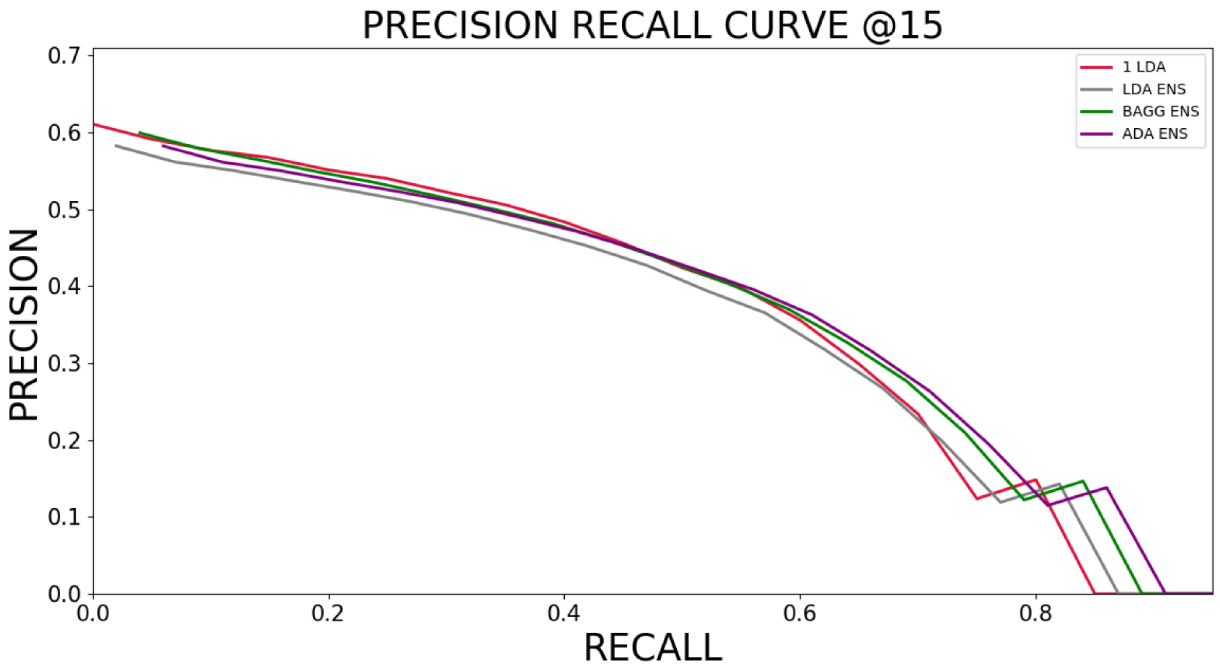


FIGURA 6.9. MED CURVA 15

En los experimentos como CACM@15 o CISI@10, método de muestreo uniforme muestra una precisión más alta para un recall más alto. Esto significa que los documentos en altas posiciones son realmente relevantes para el usuario. Por el contrario, en MED@5, claramente se ve una superioridad del método de muestreo adaptativo. Podemos ver esta diferencia en la Figura 6.10 y en la Figura 6.11.

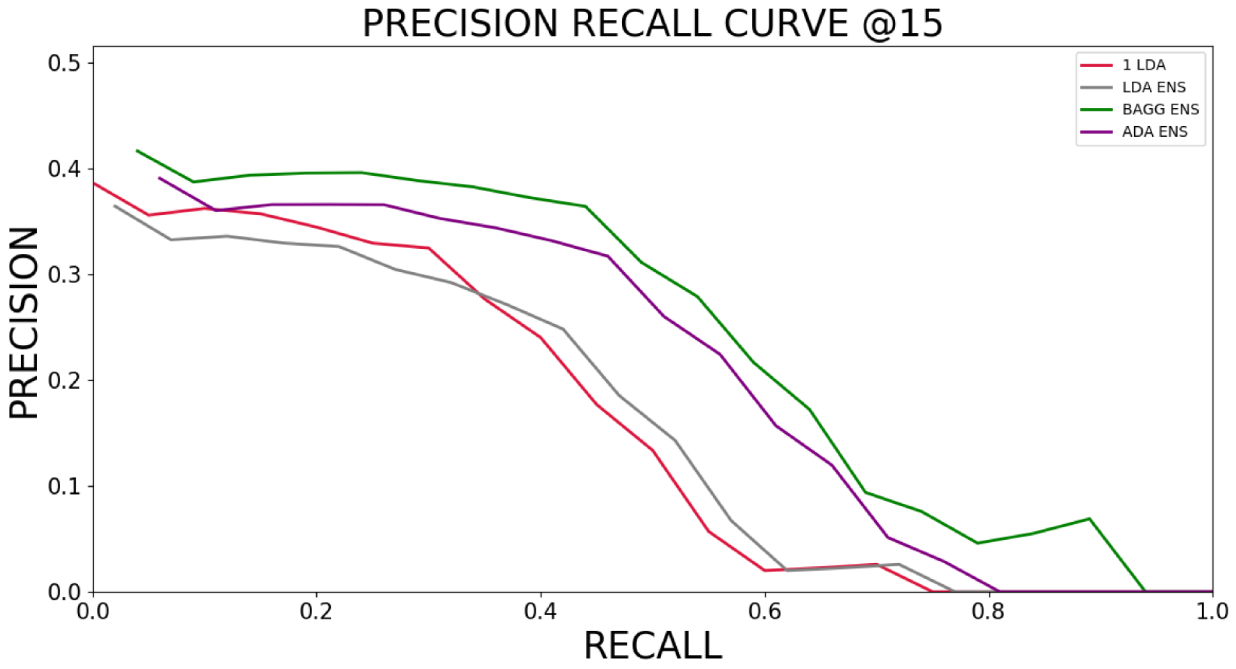


FIGURA 6.10. CACM CURVA 15

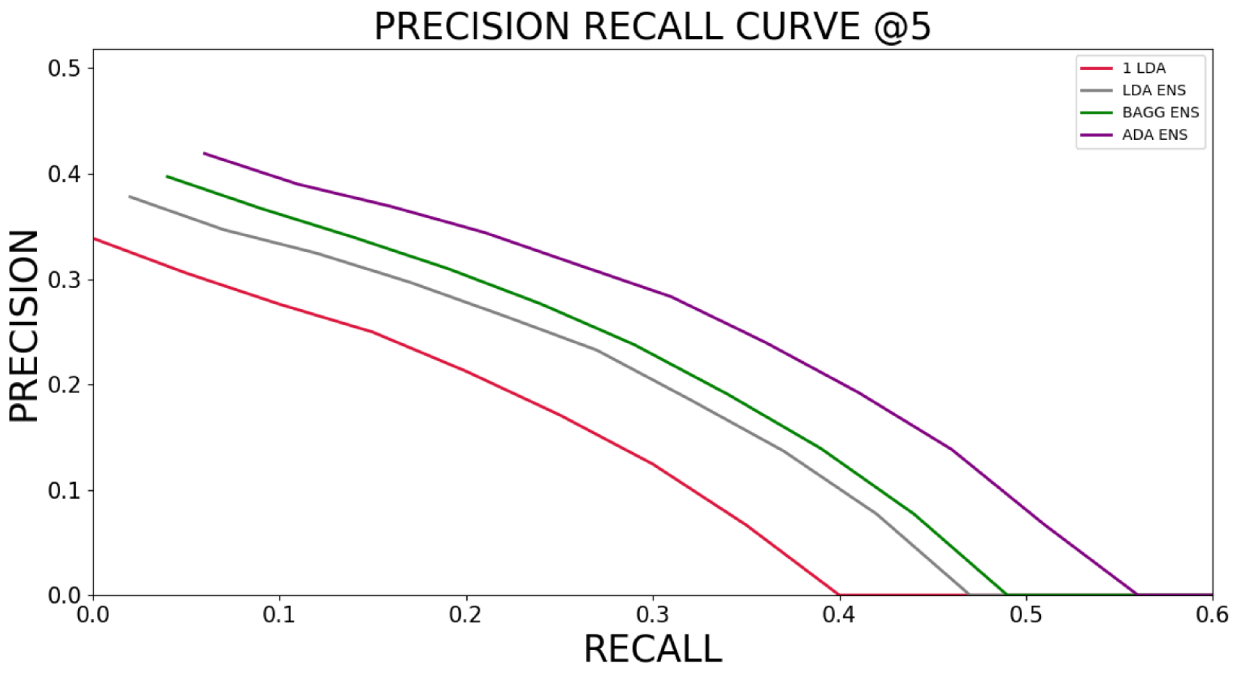


FIGURA 6.11. MED CURVA 5

Finalmente en algunos casos, por ejemplo en CACM@10, si bien los ensamblados tienen desempeño superior, es dificil hacer la diferencia, como puede verse en la Figura 6.11.

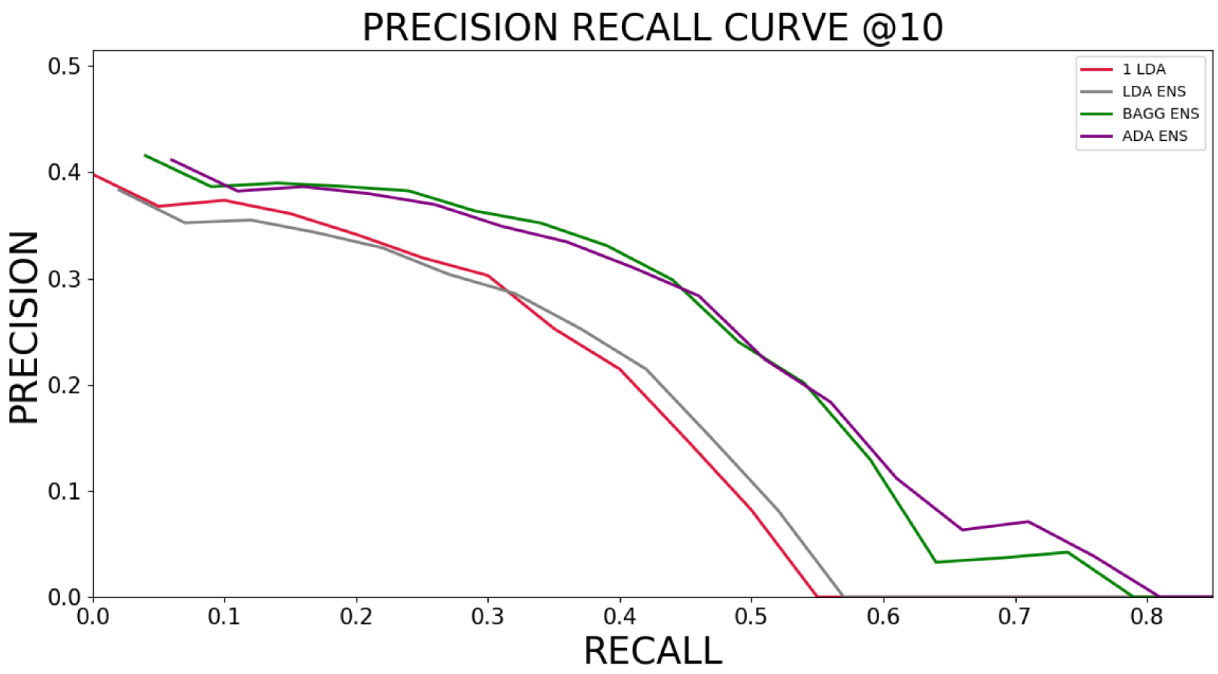


FIGURA 6.12. CACM CURVA 1O

Se compara cuantitativa y cualitativamente el desempeño de los tres métodos propuestos, además de los métodos base propuestos, **LDA** para recuperación, introducido por [**WC06a**], y **TF-IDF** [**SB88**]. Adicionalmente, se incluyen dos evaluaciones de métodos de recuperación de información basado en modelos, **DBNIRM** (Dependency Bayesian Network-based Information Retrieval Model) [**GO17**], un modelo de RI basado en redes bayesianas que alcanza un buen desempeño de recuperación al detectar las dependencias más importantes entre los términos en una red bayesiana basado en términos. El identificar pares de términos relacionados es útil en RI, al determinar relaciones semánticas entre documentos y términos de las consultas. También se incluye un segundo método de recuperación de información basado en modelos llamado **CCLR** (Concept Coupling Learning Retrieval) [**HSNC18**], que usa las estructuras conceptuales para modelar las relaciones de dependencia entre los términos del documento.

Por otro lado, se mide el impacto del número de modelos en términos de las cuatro medidas de desempeño, encontrando que ellas muestran resultados consistentes. En el Cuadro 2 se pueden ver los resultados en términos de MAP para las listas top-10.

# models		1	5	10	15	20
MED	**LDA Ens**	0.722	**0.771**	0.756	0.757	0.756
	BAGG Ens	0.802	0.797	0.805	**0.806**	0.805
	ADA Ens	**0.799**	0.778	0.773	0.771	0.766
CRAN	**LDA Ens**	0.518	**0.582**	0.576	0.575	0.576
	BAGG Ens	0.554	0.576	0.577	**0.578**	0.576
	ADA Ens	0.565	**0.576**	0.575	0.573	0.573
CISI	**LDA Ens**	0.398	**0.442**	0.392	0.391	0.393
	BAGG Ens	0.412	0.437	0.432	0.432	**0.438**
	ADA Ens	0.428	0.441	**0.443**	0.441	0.437
CACM	**LDA Ens**	0.138	0.157	**0.162**	0.161	0.160
	BAGG Ens	**0.192**	0.186	0.185	0.183	0.181
	ADA Ens	**0.188**	0.167	0.166	0.165	0.162

CUADRO 2. Efecto del número de modelos en cada estrategia de ensamblado. Los resultados se reportan usando MAP@10. Los resultados reportados consideran el promedio entre 5 intentos. Las fuentes en negrita indican las mejores configuraciones. Diferencias entre los modelos son estadisticamente significativas con un 95 % de confianza de acuerdo al test Wilcoxon.

El Cuadro 2 muestra la falta de un claro patrón de dependencia entre el número de modelos requeridos para obtener la mejor configuración y el modelo de ensamblado. Para **LDA Ens**, el mejor resultado en MED, CRAN y CISI se obtienen usando cinco modelos. Sin embargo, en CACM, **LDA Ens** requiere diez modelos. Para **BAGG Ens** alcanza su mejor resultado en MED y CRAN usando 15 models. En CISI, el mejor resultado se alcanza usando 20 modelos, pero en CACM solo 1. Finalmente, **ADA Ens** se obtiene el mejor resultado en MED y CACM usando solo un modelo, mientras que en CRAN, se requieren 5 y en CISI se requieren 10.

En la mayoría de los casos, el desempeño mejora usando más modelos. En el caso de **LDA Ens**, los mejores resultados son siempre obtenidos por cinco o más modelos. Cuando usamos **BAGG Ens**, tanto MED, como CRAN y CISI requieen al menos 15 modelos. Independiente del dataset, el más dificil es CACM, para el cual todas las estrategias consistentemente obtienen los más bajos resultados. En este dataset, **BAGG Ens** y **ADA Ens** muestran que el aprendizaje con ensamblado no alcanza mejoras en su desempeño.

		LDA	TF-IDF	DBNIRM	CCLR	LDA Ens	BAGG Ens	ADA Ens
MED	MAP	**0.869**	0.789	0.758	0.714	0.789 ± 0.001	0.867 ± 0.012	0.809 ± 0.007
	P	0.706	0.706	0.712	0.684	0.706 ± 0.002	**0.751 ± 0.008**	0.715 ± 0.011
	R	0.171	0.175	0.178	0.162	0.175 ± 0.001	**0.186 ± 0.002**	0.178 ± 0.002
	F_1	0.276	0.281	0.284	0.262	0.281 ± 0.001	**0.298 ± 0.003**	0.285 ± 0.004
CRAN	MAP	0.604	0.621	0.605	0.587	0.621 ± 0.001	**0.629 ± 0.006**	0.618 ± 0.005
	P	0.344	0.352	0.358	0.342	0.351 ± 0.002	**0.363 ± 0.004**	0.361 ± 0.001
	R	0.257	0.269	0.264	0.245	0.268 ± 0.001	**0.277 ± 0.003**	0.275 ± 0.001
	F_1	0.294	0.305	0.303	0.285	0.304 ± 0.001	**0.314 ± 0.004**	0.312 ± 0.001
CISI	MAP	0.464	0.468	0.460	0.443	**0.472 ± 0.001**	0.472 ± 0.001	0.464 ± 0.008
	P	0.307	0.292	0.298	0.286	0.285 ± 0.002	0.314 ± 0.004	**0.314 ± 0.011**
	R	0.059	0.056	0.061	0.052	0.053 ± 0.001	0.061 ± 0.002	**0.063 ± 0.002**
	F_1	0.099	0.093	0.101	0.088	0.091 ± 0.001	0.101 ± 0.003	**0.105 ± 0.003**
CACM	MAP	0.158	0.146	0.148	0.135	0.146 ± 0.001	**0.161 ± 0.004**	0.149 ± 0.004
	P	0.107	0.103	0.106	0.112	0.103 ± 0.001	**0.115 ± 0.005**	0.106 ± 0.001
	R	0.039	0.038	0.041	0.042	0.038 ± 0.001	**0.047 ± 0.004**	0.039 ± 0.001
	F_1	0.057	0.055	0.059	0.061	0.056 ± 0.002	**0.067 ± 0.005**	0.057 ± 0.001

CUADRO 3. Resultados reportados usando listas @5.

		LDA	TF-IDF	DBNIRM	CCLR	LDA Ens	BAGG Ens	ADA Ens
MED	MAP	0.802	0.756	0.780	0.689	0.771 ± 0.001	**0.806 ± 0.006**	0.799 ± 0.001
	P	**0.680**	0.611	0.625	0.606	0.607 ± 0.003	0.658 ± 0.008	0.636 ± 0.011
	R	**0.324**	0.291	0.308	0.288	0.291 ± 0.002	0.315 ± 0.005	0.307 ± 0.006
	F_1	**0.439**	0.394	0.412	0.391	0.392 ± 0.002	0.427 ± 0.006	0.414 ± 0.008
CRAN	MAP	0.568	0.572	0.573	0.447	**0.582 ± 0.001**	0.578 ± 0.007	0.576 ± 0.006
	P	0.265	0.261	0.264	0.249	0.259 ± 0.001	**0.271 ± 0.001**	0.269 ± 0.003
	R	0.386	0.384	0.381	0.346	0.384 ± 0.001	**0.394 ± 0.001**	0.391 ± 0.005
	F_1	0.315	0.311	0.311	0.289	0.309 ± 0.001	**0.321 ± 0.001**	0.319 ± 0.004
CISI	MAP	0.426	0.431	0.438	0.396	**0.442 ± 0.003**	0.438 ± 0.004	0.443 ± 0.008
	P	**0.275**	0.263	0.274	0.268	0.258 ± 0.004	0.271 ± 0.002	0.266 ± 0.003
	R	0.095	**0.111**	0.107	0.108	0.107 ± 0.003	0.101 ± 0.006	0.097 ± 0.002
	F_1	0.142	**0.156**	0.153	0.154	0.151 ± 0.003	0.146 ± 0.007	0.142 ± 0.002
CACM	MAP	**0.191**	0.161	0.184	0.165	0.162 ± 0.001	0.192 ± 0.004	0.188 ± 0.001
	P	**0.121**	0.088	0.116	0.084	0.088 ± 0.001	0.112 ± 0.003	0.101 ± 0.002
	R	**0.116**	0.078	0.099	0.101	0.078 ± 0.002	0.102 ± 0.003	0.098 ± 0.003
	F_1	**0.118**	0.082	0.106	0.092	0.082 ± 0.002	0.107 ± 0.003	0.101 ± 0.001

CUADRO 4. Resultados reportados usando listas @10.

		LDA	TF-IDF	DBNIRM	CCLR	LDA Ens	BAGG Ens	ADA Ens
MED	MAP	**0.759**	0.711	0.736	0.712	0.711 ± 0.001	0.738 ± 0.011	0.713 ± 0.001
	P	**0.596**	0.497	0.562	0.573	0.497 ± 0.002	0.558 ± 0.007	0.527 ± 0.008
	R	**0.546**	0.455	0.514	0.489	0.455 ± 0.001	0.516 ± 0.005	0.481 ± 0.008
	F_1	**0.571**	0.475	0.536	0.527	0.475 ± 0.002	0.536 ± 0.006	0.503 ± 0.008
CRAN	MAP	0.509	**0.525**	0.517	0.496	0.525 ± 0.001	0.522 ± 0.008	0.516 ± 0.002
	P	0.188	0.172	**0.198**	0.164	0.171 ± 0.001	0.181 ± 0.001	0.179 ± 0.001
	R	**0.526**	0.484	0.499	0.414	0.483 ± 0.001	0.506 ± 0.001	0.504 ± 0.003
	F_1	**0.278**	0.253	0.283	0.235	0.252 ± 0.001	0.267 ± 0.001	0.264 ± 0.001
CISI	MAP	0.385	0.396	0.391	0.351	**0.397 ± 0.003**	0.395 ± 0.011	0.386 ± 0.002
	P	**0.245**	0.221	0.237	0.208	0.214 ± 0.002	0.232 ± 0.001	0.228 ± 0.001
	R	**0.176**	0.163	0.168	0.152	0.156 ± 0.001	0.166 ± 0.003	0.161 ± 0.003
	F_1	**0.205**	0.187	0.196	0.175	0.181 ± 0.002	0.193 ± 0.002	0.189 ± 0.001
CACM	MAP	**0.188**	0.169	0.181	0.159	0.169 ± 0.001	0.184 ± 0.005	0.171 ± 0.001
	P	0.098	0.079	0.092	0.076	0.079 ± 0.001	**0.101 ± 0.004**	0.093 ± 0.001
	R	0.164	0.132	0.154	0.125	0.131 ± 0.001	**0.177 ± 0.003**	0.159 ± 0.004
	F_1	0.122	0.098	0.115	0.094	0.098 ± 0.001	**0.128 ± 0.004**	0.118 ± 0.002

CUADRO 5. Resultados reportados usando listas @20.

Para comparar los resultados de estas estrategias con los métodos base, se usan las mejores configuraciones en términos del número de modelos indicados en la Tabla 2. Los resultados en términos de MAP, Precisión, Recall y F1 se muestran para listas @5, @10 y @20 en los cuadros 3, 4, y 5, respectivamente.

La diferencia entre los modelos y los métodos base son significativos estadísticamente con un 95 % de confianza de acuerdo al test Wilcoxon. Los resultados en los cuadros 3, 4, y 5 muestran que **LDA** es competitivo, superando a **TF-IDF** en MED y CACM en todas las comparaciones. Sin embargo, el resultado de **LDA** en CRAN y CISI muestran un deterioro comparado con los que se obtienen con **TF-IDF** en todos los datasets. Este resultado indica que identificar dependencias entre pares de términos es relevante para mejorar la descripción de documentos y mejorar los calces de los términos de la consulta. Específicamente, **DBNIRM** obtiene un mejor resultado competitivo en CRAN y CISI, especialmente en listas @10 y @20, donde se las arregla para superar **LDA** y **TF-IDF** en MAP y Presicion pero obtiene bajos resultados en Recall. Por otro lado, **CCLR** muestra resultados consistentemente más bajos que **DBNIRM**, mostrando sus mejores resultados en MED y CACM para listas @20. Al extender **LDA** con aprendizaje de ensamblados, algunos resultados muestran mejoras significativas en muchos casos. Por ejemplo, **BAGG Ens** supera en MED, CRAN, y CACM a todos sus competidores por un márgen sustancial en listas @5. La diferencia entre **BAGG Ens** y **LDA** se acorta en los resultados de @10 y @20. **BAGG Ens** supera a sus competidores en MED y CRAN en resultados @10. En resultados @20, **LDA** es el método más robusto, siendo superado por **BAGG Ens** en CACM. **LDA Ens** es un método competitivo también, obteniendo buen desempeño para resultados @10, alcanzndo los mejores resultados en MAP para CRAN y CISI. **LDA Ens** mantiene sus buenos resultados en CISI para las listas @20, obteniendo el mejor desempeño en MAP. En este sentido, esta estrategia supera a sus competidores sólo en resultados @5 en CISI. En el resto de las comparaciones, **ADA Ens** falla derrotar a sus otros competidores.

El hecho de que **ADA Ens** falla en superar a sus competidores indica que el muestreo adaptativo es poco efectivo cuando se trabaja en conjunto con modelos de temas. Por otro lado, el dominio de particionamiento basado en particiones conjuntas (**LDA Ens**) o remuestreo boostrap (**BAGG Ens**) muestran ser mucho más efectivos. Estos descubrimientos se relacionan a las potencialidades y limitaciones de los modelos de temas usados para generar ensamblados, los cuales fallan en identificar temas con más valor para documentos complejos. Por el contrario, **LDA** saca más ventaja de las estrategias de remuestreo no-adaptativo. El remuestreo permite descartar documentos en particiones específicas, introduciendo una gran variedad en las muestras.

6.2. Discusión

Un interesante resultado se muestra en los cuadros 3, 4 y 5 y tiene relación a la efectividad de las técnicas de ensamblado en términos del largo de las listas de resultados. Mientras que

los resultados del aprendizaje de ensamblado son mejores para listas más cortas (@5), estos se deterioran para listas más largas. De hecho, en listas @20, **LDA** supera a los métodos de ensamblado en MED, CRAN y CISI mientras que **BAGG Ens** sólo mantiene su desempeño en CACM. Este descubrimiento indica que las técnicas de aprendizaje con ensamblado permiten identificar resultados más relevantes sólo en las primeras posiciones de la lista, sugiriendo que las listas de palabras descriptiva de los temas puede diferir. Este hecho explicaría la diferencia entre las estrategias de ensamblado.

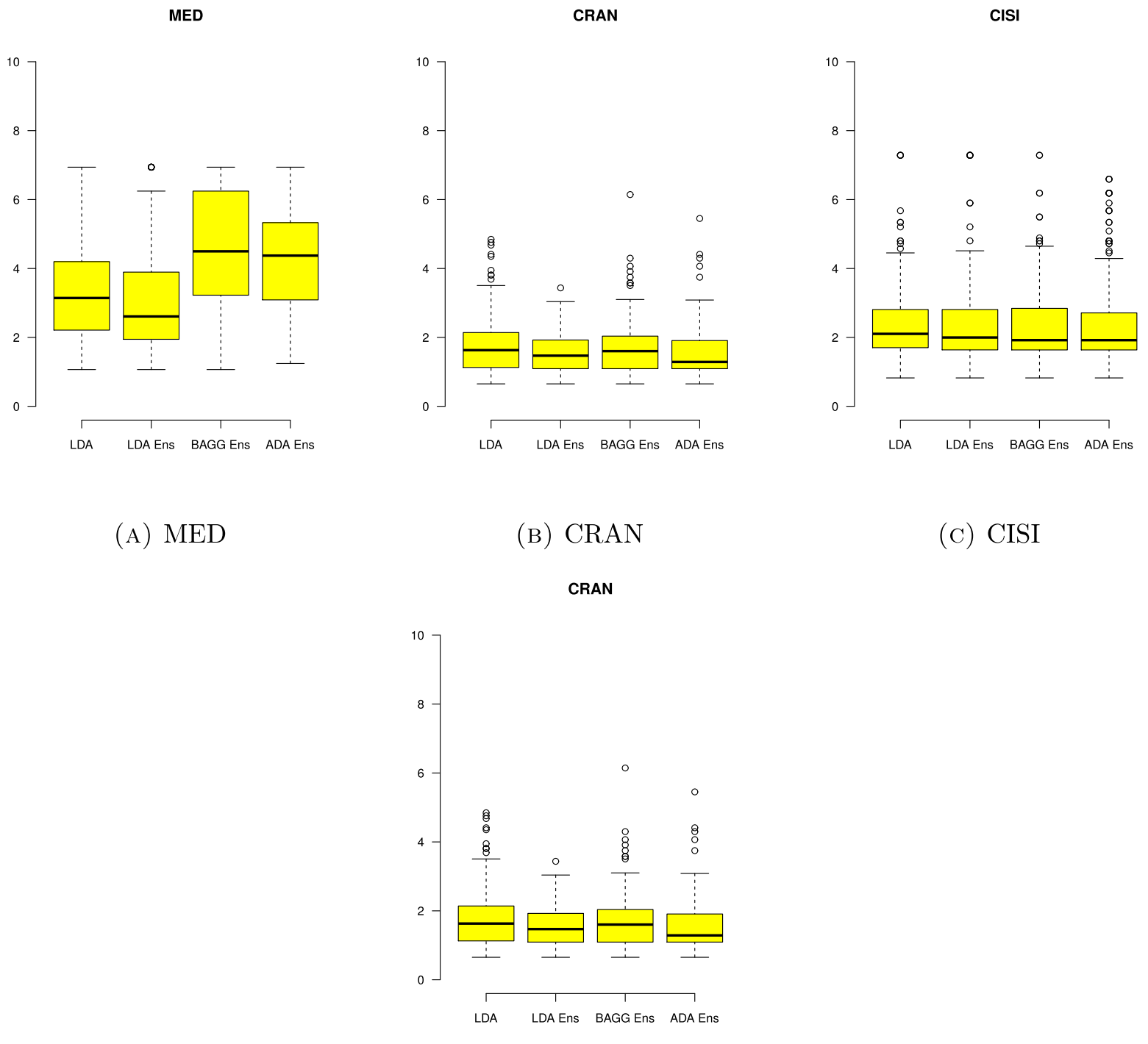
Para ilustrar las diferencias entre los cuatro métodos basados en modelos de temas, comparamos las top-5 palabras con la coherencia de temas más alta detectada por LDA para cada conjunto de datos. Estos temas fueron encontrados en los otros métodos (**LDA Ens**, **BAGG Ens** y **ADA Ens**), e identificando las diferencias entre estas listas de palabras. Para cada estrategia de temas, seleccionamos el modelo más cercano al desempeño promedio mostrado en los cuadros 3, 4, y 5, haciendo la comparación más consistente y justa. El resultado de este análisis comparativo se muestra en el Cuadro 6.

	TID	**LDA [WC06a]**	**LDA Ens**	**BAGG Ens**	**ADA Ens**
MED	1	alveolar, line, lung pulmonary, surface	acid, alveolar, lung perform, rate	alveolar, line, lung mouse, pulmonary	alveolar, information, line, lung, lymphatic
	2	female, male, rat, testosterone, tissue	demonstrate, female, intact, show, testosterone	conjugate, female, normal, plasma, testosterone	female, normal, patient, plasma, testosterone
	3	body, cool, hypothermia, perfusion, temperature	heart, hypothermia, patient perfusion, surgery	body, cool, hypothermia, perfusion, temperature	body, coronary, hypothermia, perfusion, temperature
	4	blood, brain, control, lactate, response	blood, brain, group, study, surface	blood, brain, increase, lactate, rise	blood, brain, hypoxia, lactate, rise
	5	cancer, carcinoma, case, lung, primary	cancer, carcinoma, decrease, enzyme, pulmonary	cancer, carcinoma, case, lung, tumor	cancer, carcinoma, cell, lung, radiation
CRAN	1	equation, method, numerical, problem, solution	base, equation, method, problem, solution	equation, method, problem, solution, solve	boundary, method, problem, solution, solve
	2	body, flow, hypersonic, nose, pressure	flow, hypersonic, show theory, velocity	body, flow, hypersonic, pressure, shock	flow, hypersonic, inviscid, pressure, shock
	3	buckling, cylinder, pressure, shell, theory	buckling, cylinder, shell, wall, wave	buckling, creep, cylinder, initial, shape	buckling, creep, cylinder, equation, flow
	4	airplane, altitude, boom, flight, shock	airplane, altitude, boom, flight, shock	airplane, altitude, boom, flight, mach	airplane, altitude, flight, mach, number
	5	dimensional, disturbance, flow, small, solution	aircraft, disturbance, flight, ground, level	amplitude, dimensional, disturbance, energy, wave	cone, dimensional, disturbance, surface, wave
CISI	1	book, collection, librarian, library, university	base, book, collection, concept, subject	book, circulation, collection, library, medical	book, circulation, collection, fact, size
	2	information, provide, reference, service, university	entry, information, provide, search, user	information, organization, provide, service, type	citation, information, literature, provide, reference
	3	health, library, manpower, professional, science	center, health, international, library, national	health, hospital, library, manpower, science	health, library, manpower, program, scale
	4	comparative, economic, problem, project, scientist	addition, economic, experimental, system, theoretical	country, economic, interest, problem, view	economic, international, project, series, time
	5	change, data, model, rate, storage	data, entry, large, research, storage	base, data, information, large, model	data, idea, library, memory, model
CACM	1	correctness, program, proof, prove, technique	algorithm, make, program, proof, similar	correctness, program, proof, prove, technique	correctness, program, proof, prove, specification
	2	algorithm, class, function, processor, schedule	algorithm, class, identify, improve, reduce	algorithm, class, equation, problem, solution	algorithm class, drum, schedule, time
	3	fortran, input, language, output, program	computer, input, processing, program, provide	input, machine, output, program, user	data, information, input, processing, program
	4	debug, design, feature, program, system	applicable, debug, program, solve, user	debug, input, operating, process, program	communication, debug, illustrate, program, user
	5	hash, method, search, table, technique	algorithm, efficiency, hash, length, table	hash, method, quadratic, size, table	hash, language, search, structure, table

CUADRO 6. Top-5 palabras por tema para las estrategias de ensamblado propuestas.

En el Cuadro 6, se destacan algunas palabras que complementan la lista de palabras detectadas por **LDA**. Primero, para cada tema, se calcula el valor IDF de las top-5 palabras en **LDA**. Entonces, las nuevas palabras identificadas por **LDA Ens**, **BAGG Ens**, o **ADA Ens** que estén por encima del IDF máximo o por debajo del IDF mínimo se consideran como palabras con un significado más específico o más general, respectivamente. Las palabras más genéricas se indican en rojo, mientras que las más específicas se muestran en azul.

En el Cuadro 6 se muestra que las tres estrategias de ensamblado pueden identificar nuevas palabras relativas a los temas detectado por **LDA**. Mientras que la mayoría de las palabras son genéricas, algunas palabras específicas complementan la descripción original del tema. Todas las palabras agregadas por estas estrategias tienen una relación semántica con el tema original, a excepción de *drum* (el que se indica en verde), el cual no tiene ninguna conexión semántica aparente con el tema 2 en CACM. Todos, **LDA Ens**, **BAGG Ens** y **ADA Ens** parecen detectar palabras específicas dependiendo del tema. Este descubrimiento es interesante ya que muestra que el tema detectado puedes ser más o menos específico dependiendo de la estrategia de ensamblado. Se notan algunas diferencias entre las estrategias. **LDA Ens** trabaja con particiones diferentes del corpus. Esta estrategia de partición permite detectar palabras más genéricas. En el caso de **BAGG Ens** y **ADA Ens**, ya que estas estrategias se especializan en documentos más complejos del modelo, tienden a detectar palabras más específicas. Se muestra en la Figura 6.13 el factor de distribución IDF de las estrategias en cada dataset estudiado en este trabajo para corroborar esta intuición.



(A) MED (B) CRAN (C) CISI

(D) CACM

FIGURA 6.13. Distribuciones de IDF para cada método LDA para todos los conjuntos de datos usados.

Para crear los boxplots de la Figura 6.13, se seleccionan los top-20 temas más altamente correlacionados de cada estrategia en cada dataset. Entonces, se seleccionan sus top-10 términos más descriptivos para cada tema, calculando sus valores IDF en el conjunto de datos. Los boxplot de la Figura 6.13 muestran algunos resultados interesantes. La distribución IDF en MED es más dispersa, siendo **BAGG Ens** y **ADA Ens**, las estrategias que se las arreglan en identificar palabras más específicas. Este resultado coincide con el desempeño obtenido por estas estrategias, que son las que mejores resultados obtienen en este estudio. Por otro lado, en CRAN y CACM, **ADA Ens** no puede identificar palabras específicas, obteniendo la

mediana IDF más baja de las cuatro estrategias. Es estos datasets, **LDA** y **BAGG Ens** se desempeñan un poco mejor que las otras estrategias en mediana IDF. Finalmente, en CISI, ninguna de las estrategias puede identificar palabras más específicas que el resto. Este resultado coincide con el hecho de que los desempeños de las cuatro estrategias indicadas en los cuadros 3, 4, y 5 son similares. En resumen, la Figura 6.13 muestra que la habilidad de cada estrategia de identificar palabras específicas en cada tema varía de acuerdo a los conjuntos de datos. Mientras **BAGG Ens** y **LDA** identifica palabras específicas, las otras estrategias no parecen tener una habilidad significativa para detectar palabras específicas en cada tema.

Por otro lado, se estudiaron la naturaleza de las consultas en la cual los métodos propuestos se desempeñan mejor que sus competidores. Primero, se determina el conjunto de consultas donde cada método basado en LDA vence a sus contrincantes por al menos el 10 % MAP@5, por lo que la ventaja obtenida por el método es significativa. El desempeño promedio del modelo indicado en el Cuadro 3 se usa para conducir este análisis, favoreciendo una comparación justa entre las diferentes estrategias consideradas en este trabajo. Las consultas, donde ninguno de los métodos se las arregla para ganar un margen significativo, fueron excluidas de este análisis. Se muestra en el Cuadro 7 la lista de consultas de cada conjunto de datos donde el ganador claro fue el método observado en MAP@5. Se muestra el id de la consulta, las palabras de la consulta y el nombre del método ganador.

El resultado del Cuadro 7 muestra que **LDA** y **BAGG Ens** son métodos que, al superar a sus competidores, alcanzan más ventajas en términos de MAP@5. Mientras, en MED y CISI, **BAGG Ens** se las arregla para superar a sus competidores en más consultas que el resto de los métodos, en CRAN y CACM, ambos **BAGG Ens** y **LDA** son muy competitivos. En ninguna de estas consultas, **LDA Ens** se las arregla para superar a sus competidores en MAP@5, mostrando que este método, aunque obtiene un interesante resultado promedio, no se las arregla para superar al resto de manera consistente. Por otro lado, **ADA Ens** sólo se las arregla para superar a su competidos en algunas consultas de CRAN. Sin duda, ambos **LDA** y **BAGG Ens** se las arreglan para ganar en consultas largas y cortas, no observando claramente un patrón que muestra las dependencias entre los tipos de estrategia de ensamblado y un largo de la consulta.

Los resultados muestran otro descubrimiento importante. Mientras CRAN tiene el doble de consultas que CISI, el número de consultas en la cual el método de ensamblado supera a sus competidores muestra una proporción de 4 a 1. Este cociente puede ser puede ser atribuido al hecho de que el vocabulario de CRAN es más pequeño que CISI, lo que la hace fácil de modelar. El resultado de los cuadros 3, 4, y 5 muestra que los datasets en los cuales los métodos obtienen mejores resultados son MED y CRAN, en la cual los datasets que tienen mejores resultados son MED y CRAN, los cuales tienen pequeños vocabularios.

	QID	Query words	L	Winning
MED	3	['electron', 'microscopy', 'lung']	3	BAGG Ens
	12	['effect', 'azathioprine', 'systemic', 'lupus', 'erythematosus', 'regard', 'renal', 'lesion']	8	BAGG Ens
	16	['separation', 'anxiety', 'infancy', 'year', 'preschool', 'child', 'separation', 'child', 'mother']	9	BAGG Ens
	17	['nickel', 'nutrition', 'requirement', 'method', 'analysis', 'relation', 'enzyme', 'system', 'toxicity', 'human', 'laboratory', 'animal', 'deficiency', 'sign', 'symptom', 'level', 'foodstuff', 'level', 'blood', 'tissue']	20	LDA
	21	['language', 'development', 'infancy', 'pre', 'school']	5	LDA
	22	['mycoplasma', 'infection', 'presence', 'embryo', 'fetus', 'newborn', 'infant', 'animal', 'pregnancy', 'gynecologic', 'disease', 'related', 'chromosome', 'chromosome', 'abnormality']	15	LDA
	24	['compensatory', 'renal', 'hypertrophy', 'stimulus', 'result', 'mass', 'increase', 'hypertrophy', 'cell', 'proliferation', 'hyperplasia', 'remain', 'kidney', 'unilateral', 'nephrectomy', 'mammal']	16	BAGG Ens
	25	['chlorothiazide', 'diuril', 'hydrochlorothiazide', 'hydrodiuril', 'treatment', 'nephogenic', 'diabetes', 'insipidus', 'child', 'also', 'sodium', 'aldactone', 'spironolactone', 'treatment', 'childhood', 'nephogenic', 'diabetes', 'insipidus']	18	BAGG Ens
CRAN	5	['chemical', 'kinetic', 'applicable', 'hypersonic', 'aerodynamic', 'problem']	6	LDA
	17	['three', 'dimensional', 'problem', 'transverse', 'potential', 'flow', 'body', 'revolution', 'reduce', 'two', 'dimensional']	11	LDA
	32	['approximate', 'correction', 'thickness', 'slender', 'thin', 'wing', 'theory']	7	BAGG Ens
	33	['interference', 'free', 'longitudinal', 'stability', 'measurement', 'make', 'free', 'flight', 'model', 'compare', 'similar', 'measurement', 'low', 'blockage', 'wind', 'tunnel']	16	BAGG Ens
	37	['theoretical', 'method', 'predict', 'base', 'pressure']	5	BAGG Ens
	38	['transition', 'hypersonic', 'wake', 'depend', 'body', 'geometry', 'size']	7	LDA
	40	['transition', 'phenomenon', 'hypersonic', 'wake']	4	LDA
	43	['transonic', 'flow', 'arbitrary', 'smooth', 'airfoil', 'analyse', 'simple', 'approximate']	8	BAGG Ens
	47	['exist', 'solution', 'hypersonic', 'viscous', 'interaction', 'insulate', 'flat', 'plate']	8	BAGG Ens
	60	['simple', 'practical', 'method', 'numerical', 'integration', 'mix', 'problem', 'blasius', 'three', 'point', 'boundary', 'condition']	12	LDA
	73	['role', 'effect', 'chemical', 'reaction', 'particularly', 'equilibrium', 'play', 'similitude', 'law', 'govern', 'hypersonic', 'flow', 'slender', 'aerodynamic', 'body']	15	LDA
	77	['close', 'comparison', 'shock', 'layer', 'theory', 'exist', 'experiment', 'reynolds', 'number', 'merge', 'layer', 'regime']	12	BAGG Ens
	79	['aerodynamic', 'derivative', 'measure', 'hypersonic', 'mach', 'number', 'comparison', 'theoretical', 'work']	9	ADA Ens
	88	['satellite', 'orbit', 'contract', 'action', 'drag', 'atmosphere', 'scale', 'height', 'varies', 'altitude']	10	BAGG Ens
	91	['interference', 'effect', 'transonic', 'speed']	4	BAGG Ens
	95	['theoretical', 'heat', 'transfer', 'distribution', 'hemisphere']	5	BAGG Ens
	119	['effect', 'initial', 'axisymmetric', 'deviation', 'circularity', 'linear', 'large', 'deflection', 'load', 'deflection', 'response', 'cylinder', 'hydrostatic', 'pressure']	14	BAGG Ens
	120	['previous', 'analysis', 'circumferential', 'thermal', 'buckling', 'circular', 'cylindrical', 'shell', 'unnecessarily', 'involve', 'assume', 'form', 'mode']	13	LDA
	126	['thrust', 'vector', 'control', 'fluid', 'injection', 'dash', 'paper']	7	LDA
	165	['stable', 'profile', 'compressible', 'boundary', 'layer', 'induced', 'move', 'wave']	8	LDA
	172	['solution', 'blasius', 'problem', 'three', 'point', 'boundary', 'condition']	7	BAGG Ens
	184	['work', 'small', 'oscillation', 're', 'entry', 'motion']	6	LDA
	203	['simple', 'empirical', 'method', 'estimate', 'pressure', 'distribution', 'cone']	7	ADA Ens
	204	['viscous', 'effect', 'pressure', 'distribution']	4	BAGG Ens
	222	['investigate', 'shear', 'buckling', 'stiffen', 'plate']	5	LDA
	223	['paper', 'shear', 'buckling', 'unstiffened', 'rectangular', 'plate', 'shear']	7	BAGG Ens
CISI	13	['criterion', 'developed', 'objective', 'evaluation', 'information', 'retrieval', 'dissemination', 'system']	8	BAGG Ens
	19	['technique', 'machine', 'match', 'machine', 'search', 'system', 'cod', 'match', 'method']	9	BAGG Ens
	28	['computerize', 'information', 'system', 'field', 'related', 'chemistry']	6	ADA Ens
	34	['method', 'cod', 'computerize', 'index', 'system']	5	LDA
	44	['presently', 'fifty', 'technical', 'journal', 'publish', 'average', 'million', 'article', 'year', 'attempt', 'cope', 'scientific', 'publication', 'term', 'analysis', 'control', 'storage', 'retrieval']	18	BAGG Ens
	98	['online', 'retrieval', 'system', 'difficult', 'user', 'heterogeneity', 'complexity', 'investigation', 'concerned', 'concept', 'computer', 'interface', 'mean', 'simplify', 'access', 'operation', 'heterogeneous', 'bibliographic', ...]	33	BAGG Ens
CACM	7	['interested', 'distribute', 'concurrent', 'program', 'process', 'communicate', 'message', 'passing', 'area', 'include', 'fault', 'tolerance', 'technique', 'understand', 'correctness', 'algorithm', 'Fred', 'Schneider', 'dist']	19	LDA
	14	['optimal', 'implementation', 'sort', 'algorithm', 'database', 'management', 'application', 'Kenneth', 'Wilson', 'sort', 'physic', 'Newman', 'database']	13	BAGG Ens
	28	['information', 'packet', 'network', 'algorithm', 'rout', 'deal', 'topography', 'interested', 'hardware', 'Dean', 'jJgels', 'net']	12	BAGG Ens
	36	['fast', 'algorithm', 'context', 'free', 'language', 'recognition', 'parse', 'juris', 'hartmanis', 'fast', 'lang', 'recog', 'parse']	13	BAGG Ens
	58	['algorithm', 'statistical', 'package', 'anova', 'regression', 'square', 'generalize', 'linear', 'model', 'design', 'capability', 'formula', 'interest', 'student', 'test', 'Wilcoxon', 'sign', 'multivariate', 'component', 'include']	20	LDA

Cuadro 7. Consultas y métodos que obtienen los mejores resultados.

6.3. Limitaciones de este estudio

Debido al alto costo computacional involucrado en los experimentos, lo que implicaba llevar a cabo diversos intentos de cada modelo de temas, no fue fácil experimentar en datasets de gran volumen, como los datasets de Tipster (TREC), los que son de dominio público. En vez de eso, y debido a las limitaciones de acceso a recursos computacionales, los experimentos fueron llevados a cabo en datasets de tamaño más pequeño, lo que permitió controlar el uso de recursos disponibles para este estudio. Aunque esta limitación es importante, no limita la validez de las conclusiones, ya que los cuatro datasets usados en la validación experimental se utilizan frecuentemente en estudios de este tipo. Sería deseable superar estas limitaciones con un trabajo que involucre el estudiar diferentes aspectos de la eficiencia de estos métodos, lo que permitirá que escalen en colecciones documentales más grandes. Sin embargo, el estudio de estos aspectos, sobrepasa los objetivos de este artículo, a pesar de que ellos son fundamentales para la aplicabilidad de estos métodos.

Capítulo 7

Conlusiones y Trabajos Futuros

En este capítulo resumiremos los principales resultados y contribuciones de esta investigación. También mostraremos algunas opciones que podríamos tomar para trabajos futuros.

7.1. Conclusiones de la Tesis

En esta tesis se ha estudiado la aplicación de métodos de ensamblado utilizando modelos de temas para diferentes configuraciones y colecciones documentales, en la tarea de recuperación de información adhoc. Para esto usamos Latent Dirichlet Allocation como inductor base ya que es uno de los métodos más utilizados en modelamiento de temas. A continuación, los modelos se combinan utilizando diferentes estrategias con los documentos del corpus (sin muestreo, muestreo uniforme y muestreo adaptativo).

Nuestra hipótesis inicial postula que podemos mejorar el desempeño del método base usando ensamblados, por lo que se propusieron diferentes configuraciones del estado del arte, y la propuesta. Todas ellas fueron probadas en la tarea de ranking de documentos y el desempeño se midió de acuerdo a ciertos criterios de relevancia de cada dataset.

En general se puede ver que el desempeño del método mejora en relación a las diferentes métricas presentadas. Las configuraciones mostradas permiten explorar esto, y en realizar la tarea de buena forma. Los métodos de ensamblado a los que llamamos **LDA Ens** (sin muestreo), **LDA Ens** (sin muestreo), **Bagg Ens** (muestreo uniforme) y **Ada Ens** (muestreo adaptativo) mostraron, en general, un buen desempeño en comparación al método base.

De los resultados experimentales obtenidos en el capítulo 6, podemos ver que el resultado depende de la configuración de ensamblado utilizado. El método propuesto obtiene mejor resultado de recuperación para listas de ranking para 5 o 10 elementos.

7.2. Contribuciones

Esta investigación usa métodos de ensamblado para la tarea de recuperación de información adhoc. En el momento de que esta investigación comenzó, la idea había sido poco explotada. Se han publicado algunos trabajos en relación a ensamblados y a la tarea de recuperación adhoc.

Este trabajo también explora los ensamblados de modelos de temas, los que abren nuevas direcciones para futuras investigaciones en ensamblados para este tipo de tareas.

En relación al objetivo **O1** se presentó una revisión bibliográfica de ensamblado de modelos de temas y conceptos relacionados con recuperación de información. Se incluye una taxonomía para mostrar la manera en que se pueden manipular los datos en este tipo de aplicaciones.

En relación al objetivo **O2** se desarrolla un marco teórico para construir ensamblados de modelos de temas.

En relación al objetivo **O3** se construyó una familia de ensamblados de modelos de temas, cuyos métodos permiten realizar tareas de recuperación de información adhoc. Estos métodos se basaron en los métodos de ensamblado tradicionales.

En relación al objetivo **O4**, las métricas anteriores se aplicaron a conjuntos de datos reales usados en la literatura. Los datos usados corresponden a 4 colecciones estándar de documentos (CACM, CISI,CRAN,MED) [**Col19**].

7.3. Trabajos Futuros

En este sentido se proponen varias opciones y lı'neas de investigación a seguir. Primero se propone revisar y estudiar teóricamente los algoritmos de ensamblado propuestos. Por otro lado, pueden explorarse nuevas formas de ensamblar, ya que en la literatura existen una serie de métodos para diferentes aplicaciones.

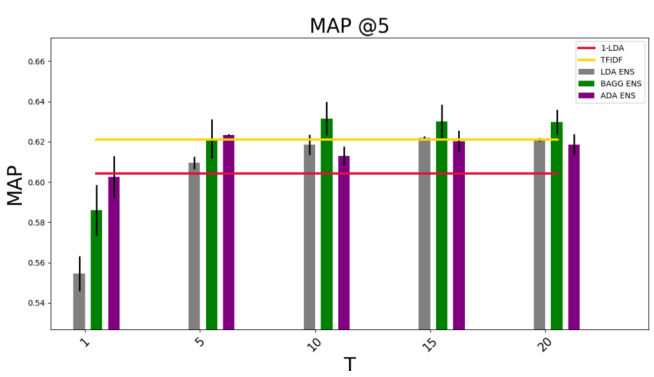
Por otro lado, podemos estudiar el efecto de la coherencia de temas en los ensamblados, así como otros modelos de temas diferentes al utilizado en este estudio.

Finalmente, puede estudiarse en efecto de estas metodología en otro tipos de conjuntos de datos, como los de Tipster, que también son ampliamente utilizados en la literatura.

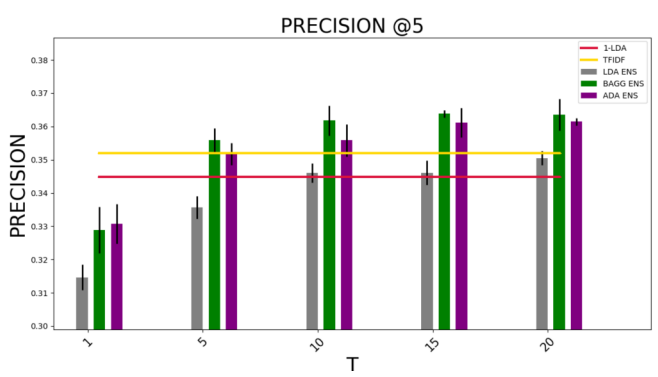
Apéndice A

Appendix

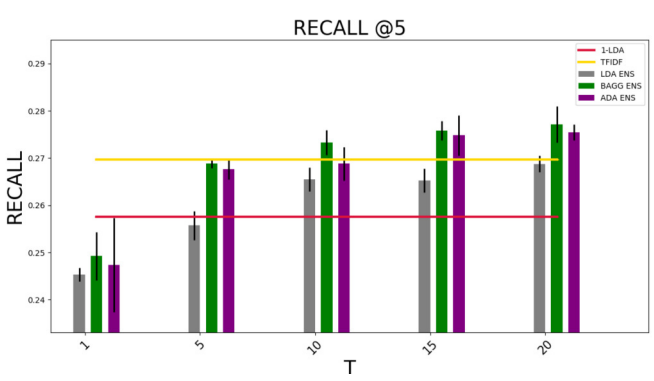
A.1. Imágenes CRAN



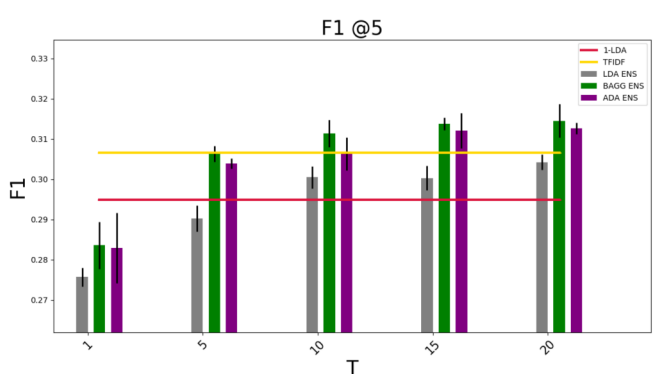
(a) MAP



(b) PRECISION

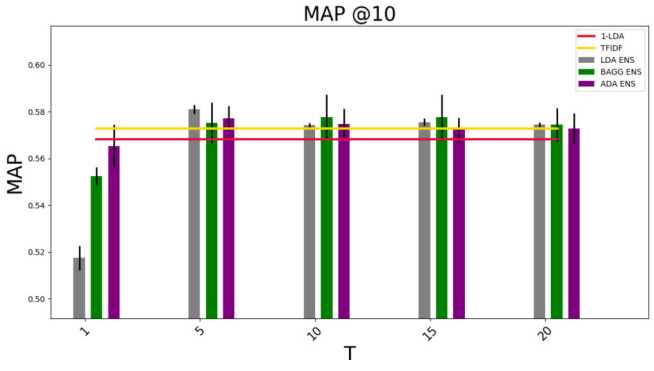


(c) RECALL

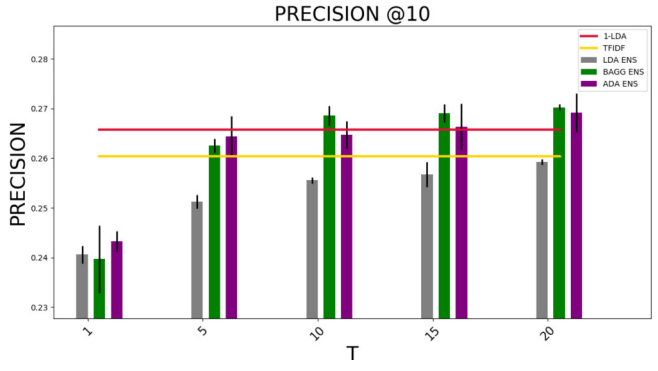


(d) F1

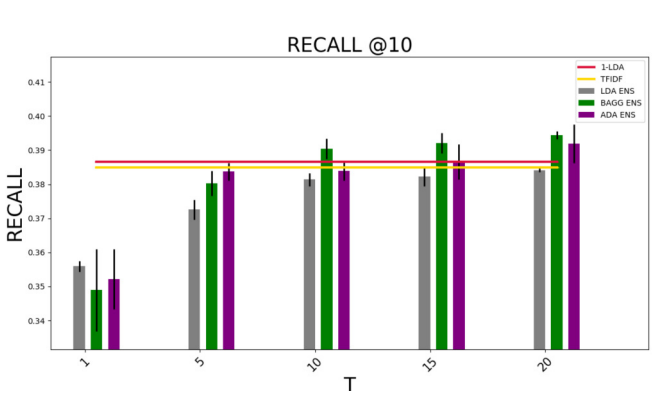
Figura A.1. CRAN @5



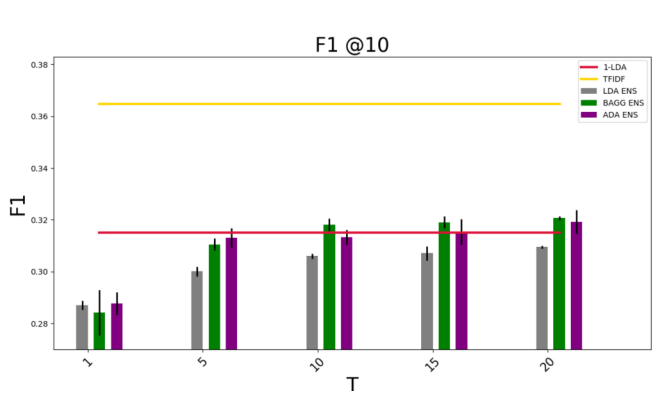
(A) MAP



(B) PRECISION

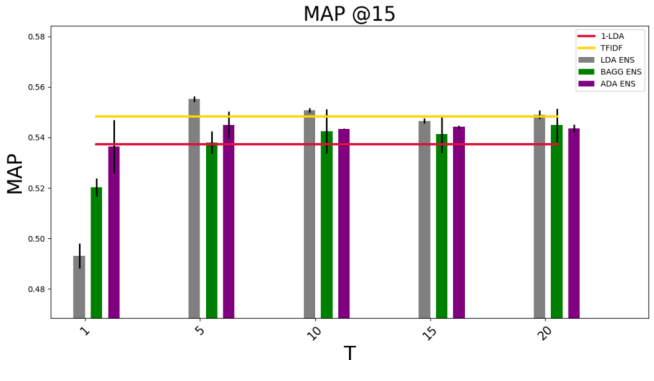


(C) RECALL

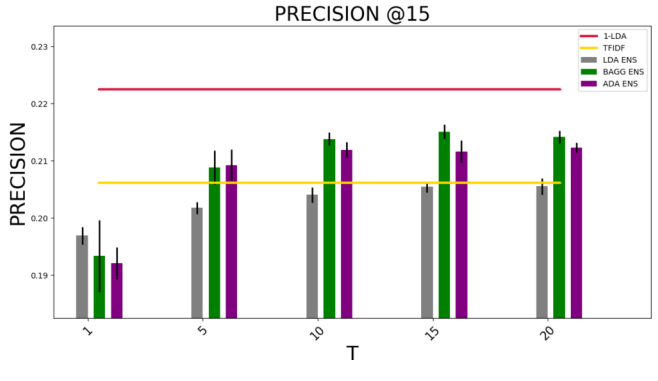


(D) F1

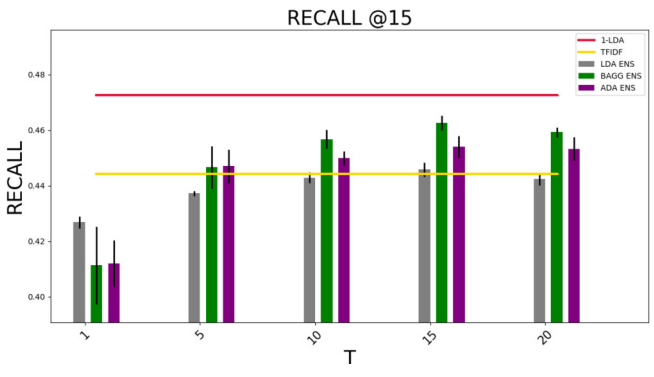
FIGURA A.2. CRAN @10



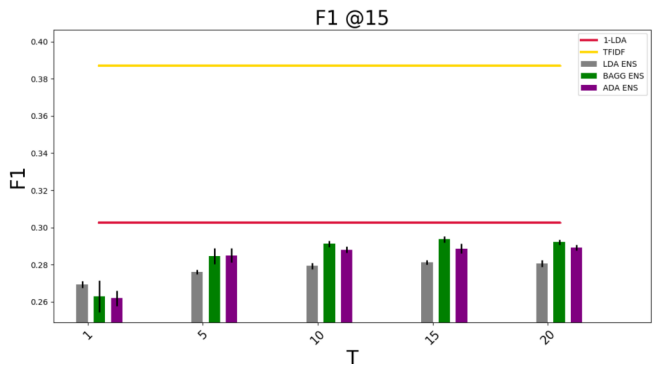
(A) MAP



(B) PRECISION

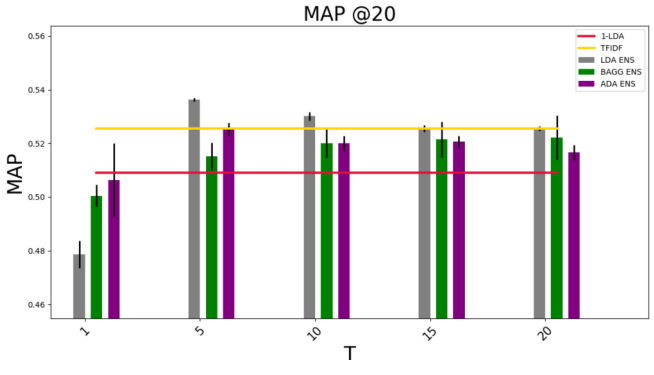


(C) RECALL

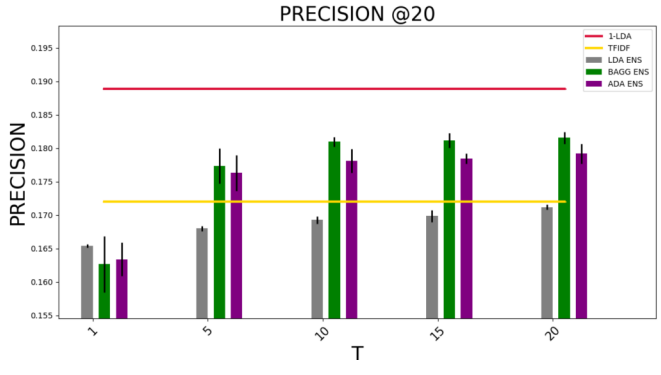


(D) F1

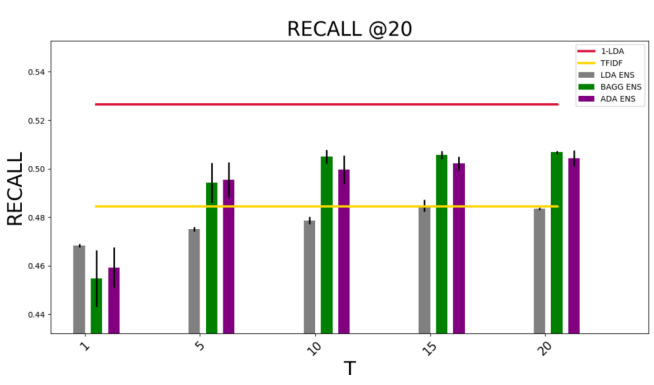
FIGURA A.3. CRAN @15



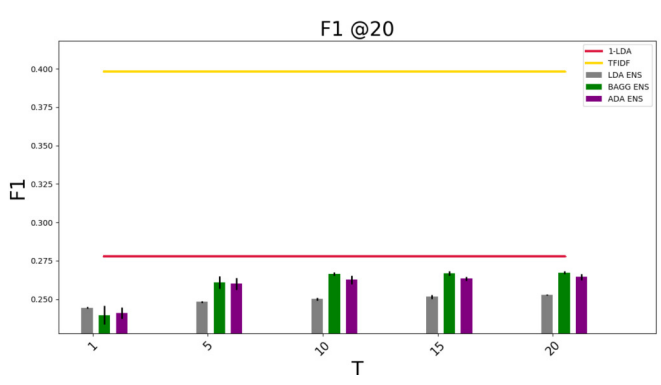
(A) MAP



(B) PRECISION



(C) RECALL



(D) F1

FIGURA A.4. CRAN @20

A.2. Tablas CRAN

A.2.1. CRAN MAP .

Metodo \T	1	5	10	15	20
1-LDA	0.6043 ± 0.0000	0.6043 ± 0.0000	0.6043 ± 0.0000	0.6043 ± 0.0000	0.6043 ± 0.0000
TFIDF	0.62096 ± 0.0000	0.62096 ± 0.0000	0.62096 ± 0.0000	0.62096 ± 0.0000	0.62096 ± 0.0000
LDA ENS	0.55441 ± 0.0087	0.60955 ± 0.00307	0.61853 ± 0.00513	0.62177 ± 0.00111	0.62101 ± 0.00078
BAGG ENS	0.58603 ± 0.01265	0.62153 ± 0.00971	0.63158 ± 0.00815	0.63018 ± 0.00834	0.62974 ± 0.00625
ADA ENS	0.60254 ± 0.01028	0.62327 ± 0.0006	0.61305 ± 0.00456	0.62027 ± 0.0053	0.61856 ± 0.0052
CRAN MAP P@5					
Metodo \T	1	5	10	15	20
1-LDA	0.56802 ± 0.0000	0.56802 ± 0.0000	0.56802 ± 0.0000	0.56802 ± 0.0000	0.56802 ± 0.0000
TFIDF	0.57274 ± 0.0000	0.57274 ± 0.0000	0.57274 ± 0.0000	0.57274 ± 0.0000	0.57274 ± 0.0000
LDA ENS	0.51736 ± 0.00523	0.58102 ± 0.00198	0.57416 ± 0.00093	0.57551 ± 0.00156	0.57449 ± 0.00095
BAGG ENS	0.55242 ± 0.00374	0.57505 ± 0.00875	0.57762 ± 0.0096	0.57749 ± 0.00989	0.57435 ± 0.00715
ADA ENS	0.56531 ± 0.00912	0.57709 ± 0.00524	0.57476 ± 0.00637	0.5723 ± 0.00511	0.57286 ± 0.00638
CRAN MAP P@10					
Metodo \T	1	5	10	15	20
1-LDA	0.53727 ± 0.0000	0.53727 ± 0.0000	0.53727 ± 0.0000	0.53727 ± 0.0000	0.53727 ± 0.0000
TFIDF	0.54838 ± 0.0000	0.54838 ± 0.0000	0.54838 ± 0.0000	0.54838 ± 0.0000	0.54838 ± 0.0000
LDA ENS	0.49305 ± 0.00498	0.55533 ± 0.00113	0.55084 ± 0.0009	0.54659 ± 0.00106	0.54905 ± 0.00167
BAGG ENS	0.52021 ± 0.00359	0.53801 ± 0.00435	0.54239 ± 0.00874	0.54127 ± 0.00736	0.5449 ± 0.0066
ADA ENS	0.53646 ± 0.01059	0.54501 ± 0.00524	0.54339 ± 0.00012	0.54414 ± 0.00059	0.54368 ± 0.00136
CRAN MAP P@15					
Metodo \T	1	5	10	15	20
1-LDA	0.50915 ± 0.0000	0.50915 ± 0.0000	0.50915 ± 0.0000	0.50915 ± 0.0000	0.50915 ± 0.0000
TFIDF	0.5256 ± 0.0000	0.5256 ± 0.0000	0.5256 ± 0.0000	0.5256 ± 0.0000	0.5256 ± 0.0000
LDA ENS	0.47868 ± 0.00505	0.53633 ± 0.0007	0.53013 ± 0.00164	0.52555 ± 0.00129	0.52549 ± 0.00082
BAGG ENS	0.50049 ± 0.00407	0.5152 ± 0.00501	0.52014 ± 0.00526	0.52144 ± 0.00672	0.52215 ± 0.00817
ADA ENS	0.5064 ± 0.01356	0.52515 ± 0.00243	0.51996 ± 0.00294	0.52064 ± 0.00213	0.5166 ± 0.00279
CRAN MAP P@20					

FIGURA A.5. CRAN MAP

A.2.2. CRAN Precision .

Metodo \T	1	5	10	15	20
1-LDA	0.34489 ± 0.0000	0.34489 ± 0.0000	0.34489 ± 0.0000	0.34489 ± 0.0000	0.34489 ± 0.0000
TFIDF	0.352 ± 0.0000	0.352 ± 0.0000	0.352 ± 0.0000	0.352 ± 0.0000	0.352 ± 0.0000
LDA ENS	0.31467 ± 0.00384	0.3357 ± 0.00343	0.34607 ± 0.00293	0.34607 ± 0.00365	0.35052 ± 0.0021
BAGG ENS	0.32889 ± 0.00692	0.35585 ± 0.00358	0.36178 ± 0.00453	0.36385 ± 0.00111	0.36356 ± 0.00476
ADA ENS	0.33067 ± 0.00594	0.3517 ± 0.00327	0.35585 ± 0.00483	0.36119 ± 0.00443	0.36148 ± 0.00111

CRAN Precision@5

Metodo \T	1	5	10	15	20
1-LDA	0.26578 ± 0.0000	0.26578 ± 0.0000	0.26578 ± 0.0000	0.26578 ± 0.0000	0.26578 ± 0.0000
TFIDF	0.26044 ± 0.0000	0.26044 ± 0.0000	0.26044 ± 0.0000	0.26044 ± 0.0000	0.26044 ± 0.0000
LDA ENS	0.24059 ± 0.00179	0.25126 ± 0.00137	0.25556 ± 0.00063	0.25674 ± 0.00255	0.25926 ± 0.00055
BAGG ENS	0.2397 ± 0.00681	0.26252 ± 0.00147	0.26859 ± 0.002	0.26904 ± 0.00183	0.27022 ± 0.00063
ADA ENS	0.24326 ± 0.0021	0.26444 ± 0.00409	0.26474 ± 0.00272	0.26637 ± 0.00459	0.26919 ± 0.00388

CRAN Precision@10

Metodo \T	1	5	10	15	20
1-LDA	0.22252 ± 0.0000	0.22252 ± 0.0000	0.22252 ± 0.0000	0.22252 ± 0.0000	0.22252 ± 0.0000
TFIDF	0.20622 ± 0.0000	0.20622 ± 0.0000	0.20622 ± 0.0000	0.20622 ± 0.0000	0.20622 ± 0.0000
LDA ENS	0.19694 ± 0.00156	0.20178 ± 0.00105	0.20405 ± 0.00133	0.20543 ± 0.00098	0.20553 ± 0.00142
BAGG ENS	0.19338 ± 0.0063	0.20889 ± 0.00297	0.21383 ± 0.00114	0.21511 ± 0.00126	0.21422 ± 0.00111
ADA ENS	0.1921 ± 0.00275	0.20928 ± 0.00275	0.21195 ± 0.00133	0.21165 ± 0.00194	0.21235 ± 0.00085

CRAN Precision@15

Metodo \T	1	5	10	15	20
1-LDA	0.18889 ± 0.0000	0.18889 ± 0.0000	0.18889 ± 0.0000	0.18889 ± 0.0000	0.18889 ± 0.0000
TFIDF	0.172 ± 0.0000	0.172 ± 0.0000	0.172 ± 0.0000	0.172 ± 0.0000	0.172 ± 0.0000
LDA ENS	0.16541 ± 0.00028	0.168 ± 0.00036	0.16926 ± 0.00058	0.16985 ± 0.00091	0.17119 ± 0.00038
BAGG ENS	0.16267 ± 0.00419	0.17733 ± 0.00264	0.18096 ± 0.00069	0.18119 ± 0.00111	0.18156 ± 0.00091
ADA ENS	0.16341 ± 0.00248	0.1763 ± 0.00266	0.17815 ± 0.00179	0.17844 ± 0.00079	0.17919 ± 0.00147

CRAN Precision@20

Figura A.6. cran precision

A.2.3. CRAN Recall .

Metodo \T	1	5	10	15	20
1-LDA	0.25761 ± 0.0000	0.25761 ± 0.0000	0.25761 ± 0.0000	0.25761 ± 0.0000	0.25761 ± 0.0000
TFIDF	0.26972 ± 0.0000	0.26972 ± 0.0000	0.26972 ± 0.0000	0.26972 ± 0.0000	0.26972 ± 0.0000
LDA ENS	0.24534 ± 0.00143	0.25573 ± 0.00303	0.2655 ± 0.00255	0.26525 ± 0.0025	0.26878 ± 0.00172
BAGG ENS	0.24925 ± 0.00507	0.26882 ± 0.00099	0.27328 ± 0.00265	0.27581 ± 0.00208	0.27717 ± 0.00383
ADA ENS	0.24737 ± 0.00999	0.26762 ± 0.0021	0.26882 ± 0.00353	0.27478 ± 0.00423	0.27544 ± 0.00171

CRAN Recall@5

Metodo \T	1	5	10	15	20
1-LDA	0.38661 ± 0.0000	0.38661 ± 0.0000	0.38661 ± 0.0000	0.38661 ± 0.0000	0.38661 ± 0.0000
TFIDF	0.38497 ± 0.0000	0.38497 ± 0.0000	0.38497 ± 0.0000	0.38497 ± 0.0000	0.38497 ± 0.0000
LDA ENS	0.35592 ± 0.00163	0.37256 ± 0.00294	0.3814 ± 0.0019	0.3823 ± 0.00287	0.38411 ± 0.00057
BAGG ENS	0.34896 ± 0.01204	0.38018 ± 0.00366	0.3904 ± 0.00307	0.39204 ± 0.00295	0.39439 ± 0.00109
ADA ENS	0.3521 ± 0.00881	0.38367 ± 0.00262	0.38392 ± 0.00276	0.38654 ± 0.00512	0.39188 ± 0.00565

CRAN Recall@10

Metodo \T	1	5	10	15	20
1-LDA	0.47249 ± 0.0000	0.47249 ± 0.0000	0.47249 ± 0.0000	0.47249 ± 0.0000	0.47249 ± 0.0000
TFIDF	0.44413 ± 0.0000	0.44413 ± 0.0000	0.44413 ± 0.0000	0.44413 ± 0.0000	0.44413 ± 0.0000
LDA ENS	0.4268 ± 0.00213	0.4372 ± 0.00085	0.44287 ± 0.00189	0.44575 ± 0.00255	0.44235 ± 0.00224
BAGG ENS	0.4113 ± 0.01393	0.44657 ± 0.00758	0.45674 ± 0.00342	0.46265 ± 0.00265	0.45921 ± 0.00167
ADA ENS	0.412 ± 0.00841	0.44701 ± 0.00604	0.44982 ± 0.00254	0.45396 ± 0.00384	0.45327 ± 0.00421

CRAN Recall@15

Metodo \T	1	5	10	15	20
1-LDA	0.52653 ± 0.0000	0.52653 ± 0.0000	0.52653 ± 0.0000	0.52653 ± 0.0000	0.52653 ± 0.0000
TFIDF	0.48454 ± 0.0000	0.48454 ± 0.0000	0.48454 ± 0.0000	0.48454 ± 0.0000	0.48454 ± 0.0000
LDA ENS	0.46838 ± 0.00071	0.4751 ± 0.00093	0.4787 ± 0.00157	0.48474 ± 0.00249	0.48352 ± 0.00045
BAGG ENS	0.45478 ± 0.01172	0.49425 ± 0.00826	0.5051 ± 0.00286	0.50571 ± 0.00165	0.50682 ± 0.00059
ADA ENS	0.45929 ± 0.00829	0.49544 ± 0.00734	0.49975 ± 0.00583	0.5022 ± 0.00272	0.50439 ± 0.00327

CRAN Recall@20

FIGURA A.7. CRAN RECALL

A.2.4. CRAN F1 .

Metodo \T	1	5	10	15	20
1-LDA	0.29493 ± 0.0000	0.29493 ± 0.0000	0.29493 ± 0.0000	0.29493 ± 0.0000	0.29493 ± 0.0000
TFIDF	0.3066 ± 0.0000	0.3066 ± 0.0000	0.3066 ± 0.0000	0.3066 ± 0.0000	0.3066 ± 0.0000
LDA ENS	0.27571 ± 0.00235	0.29031 ± 0.00322	0.30048 ± 0.00271	0.30032 ± 0.00298	0.30425 ± 0.00189
BAGG ENS	0.28358 ± 0.00583	0.30627 ± 0.00196	0.31136 ± 0.00339	0.31377 ± 0.00161	0.31454 ± 0.00421
ADA ENS	0.28298 ± 0.00873	0.30394 ± 0.00131	0.30627 ± 0.00407	0.31211 ± 0.00434	0.31265 ± 0.0014

CRAN F1@5

Metodo \T	1	5	10	15	20
1-LDA	0.315 ± 0.0000	0.315 ± 0.0000	0.315 ± 0.0000	0.315 ± 0.0000	0.315 ± 0.0000
TFIDF	0.36475 ± 0.0000	0.36475 ± 0.0000	0.36475 ± 0.0000	0.36475 ± 0.0000	0.36475 ± 0.0000
LDA ENS	0.28711 ± 0.0018	0.30011 ± 0.0019	0.30604 ± 0.00102	0.30719 ± 0.00275	0.30957 ± 0.00055
BAGG ENS	0.28419 ± 0.00875	0.31058 ± 0.00223	0.31824 ± 0.00241	0.31909 ± 0.00225	0.32071 ± 0.0008
ADA ENS	0.28771 ± 0.00432	0.31309 ± 0.00374	0.31338 ± 0.00282	0.31539 ± 0.00489	0.31915 ± 0.0046

CRAN F1@10

Metodo \T	1	5	10	15	20
1-LDA	0.30255 ± 0.0000	0.30255 ± 0.0000	0.30255 ± 0.0000	0.30255 ± 0.0000	0.30255 ± 0.0000
TFIDF	0.38708 ± 0.0000	0.38708 ± 0.0000	0.38708 ± 0.0000	0.38708 ± 0.0000	0.38708 ± 0.0000
LDA ENS	0.26951 ± 0.00187	0.27612 ± 0.00112	0.27938 ± 0.00162	0.28125 ± 0.00127	0.28066 ± 0.00177
BAGG ENS	0.26307 ± 0.00861	0.28464 ± 0.0043	0.29129 ± 0.00169	0.29368 ± 0.0017	0.29215 ± 0.00134
ADA ENS	0.26202 ± 0.00406	0.28509 ± 0.00377	0.28814 ± 0.00175	0.2887 ± 0.00256	0.2892 ± 0.00162

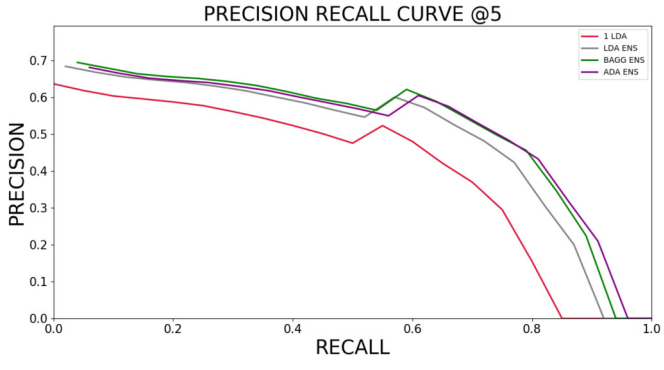
CRAN F1@15

Metodo \T	1	5	10	15	20
1-LDA	0.27803 ± 0.0000	0.27803 ± 0.0000	0.27803 ± 0.0000	0.27803 ± 0.0000	0.27803 ± 0.0000
TFIDF	0.39836 ± 0.0000	0.39836 ± 0.0000	0.39836 ± 0.0000	0.39836 ± 0.0000	0.39836 ± 0.0000
LDA ENS	0.24448 ± 0.0004	0.24823 ± 0.00049	0.25009 ± 0.00085	0.25156 ± 0.0013	0.25285 ± 0.00047
BAGG ENS	0.23962 ± 0.00614	0.26102 ± 0.00401	0.26646 ± 0.00113	0.26679 ± 0.00142	0.26734 ± 0.00106
ADA ENS	0.24105 ± 0.00374	0.26006 ± 0.0039	0.26266 ± 0.00275	0.26332 ± 0.00121	0.26443 ± 0.00198

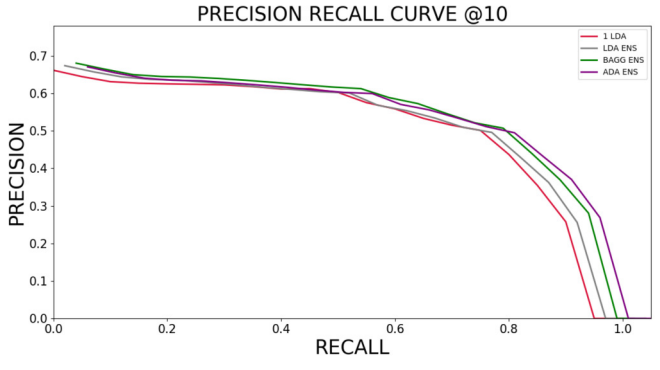
CRAN F1@20

Figura A.8. CRAN F1

A.3. Curvas Precision Recall CRAN



(A) CURVA CRAN @5



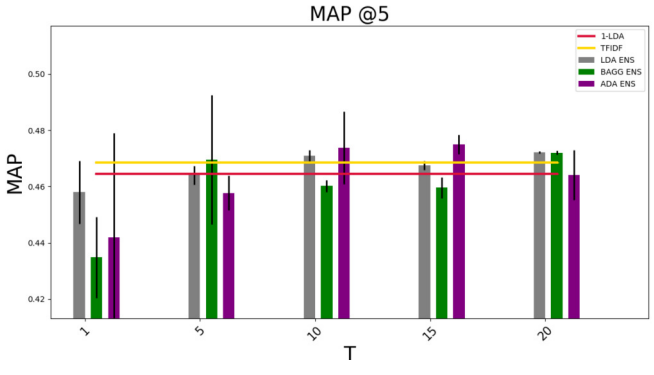
(B) CURVA CRAN @10

(C) CURVA CRAN @15

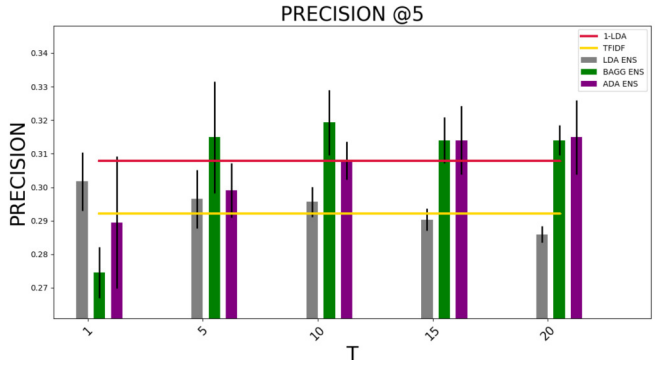
(D) CURVA CRAN @20

FIGURA A.9. CURVAS PRECISION RECALL CRAN

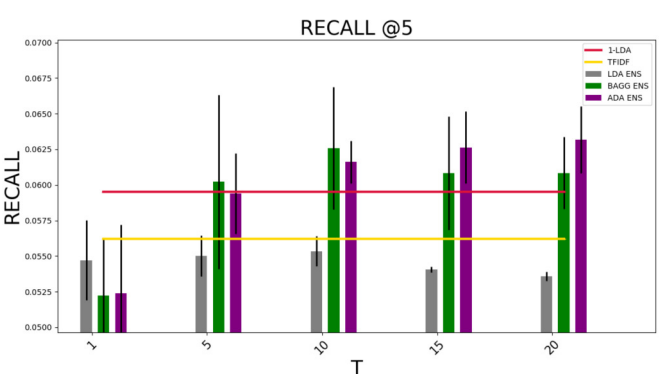
A.4. IMÁGENES CISI



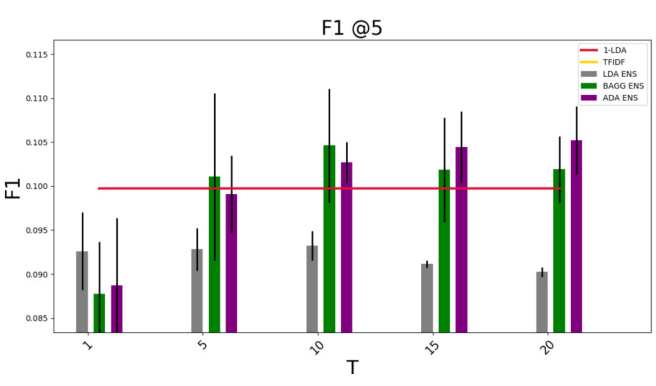
(A) MAP



(B) PRECISION

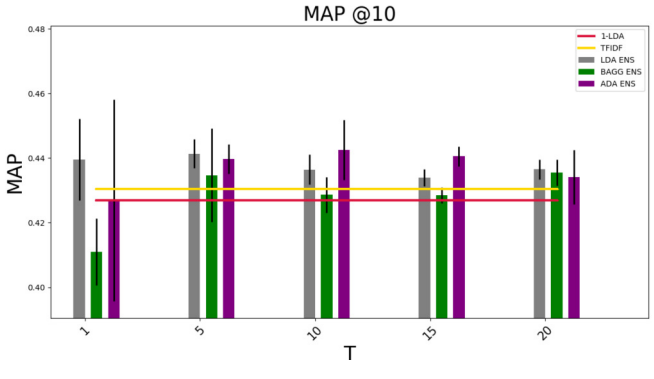


(C) RECALL

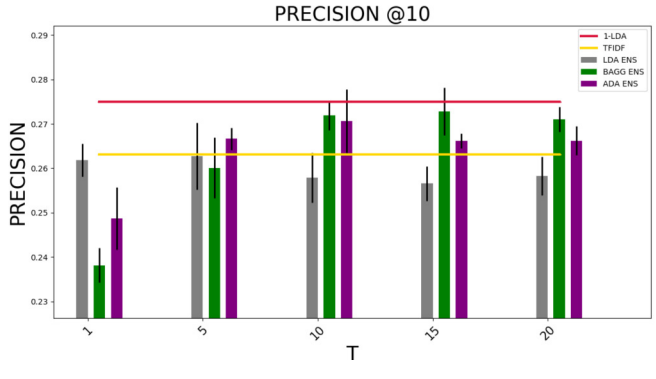


(D) F1

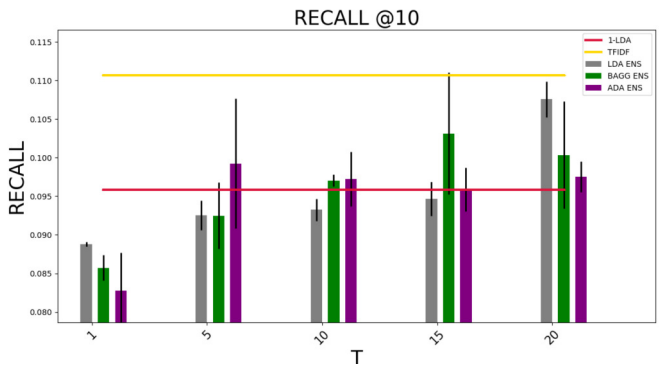
FIGURA A.10. CISI @5



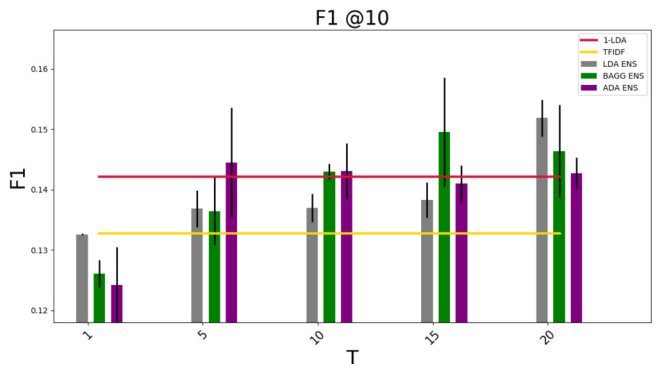
(A) MAP



(B) PRECISION

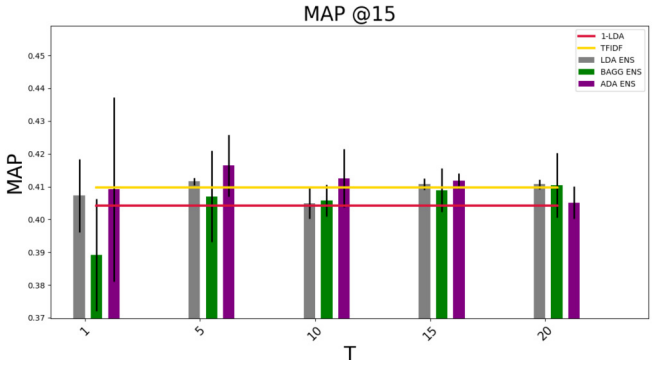


(C) RECALL

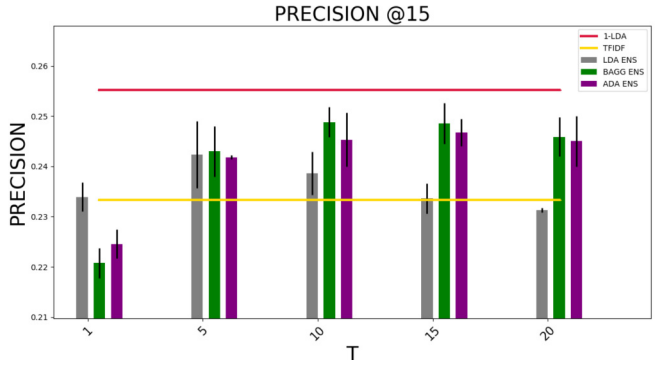


(D) F1

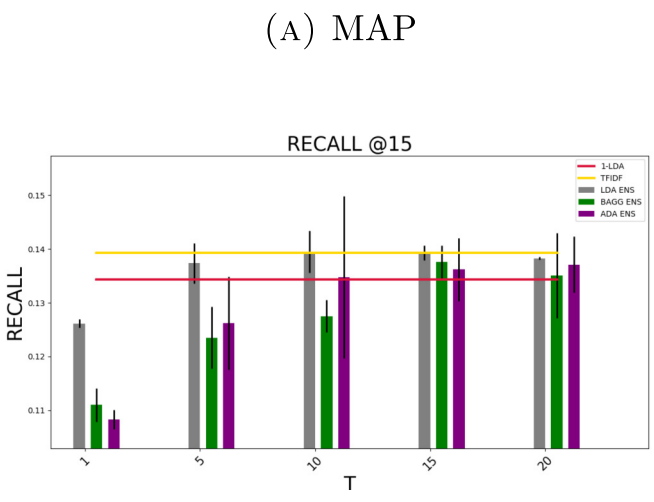
FIGURA A.11. CISI @10



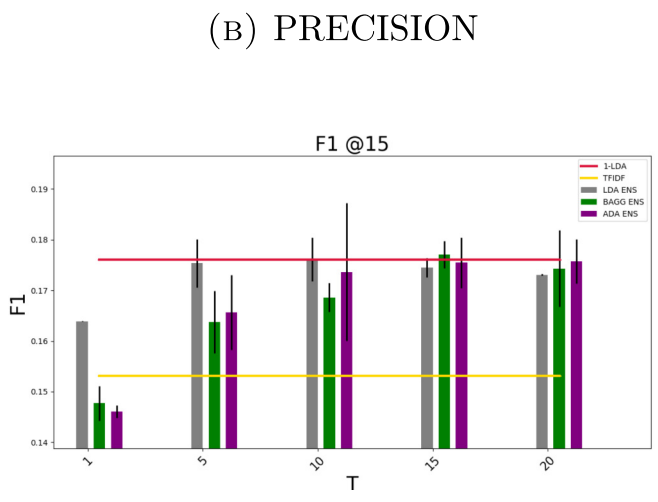
(A) MAP



(B) PRECISION

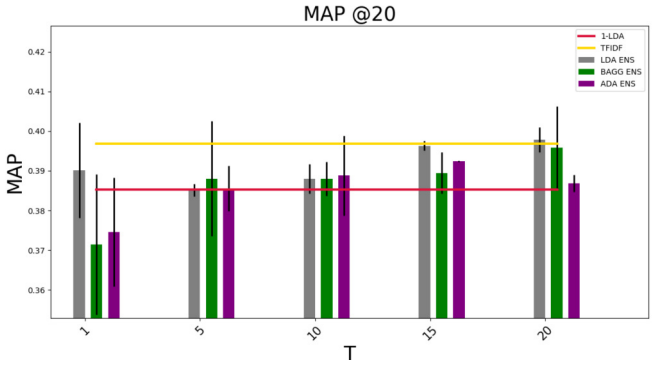


(C) RECALL

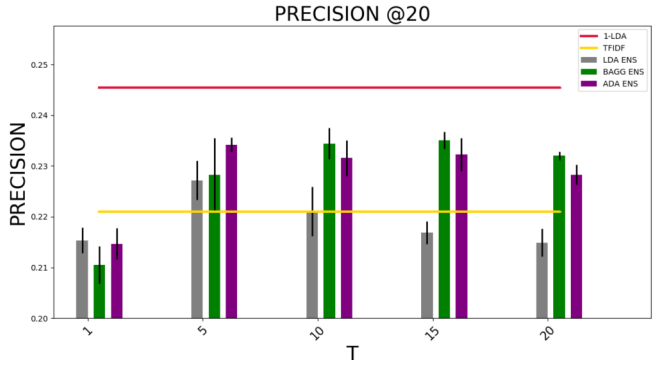


(D) F1

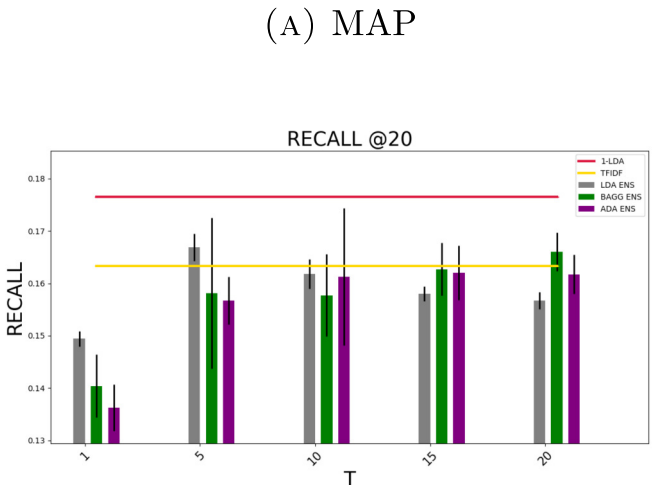
FIGURA A.12. CISI @15



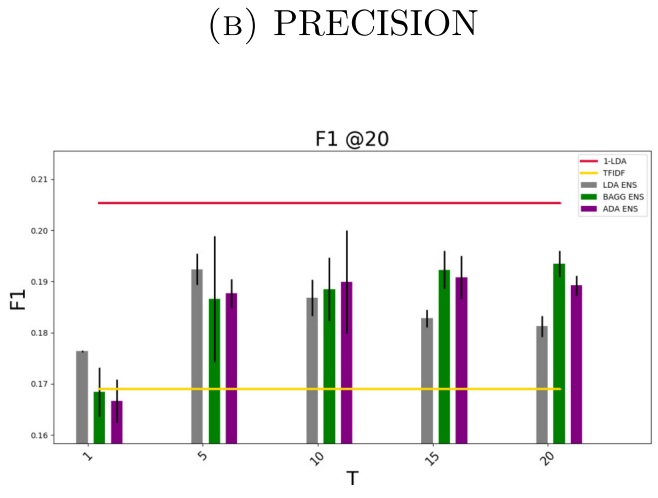
(A) MAP



(B) PRECISION



(C) RECALL



(D) F1

FIGURA A.13. CISI @20

A.5. Tablas CISI

A.5.1. CISI MAP .

Metodo \T	1	5	10	15	20
1-LDA	0.46446 ± 0.0000	0.46446 ± 0.0000	0.46446 ± 0.0000	0.46446 ± 0.0000	0.46446 ± 0.0000
TFIDF	0.46848 ± 0.0000	0.46848 ± 0.0000	0.46848 ± 0.0000	0.46848 ± 0.0000	0.46848 ± 0.0000
LDA ENS	0.45796 ± 0.01123	0.46404 ± 0.00333	0.47096 ± 0.002	0.46751 ± 0.00169	0.47218 ± 0.0004
BAGG ENS	0.43479 ± 0.01435	0.46957 ± 0.02303	0.46023 ± 0.00208	0.45958 ± 0.00377	0.47205 ± 0.00071
ADA ENS	0.44186 ± 0.03716	0.45772 ± 0.00621	0.47383 ± 0.0129	0.47505 ± 0.00343	0.46416 ± 0.00885

CISI MAP P@5

Metodo \T	1	5	10	15	20
1-LDA	0.42691 ± 0.0000	0.42691 ± 0.0000	0.42691 ± 0.0000	0.42691 ± 0.0000	0.42691 ± 0.0000
TFIDF	0.43039 ± 0.0000	0.43039 ± 0.0000	0.43039 ± 0.0000	0.43039 ± 0.0000	0.43039 ± 0.0000
LDA ENS	0.43948 ± 0.01264	0.44137 ± 0.00443	0.43644 ± 0.00465	0.43397 ± 0.00267	0.4365 ± 0.00311
BAGG ENS	0.41096 ± 0.01035	0.43472 ± 0.01446	0.42863 ± 0.0055	0.42846 ± 0.00245	0.43555 ± 0.00403
ADA ENS	0.42692 ± 0.03118	0.43968 ± 0.00457	0.44256 ± 0.00927	0.44054 ± 0.00308	0.43414 ± 0.00839

CISI MAP P@10

Metodo \T	1	5	10	15	20
1-LDA	0.40424 ± 0.0000	0.40424 ± 0.0000	0.40424 ± 0.0000	0.40424 ± 0.0000	0.40424 ± 0.0000
TFIDF	0.40982 ± 0.0000	0.40982 ± 0.0000	0.40982 ± 0.0000	0.40982 ± 0.0000	0.40982 ± 0.0000
LDA ENS	0.40729 ± 0.01114	0.41161 ± 0.00115	0.40497 ± 0.00482	0.41083 ± 0.00176	0.41072 ± 0.00143
BAGG ENS	0.38925 ± 0.01711	0.407 ± 0.01391	0.40583 ± 0.00487	0.40896 ± 0.00658	0.41045 ± 0.00986
ADA ENS	0.40916 ± 0.02807	0.41642 ± 0.00943	0.41259 ± 0.00893	0.41174 ± 0.00225	0.40515 ± 0.00487

CISI MAP P@15

Metodo \T	1	5	10	15	20
1-LDA	0.38535 ± 0.0000	0.38535 ± 0.0000	0.38535 ± 0.0000	0.38535 ± 0.0000	0.38535 ± 0.0000
TFIDF	0.39687 ± 0.0000	0.39687 ± 0.0000	0.39687 ± 0.0000	0.39687 ± 0.0000	0.39687 ± 0.0000
LDA ENS	0.39016 ± 0.01194	0.38514 ± 0.00163	0.38796 ± 0.00372	0.39633 ± 0.00125	0.39784 ± 0.00314
BAGG ENS	0.37145 ± 0.01762	0.38802 ± 0.01449	0.38799 ± 0.00434	0.38948 ± 0.00527	0.39586 ± 0.0104
ADA ENS	0.37455 ± 0.01371	0.38552 ± 0.00572	0.38881 ± 0.01008	0.39237 ± 0.00014	0.38684 ± 0.0021

CISI MAP P@20

Figura A.14. cisi map

A.5.2. CISI Precision .

Metodo \T	1	5	10	15	20
1-LDA	0.30789 ± 0.0000	0.30789 ± 0.0000	0.30789 ± 0.0000	0.30789 ± 0.0000	0.30789 ± 0.0000
TFIDF	0.29211 ± 0.0000	0.29211 ± 0.0000	0.29211 ± 0.0000	0.29211 ± 0.0000	0.29211 ± 0.0000
LDA ENS	0.30175 ± 0.00868	0.29649 ± 0.00868	0.29561 ± 0.00447	0.29035 ± 0.00328	0.28596 ± 0.00248
BAGG ENS	0.27456 ± 0.00755	0.31491 ± 0.01669	0.3193 ± 0.00969	0.31404 ± 0.00691	0.31404 ± 0.00447
ADA ENS	0.28947 ± 0.01969	0.29912 ± 0.00813	0.30789 ± 0.00568	0.31404 ± 0.01015	0.31491 ± 0.01103

CISI Precision@5

Metodo \T	1	5	10	15	20
1-LDA	0.275 ± 0.0000	0.275 ± 0.0000	0.275 ± 0.0000	0.275 ± 0.0000	0.275 ± 0.0000
TFIDF	0.26316 ± 0.0000	0.26316 ± 0.0000	0.26316 ± 0.0000	0.26316 ± 0.0000	0.26316 ± 0.0000
LDA ENS	0.26184 ± 0.00372	0.26272 ± 0.00755	0.25789 ± 0.00558	0.25658 ± 0.00387	0.25833 ± 0.00434
BAGG ENS	0.23816 ± 0.00387	0.26009 ± 0.00682	0.27193 ± 0.00328	0.27281 ± 0.00541	0.27105 ± 0.00284
ADA ENS	0.24868 ± 0.00704	0.26667 ± 0.00248	0.27061 ± 0.00715	0.26623 ± 0.00164	0.26623 ± 0.00328

CISI Precision@10

Metodo \T	1	5	10	15	20
1-LDA	0.25526 ± 0.0000	0.25526 ± 0.0000	0.25526 ± 0.0000	0.25526 ± 0.0000	0.25526 ± 0.0000
TFIDF	0.23333 ± 0.0000	0.23333 ± 0.0000	0.23333 ± 0.0000	0.23333 ± 0.0000	0.23333 ± 0.0000
LDA ENS	0.23392 ± 0.00289	0.2424 ± 0.00665	0.2386 ± 0.0043	0.23363 ± 0.00298	0.23129 ± 0.00041
BAGG ENS	0.22076 ± 0.00298	0.24298 ± 0.00501	0.24883 ± 0.00298	0.24854 ± 0.00407	0.24591 ± 0.00394
ADA ENS	0.22456 ± 0.00286	0.24181 ± 0.00041	0.24532 ± 0.00538	0.24678 ± 0.00271	0.24503 ± 0.00503

CISI Precision@15

Metodo \T	1	5	10	15	20
1-LDA	0.24539 ± 0.0000	0.24539 ± 0.0000	0.24539 ± 0.0000	0.24539 ± 0.0000	0.24539 ± 0.0000
TFIDF	0.22105 ± 0.0000	0.22105 ± 0.0000	0.22105 ± 0.0000	0.22105 ± 0.0000	0.22105 ± 0.0000
LDA ENS	0.21535 ± 0.00248	0.22719 ± 0.00381	0.22105 ± 0.00483	0.21689 ± 0.00224	0.21491 ± 0.00276
BAGG ENS	0.21053 ± 0.00372	0.22829 ± 0.00727	0.23443 ± 0.00305	0.23509 ± 0.00164	0.23202 ± 0.00082
ADA ENS	0.21469 ± 0.00305	0.23421 ± 0.00142	0.23158 ± 0.00352	0.23224 ± 0.00322	0.22829 ± 0.00194

CISI Precision@20

FIGURA A.15. CISI PRECISION

A.5.3. CISI Recall .

Metodo \T	1	5	10	15	20
1-LDA	0.05951 ± 0.0000	0.05951 ± 0.0000	0.05951 ± 0.0000	0.05951 ± 0.0000	0.05951 ± 0.0000
TFIDF	0.05621 ± 0.0000	0.05621 ± 0.0000	0.05621 ± 0.0000	0.05621 ± 0.0000	0.05621 ± 0.0000
LDA ENS	0.05471 ± 0.00281	0.05503 ± 0.00143	0.05535 ± 0.00107	0.05405 ± 0.00021	0.05357 ± 0.00032
BAGG ENS	0.05224 ± 0.00403	0.06022 ± 0.00613	0.06257 ± 0.00429	0.06083 ± 0.00399	0.06083 ± 0.00253
ADA ENS	0.0524 ± 0.00482	0.05939 ± 0.00282	0.06161 ± 0.00149	0.06264 ± 0.00251	0.06317 ± 0.00235
CISI Recall@5					
Metodo \T	1	5	10	15	20
1-LDA	0.09583 ± 0.0000	0.09583 ± 0.0000	0.09583 ± 0.0000	0.09583 ± 0.0000	0.09583 ± 0.0000
TFIDF	0.1107 ± 0.0000	0.1107 ± 0.0000	0.1107 ± 0.0000	0.1107 ± 0.0000	0.1107 ± 0.0000
LDA ENS	0.08876 ± 0.0003	0.09254 ± 0.00193	0.09326 ± 0.00144	0.09464 ± 0.00222	0.10757 ± 0.00233
BAGG ENS	0.08573 ± 0.00166	0.09248 ± 0.00431	0.09703 ± 0.00077	0.10314 ± 0.0079	0.10035 ± 0.00692
ADA ENS	0.08277 ± 0.00492	0.09924 ± 0.00843	0.09724 ± 0.00355	0.0959 ± 0.00282	0.0975 ± 0.002
CISI Recall@10					
1-LDA	0.13431 ± 0.0000	0.13431 ± 0.0000	0.13431 ± 0.0000	0.13431 ± 0.0000	0.13431 ± 0.0000
TFIDF	0.13932 ± 0.0000	0.13932 ± 0.0000	0.13932 ± 0.0000	0.13932 ± 0.0000	0.13932 ± 0.0000
LDA ENS	0.12614 ± 0.00078	0.13735 ± 0.00376	0.13954 ± 0.0039	0.13929 ± 0.00139	0.13827 ± 0.00024
BAGG ENS	0.111 ± 0.00313	0.12349 ± 0.00574	0.12751 ± 0.003	0.13755 ± 0.00313	0.13506 ± 0.00795
ADA ENS	0.10829 ± 0.00181	0.12619 ± 0.00862	0.13476 ± 0.01512	0.1362 ± 0.00588	0.13711 ± 0.00522
CISI Recall@15					
Metodo \T	1	5	10	15	20
1-LDA	0.17651 ± 0.0000	0.17651 ± 0.0000	0.17651 ± 0.0000	0.17651 ± 0.0000	0.17651 ± 0.0000
TFIDF	0.16331 ± 0.0000	0.16331 ± 0.0000	0.16331 ± 0.0000	0.16331 ± 0.0000	0.16331 ± 0.0000
LDA ENS	0.14941 ± 0.00146	0.16685 ± 0.00262	0.16177 ± 0.0028	0.15803 ± 0.0014	0.15673 ± 0.0016
BAGG ENS	0.14039 ± 0.00605	0.15807 ± 0.0144	0.15771 ± 0.00783	0.16271 ± 0.00501	0.166 ± 0.00369
ADA ENS	0.13621 ± 0.00443	0.15669 ± 0.00458	0.16128 ± 0.01309	0.16201 ± 0.00517	0.16171 ± 0.00373
CISI Recall@20					

Figura A.16. CISI RECALL

A.5.4. CISI F1 .

Metodo \T	1	5	10	15	20
1-LDA	0.09974 ± 0.0000	0.09974 ± 0.0000	0.09974 ± 0.0000	0.09974 ± 0.0000	0.09974 ± 0.0000
TFIDF	0.0777 ± 0.0000	0.0777 ± 0.0000	0.0777 ± 0.0000	0.0777 ± 0.0000	0.0777 ± 0.0000
LDA ENS	0.09261 ± 0.00445	0.09282 ± 0.00241	0.09324 ± 0.00167	0.09114 ± 0.00042	0.09023 ± 0.00058
BAGG ENS	0.08773 ± 0.00595	0.10107 ± 0.00952	0.10461 ± 0.0065	0.10188 ± 0.00593	0.10191 ± 0.00377
ADA ENS	0.0887 ± 0.0077	0.09909 ± 0.00437	0.10268 ± 0.00234	0.10444 ± 0.00405	0.10523 ± 0.00386

CISI F1@5

Metodo \T	1	5	10	15	20
1-LDA	0.14213 ± 0.0000	0.14213 ± 0.0000	0.14213 ± 0.0000	0.14213 ± 0.0000	0.14213 ± 0.0000
TFIDF	0.13274 ± 0.0000	0.13274 ± 0.0000	0.13274 ± 0.0000	0.13274 ± 0.0000	0.13274 ± 0.0000
LDA ENS	0.13257 ± 0.00015	0.13687 ± 0.00305	0.13698 ± 0.00233	0.13827 ± 0.00291	0.15189 ± 0.00307
BAGG ENS	0.12607 ± 0.00224	0.13643 ± 0.00561	0.14303 ± 0.00126	0.14958 ± 0.00901	0.14637 ± 0.00771
ADA ENS	0.12418 ± 0.00634	0.14448 ± 0.00912	0.14305 ± 0.0046	0.14098 ± 0.00303	0.14273 ± 0.0026

CISI F1@10

Metodo \T	1	5	10	15	20
1-LDA	0.17601 ± 0.0000	0.17601 ± 0.0000	0.17601 ± 0.0000	0.17601 ± 0.0000	0.17601 ± 0.0000
TFIDF	0.15307 ± 0.0000	0.15307 ± 0.0000	0.15307 ± 0.0000	0.15307 ± 0.0000	0.15307 ± 0.0000
LDA ENS	0.16389 ± 6e-05	0.17534 ± 0.00474	0.17609 ± 0.00427	0.17452 ± 0.0019	0.17307 ± 0.00013
BAGG ENS	0.14771 ± 0.00343	0.16373 ± 0.00616	0.16859 ± 0.00287	0.17706 ± 0.0027	0.17428 ± 0.00755
ADA ENS	0.1461 ± 0.00125	0.16567 ± 0.00741	0.17362 ± 0.01359	0.17545 ± 0.00496	0.17575 ± 0.00437

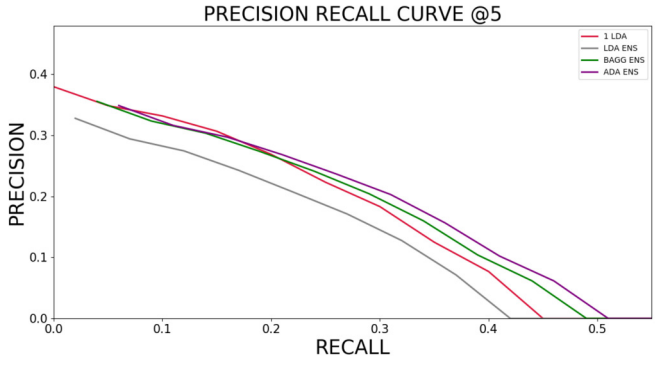
CISI F1@15

Metodo \T	1	5	10	15	20
1-LDA	0.20533 ± 0.0000	0.20533 ± 0.0000	0.20533 ± 0.0000	0.20533 ± 0.0000	0.20533 ± 0.0000
TFIDF	0.169 ± 0.0000	0.169 ± 0.0000	0.169 ± 0.0000	0.169 ± 0.0000	0.169 ± 0.0000
LDA ENS	0.1764 ± 0.00017	0.1924 ± 0.00306	0.18682 ± 0.00359	0.18284 ± 0.00172	0.18126 ± 0.00204
BAGG ENS	0.16838 ± 0.00481	0.1866 ± 0.01233	0.18848 ± 0.00623	0.19227 ± 0.00371	0.19351 ± 0.00255
ADA ENS	0.16666 ± 0.00425	0.18771 ± 0.00283	0.18994 ± 0.01011	0.19083 ± 0.00417	0.18927 ± 0.00194

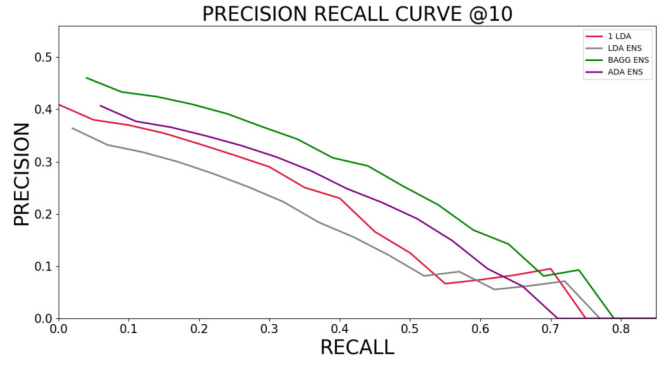
CISI F1@20

FIGURA A.17. CISI F1

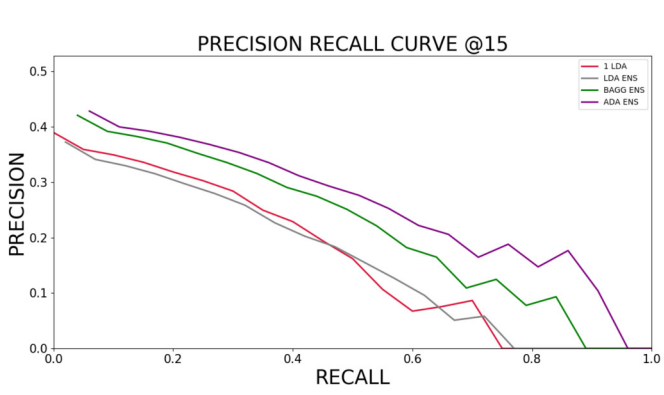
A.6. CURVAS PRECISION RECALL CISI



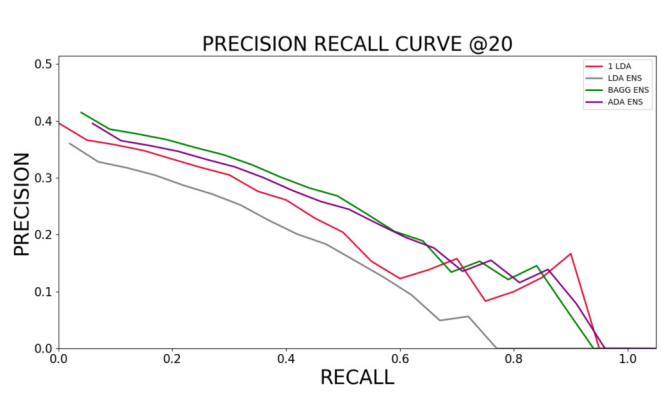
(A) CURVA CISI @5



(B) CURVA CISI @10



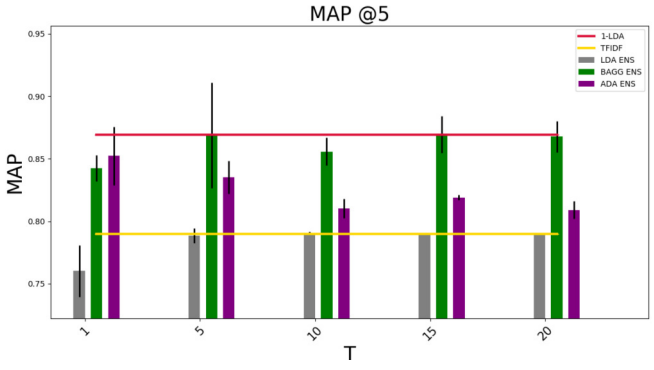
(C) CURVA CISI @15



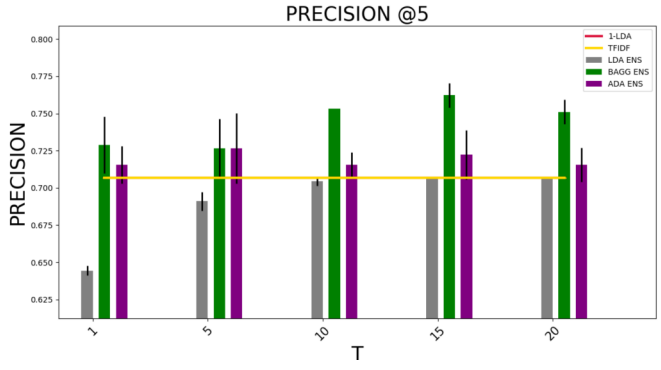
(D) CURVA CISI @20

FIGURA A.18. CURVAS PRECISION RECALL CISI

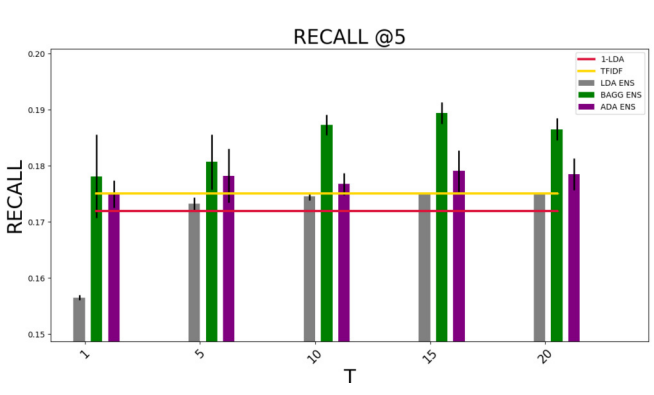
A.7. IMÁGENES MED



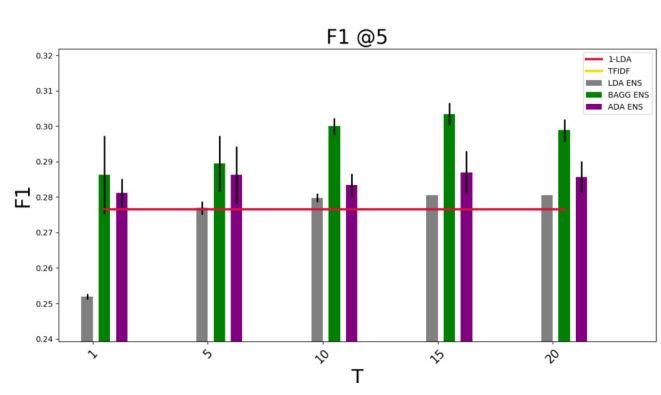
(A) MAP



(B) PRECISION

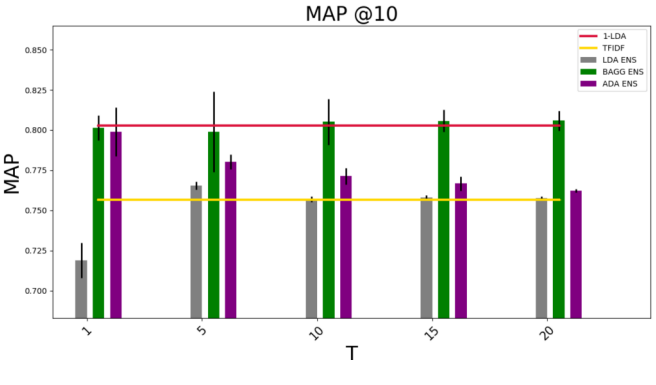


(C) RECALL

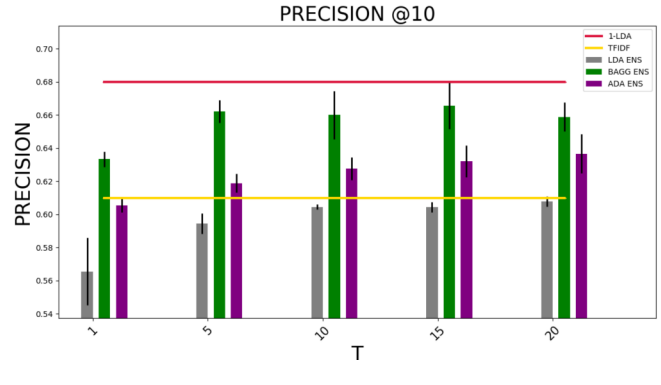


(D) F1

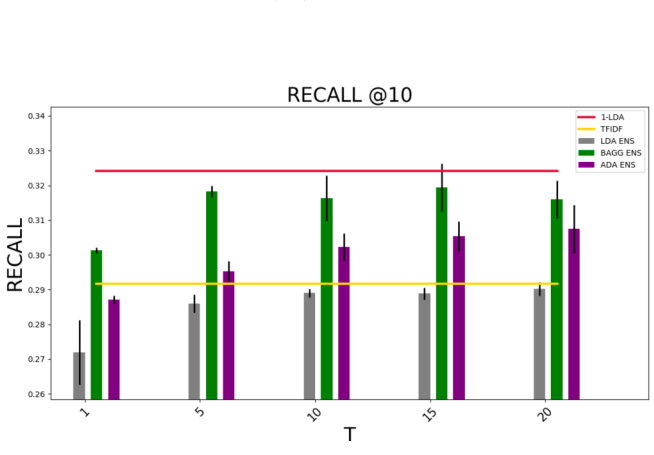
FIGURA A.19. MED @5



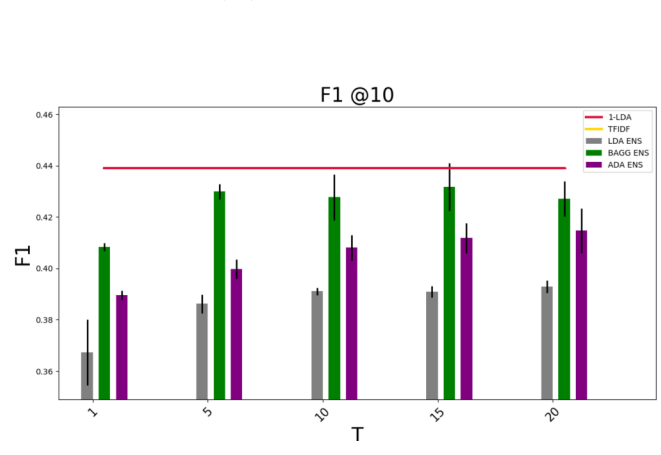
(A) MAP



(B) PRECISION

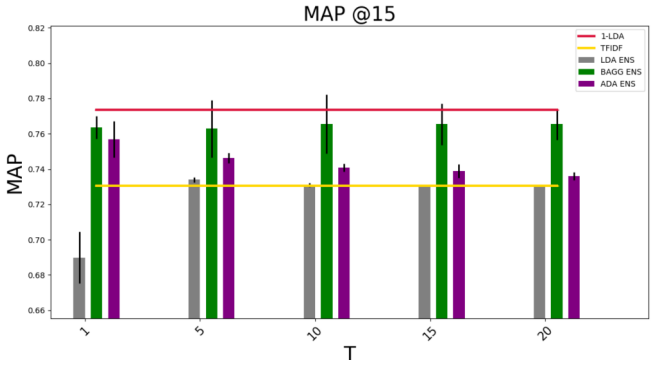


(C) RECALL

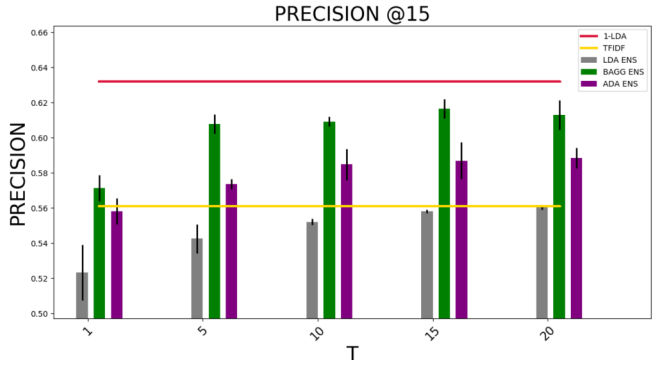


(D) F1

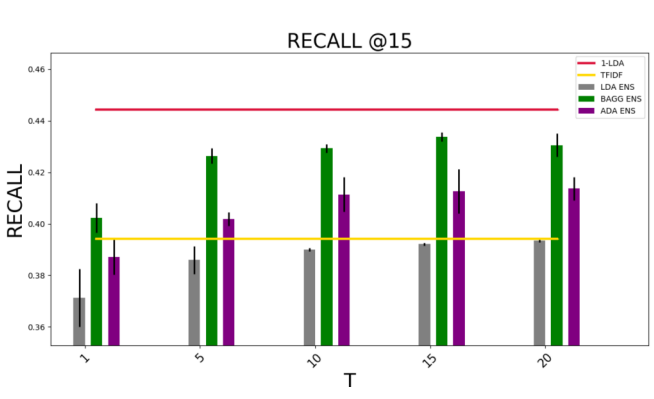
FIGURA A.20. MED @10



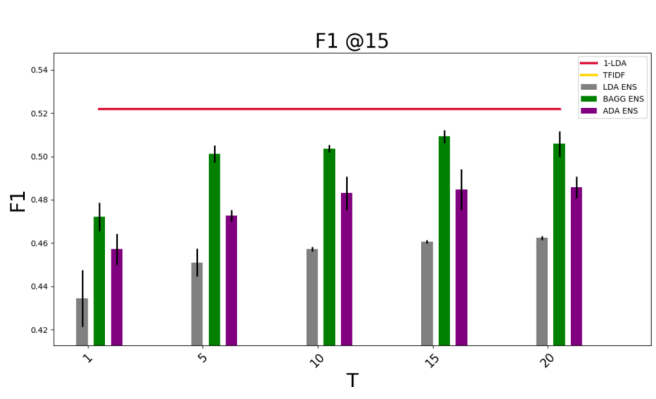
(A) MAP



(B) PRECISION

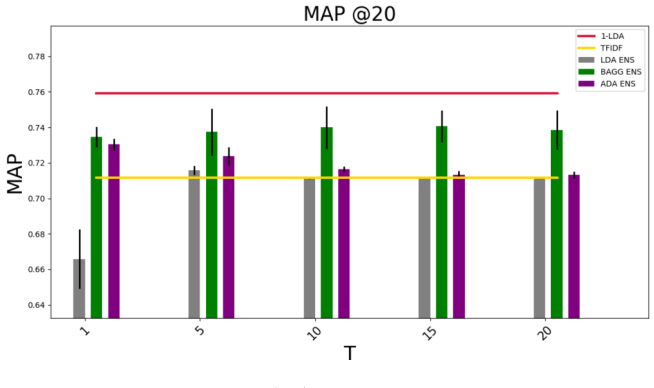


(C) RECALL

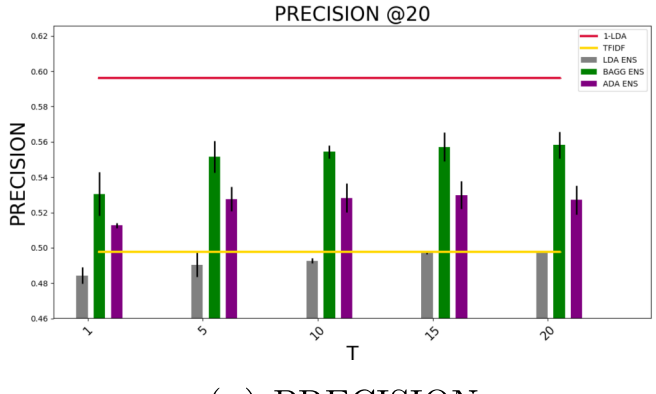


(D) F1

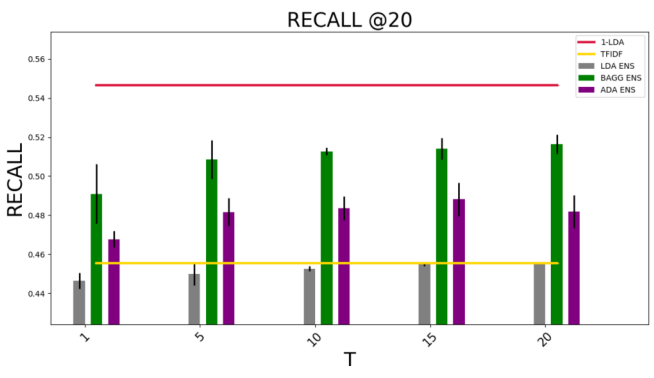
FIGURA A.21. MED @15



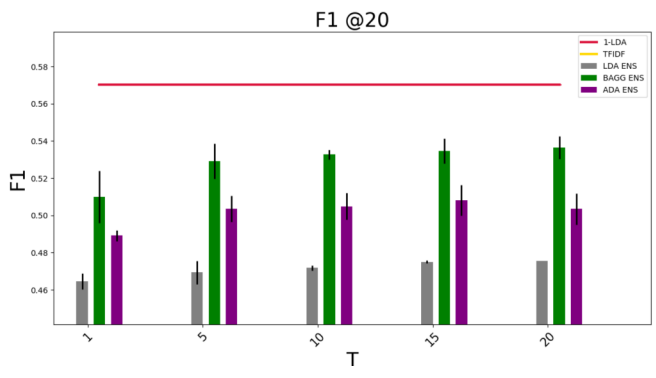
(A) MAP



(B) PRECISION



(C) RECALL



(D) F1

FIGURA A.22. MED @20

A.8. Tablas MED

A.8.1. MED MAP .

Metodo \T	1	5	10	15	20
1-LDA	0.86907 ± 0.0000	0.86907 ± 0.0000	0.86907 ± 0.0000	0.86907 ± 0.0000	0.86907 ± 0.0000
TFIDF	0.78977 ± 0.0000	0.78977 ± 0.0000	0.78977 ± 0.0000	0.78977 ± 0.0000	0.78977 ± 0.0000
LDA ENS	0.76022 ± 0.02078	0.7885 ± 0.00582	0.79056 ± 0.00111	0.78977 ± 0.0	0.78977 ± 0.0
BAGG ENS	0.84256 ± 0.01053	0.8687 ± 0.0423	0.8558 ± 0.01108	0.86966 ± 0.01478	0.86769 ± 0.01235
ADA ENS	0.85244 ± 0.02337	0.83526 ± 0.01302	0.81048 ± 0.0077	0.81907 ± 0.00202	0.8091 ± 0.00709
MED MAP P@5					
Metodo \T	1	5	10	15	20
1-LDA	0.80278 ± 0.0000	0.80278 ± 0.0000	0.80278 ± 0.0000	0.80278 ± 0.0000	0.80278 ± 0.0000
TFIDF	0.75679 ± 0.0000	0.75679 ± 0.0000	0.75679 ± 0.0000	0.75679 ± 0.0000	0.75679 ± 0.0000
LDA ENS	0.71888 ± 0.01086	0.76543 ± 0.00245	0.75702 ± 0.00185	0.75799 ± 0.00162	0.75761 ± 0.00115
BAGG ENS	0.80134 ± 0.00764	0.79886 ± 0.02494	0.80522 ± 0.01429	0.8057 ± 0.00688	0.80592 ± 0.00615
ADA ENS	0.79878 ± 0.01517	0.78014 ± 0.00455	0.77134 ± 0.00514	0.76671 ± 0.00432	0.76234 ± 0.00111
MED MAP P@10					
Metodo \T	1	5	10	15	20
1-LDA	0.7737 ± 0.0000	0.7737 ± 0.0000	0.7737 ± 0.0000	0.7737 ± 0.0000	0.7737 ± 0.0000
TFIDF	0.73051 ± 0.0000	0.73051 ± 0.0000	0.73051 ± 0.0000	0.73051 ± 0.0000	0.73051 ± 0.0000
LDA ENS	0.68987 ± 0.01462	0.73399 ± 0.0015	0.73104 ± 0.00103	0.7307 ± 8e-05	0.73059 ± 0.00012
BAGG ENS	0.76362 ± 0.00655	0.76287 ± 0.01627	0.76547 ± 0.01666	0.76544 ± 0.0118	0.76542 ± 0.00885
ADA ENS	0.75696 ± 0.01033	0.74627 ± 0.00292	0.74088 ± 0.00221	0.73892 ± 0.00393	0.73593 ± 0.00232
MED MAP P@15					
Metodo \T	1	5	10	15	20
1-LDA	0.7593 ± 0.0000	0.7593 ± 0.0000	0.7593 ± 0.0000	0.7593 ± 0.0000	0.7593 ± 0.0000
TFIDF	0.71162 ± 0.0000	0.71162 ± 0.0000	0.71162 ± 0.0000	0.71162 ± 0.0000	0.71162 ± 0.0000
LDA ENS	0.66581 ± 0.01673	0.71574 ± 0.00255	0.7112 ± 0.00033	0.71181 ± 0.00032	0.71156 ± 4e-05
BAGG ENS	0.73447 ± 0.00574	0.73742 ± 0.01317	0.73998 ± 0.01191	0.74075 ± 0.00895	0.7385 ± 0.01099
ADA ENS	0.73039 ± 0.00329	0.72368 ± 0.00517	0.71632 ± 0.0016	0.71338 ± 0.00211	0.71316 ± 0.00186
MED MAP P@20					

Figura A.23. med map

A.8.2. MED Precision .

Metodo \T	1	5	10	15	20
1-LDA	0.70667 ± 0.0000	0.70667 ± 0.0000	0.70667 ± 0.0000	0.70667 ± 0.0000	0.70667 ± 0.0000
TFIDF	0.70667 ± 0.0000	0.70667 ± 0.0000	0.70667 ± 0.0000	0.70667 ± 0.0000	0.70667 ± 0.0000
LDA ENS	0.64444 ± 0.00314	0.69111 ± 0.00629	0.70444 ± 0.00314	0.70667 ± 0.0	0.70667 ± 0.0
BAGG ENS	0.72889 ± 0.01912	0.72667 ± 0.01963	0.75333 ± 0.0	0.76222 ± 0.00831	0.75111 ± 0.00831
ADA ENS	0.71556 ± 0.01257	0.72667 ± 0.02373	0.71556 ± 0.00831	0.72222 ± 0.01663	0.71556 ± 0.01133

MED Precision@5

Metodo \T	1	5	10	15	20
1-LDA	0.68 ± 0.0000	0.68 ± 0.0000	0.68 ± 0.0000	0.68 ± 0.0000	0.68 ± 0.0000
TFIDF	0.61 ± 0.0000	0.61 ± 0.0000	0.61 ± 0.0000	0.61 ± 0.0000	0.61 ± 0.0000
LDA ENS	0.56556 ± 0.02043	0.59444 ± 0.00629	0.60444 ± 0.00157	0.60444 ± 0.00314	0.60778 ± 0.00314
BAGG ENS	0.63333 ± 0.00471	0.66222 ± 0.00685	0.66 ± 0.0144	0.66556 ± 0.01397	0.65889 ± 0.00875
ADA ENS	0.60556 ± 0.00416	0.61889 ± 0.00567	0.62778 ± 0.00685	0.63222 ± 0.00956	0.63667 ± 0.01186

MED Precision@10

Metodo \T	1	5	10	15	20
1-LDA	0.63214 ± 0.0000	0.63214 ± 0.0000	0.63214 ± 0.0000	0.63214 ± 0.0000	0.63214 ± 0.0000
TFIDF	0.56103 ± 0.0000	0.56103 ± 0.0000	0.56103 ± 0.0000	0.56103 ± 0.0000	0.56103 ± 0.0000
LDA ENS	0.52325 ± 0.01571	0.54251 ± 0.00818	0.55214 ± 0.00181	0.55806 ± 0.00105	0.56028 ± 0.00105
BAGG ENS	0.5714 ± 0.00733	0.60769 ± 0.00544	0.60917 ± 0.00277	0.61658 ± 0.00544	0.61288 ± 0.00838
ADA ENS	0.55806 ± 0.00733	0.57362 ± 0.00277	0.58473 ± 0.00895	0.58695 ± 0.01048	0.58843 ± 0.00583

MED Precision@15

Metodo \T	1	5	10	15	20
1-LDA	0.59603 ± 0.0000	0.59603 ± 0.0000	0.59603 ± 0.0000	0.59603 ± 0.0000	0.59603 ± 0.0000
TFIDF	0.49769 ± 0.0000	0.49769 ± 0.0000	0.49769 ± 0.0000	0.49769 ± 0.0000	0.49769 ± 0.0000
LDA ENS	0.48436 ± 0.00471	0.49047 ± 0.00685	0.49269 ± 0.00136	0.49714 ± 0.00079	0.49769 ± 0.0
BAGG ENS	0.53047 ± 0.01235	0.55158 ± 0.00885	0.55436 ± 0.0036	0.55714 ± 0.0082	0.55825 ± 0.00749
ADA ENS	0.51269 ± 0.00136	0.52769 ± 0.0068	0.52825 ± 0.0082	0.52991 ± 0.00797	0.52714 ± 0.0082

MED Precision@20

FIGURA A.24. MED PRECISION

A.8.3. MED Recall .

Metodo \T	1	5	10	15	20
1-LDA	0.17196 ± 0.0000	0.17196 ± 0.0000	0.17196 ± 0.0000	0.17196 ± 0.0000	0.17196 ± 0.0000
TFIDF	0.17506 ± 0.0000	0.17506 ± 0.0000	0.17506 ± 0.0000	0.17506 ± 0.0000	0.17506 ± 0.0000
LDA ENS	0.15652 ± 0.00043	0.17321 ± 0.00114	0.17457 ± 0.0007	0.17506 ± 0.0	0.17506 ± 0.0
BAGG ENS	0.17814 ± 0.0074	0.1807 ± 0.00491	0.18729 ± 0.00183	0.1894 ± 0.00191	0.18653 ± 0.002
ADA ENS	0.17497 ± 0.00243	0.17824 ± 0.00477	0.17676 ± 0.00198	0.17909 ± 0.00367	0.17847 ± 0.00284
MED Recall@5					
Metodo \T	1	5	10	15	20
1-LDA	0.32417 ± 0.0000	0.32417 ± 0.0000	0.32417 ± 0.0000	0.32417 ± 0.0000	0.32417 ± 0.0000
TFIDF	0.29165 ± 0.0000	0.29165 ± 0.0000	0.29165 ± 0.0000	0.29165 ± 0.0000	0.29165 ± 0.0000
LDA ENS	0.27197 ± 0.00935	0.28601 ± 0.00263	0.28908 ± 0.00123	0.28888 ± 0.00175	0.29024 ± 0.002
BAGG ENS	0.30135 ± 0.00083	0.31824 ± 0.00161	0.31634 ± 0.00654	0.3195 ± 0.00687	0.31598 ± 0.00543
ADA ENS	0.28711 ± 0.00116	0.29524 ± 0.00292	0.3023 ± 0.00394	0.30534 ± 0.00423	0.30745 ± 0.00687
MED Recall@10					
1-LDA	0.44427 ± 0.0000	0.44427 ± 0.0000	0.44427 ± 0.0000	0.44427 ± 0.0000	0.44427 ± 0.0000
TFIDF	0.39412 ± 0.0000	0.39412 ± 0.0000	0.39412 ± 0.0000	0.39412 ± 0.0000	0.39412 ± 0.0000
LDA ENS	0.37132 ± 0.01124	0.38593 ± 0.00534	0.39006 ± 0.00062	0.3921 ± 0.0006	0.39359 ± 0.00075
BAGG ENS	0.40226 ± 0.00572	0.42642 ± 0.00307	0.42931 ± 0.00166	0.43377 ± 0.00172	0.4306 ± 0.00446
ADA ENS	0.38713 ± 0.00672	0.4019 ± 0.00257	0.41145 ± 0.00676	0.41274 ± 0.0086	0.4137 ± 0.00448
MED Recall@15					
Metodo \T	1	5	10	15	20
1-LDA	0.54664 ± 0.0000	0.54664 ± 0.0000	0.54664 ± 0.0000	0.54664 ± 0.0000	0.54664 ± 0.0000
TFIDF	0.45549 ± 0.0000	0.45549 ± 0.0000	0.45549 ± 0.0000	0.45549 ± 0.0000	0.45549 ± 0.0000
LDA ENS	0.44639 ± 0.00399	0.44999 ± 0.00582	0.45259 ± 0.00132	0.45487 ± 0.00087	0.45549 ± 0.0
BAGG ENS	0.49097 ± 0.0153	0.50848 ± 0.0099	0.51272 ± 0.00197	0.51409 ± 0.00542	0.51636 ± 0.00505
ADA ENS	0.46775 ± 0.00431	0.48162 ± 0.0071	0.48356 ± 0.0061	0.48815 ± 0.0086	0.48189 ± 0.00848
MED Recall@20					

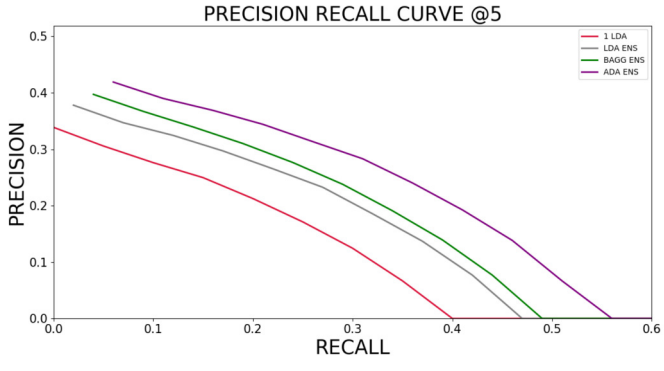
FIGURA A.25. MED RECALL

A.8.4. MED F1 .

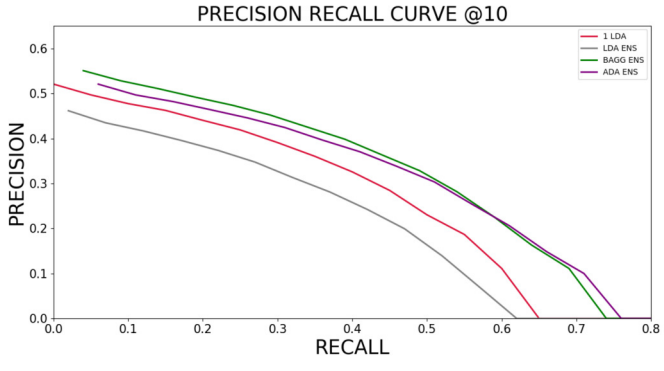
Metodo \T	1	5	10	15	20
1-LDA	0.27661 ± 0.0000	0.27661 ± 0.0000	0.27661 ± 0.0000	0.27661 ± 0.0000	0.27661 ± 0.0000
TFIDF	0.19697 ± 0.0000	0.19697 ± 0.0000	0.19697 ± 0.0000	0.19697 ± 0.0000	0.19697 ± 0.0000
LDA ENS	0.25186 ± 0.0008	0.27699 ± 0.00181	0.2798 ± 0.00115	0.28061 ± 0.0	0.28061 ± 0.0
BAGG ENS	0.28629 ± 0.011	0.28943 ± 0.00784	0.29999 ± 0.00235	0.30341 ± 0.00311	0.29884 ± 0.00314
ADA ENS	0.28118 ± 0.00396	0.28626 ± 0.00799	0.28349 ± 0.00319	0.287 ± 0.00603	0.28568 ± 0.00447
MED F1@5					
1-LDA	0.43904 ± 0.0000	0.43904 ± 0.0000	0.43904 ± 0.0000	0.43904 ± 0.0000	0.43904 ± 0.0000
TFIDF	0.27872 ± 0.0000	0.27872 ± 0.0000	0.27872 ± 0.0000	0.27872 ± 0.0000	0.27872 ± 0.0000
LDA ENS	0.36731 ± 0.01283	0.3862 ± 0.00372	0.3911 ± 0.00145	0.39092 ± 0.00223	0.39287 ± 0.00248
BAGG ENS	0.40838 ± 0.00158	0.42989 ± 0.00291	0.42769 ± 0.00899	0.43174 ± 0.00919	0.42713 ± 0.00678
ADA ENS	0.38953 ± 0.00174	0.39977 ± 0.00369	0.40809 ± 0.00497	0.4118 ± 0.00587	0.41466 ± 0.00876
MED F1@10					
Metodo \T	1	5	10	15	20
1-LDA	0.52181 ± 0.0000	0.52181 ± 0.0000	0.52181 ± 0.0000	0.52181 ± 0.0000	0.52181 ± 0.0000
TFIDF	0.34057 ± 0.0000	0.34057 ± 0.0000	0.34057 ± 0.0000	0.34057 ± 0.0000	0.34057 ± 0.0000
LDA ENS	0.43438 ± 0.0131	0.45102 ± 0.00645	0.45716 ± 0.00105	0.46059 ± 0.00077	0.46237 ± 0.00087
BAGG ENS	0.47214 ± 0.00644	0.50117 ± 0.00397	0.50366 ± 0.00163	0.50926 ± 0.00303	0.50581 ± 0.00587
ADA ENS	0.45714 ± 0.00715	0.47264 ± 0.00267	0.48302 ± 0.00762	0.48467 ± 0.0095	0.48583 ± 0.005
MED F1@15					
Metodo \T	1	5	10	15	20
1-LDA	0.57027 ± 0.0000	0.57027 ± 0.0000	0.57027 ± 0.0000	0.57027 ± 0.0000	0.57027 ± 0.0000
TFIDF	0.37299 ± 0.0000	0.37299 ± 0.0000	0.37299 ± 0.0000	0.37299 ± 0.0000	0.37299 ± 0.0000
LDA ENS	0.4646 ± 0.00433	0.46936 ± 0.00626	0.47179 ± 0.00132	0.47507 ± 0.00083	0.47566 ± 0.0
BAGG ENS	0.50995 ± 0.01394	0.52915 ± 0.00936	0.53272 ± 0.00253	0.53474 ± 0.00668	0.53649 ± 0.00618
ADA ENS	0.48919 ± 0.00292	0.5036 ± 0.00694	0.50491 ± 0.00706	0.50817 ± 0.00829	0.5035 ± 0.00834
MED F1@20					

Figura A.26. MED F1

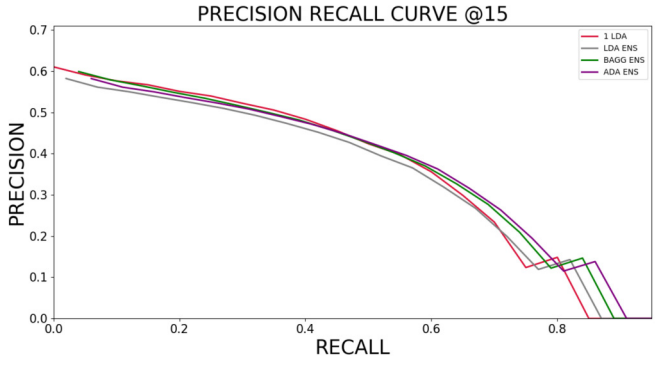
A.9. Curvas Precision Recall MED



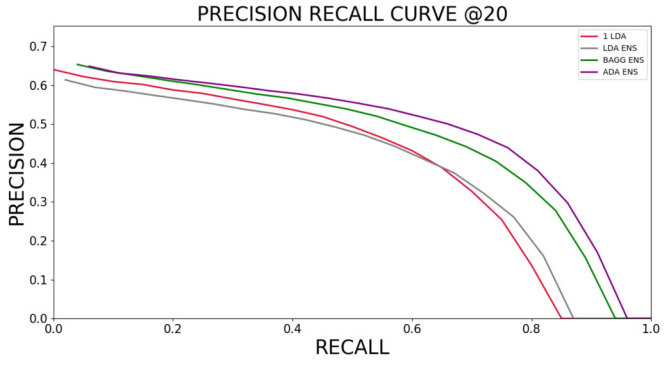
(A) CURVA MED @5



(B) CURVA MED @10



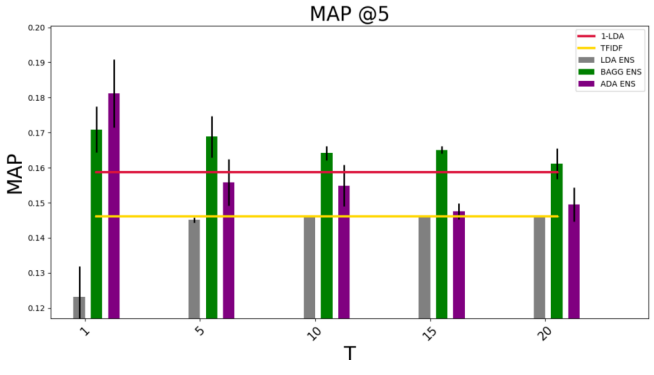
(C) CURVA MED @15



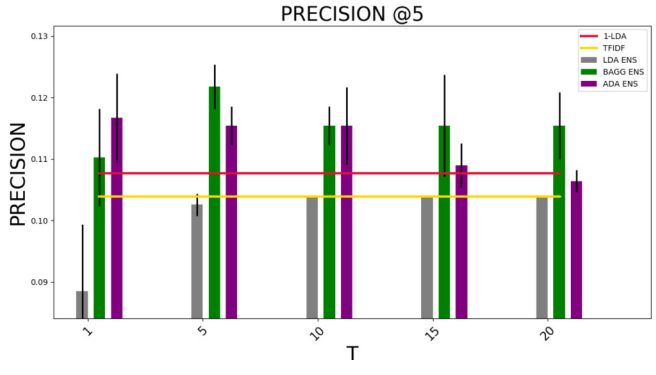
(D) CURVA MED @20

FIGURA A.27. CURVAS PRECISION RECALL MED

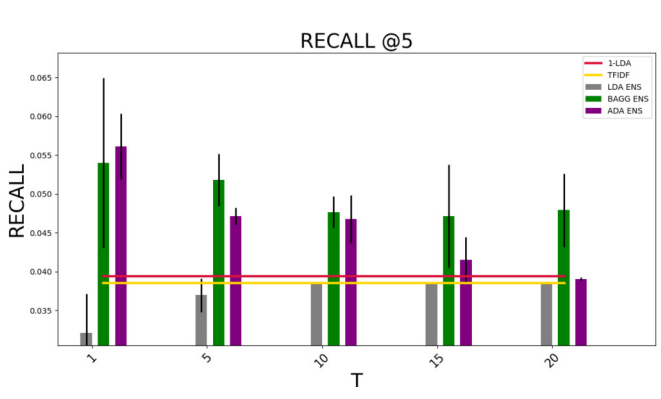
A.10. IMÁGENES CACM



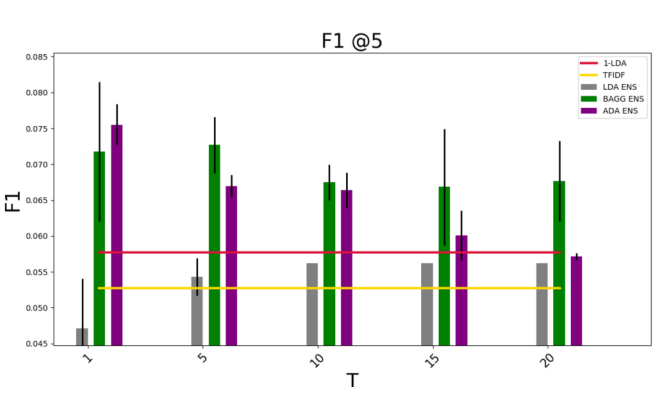
(A) MAP



(B) PRECISION

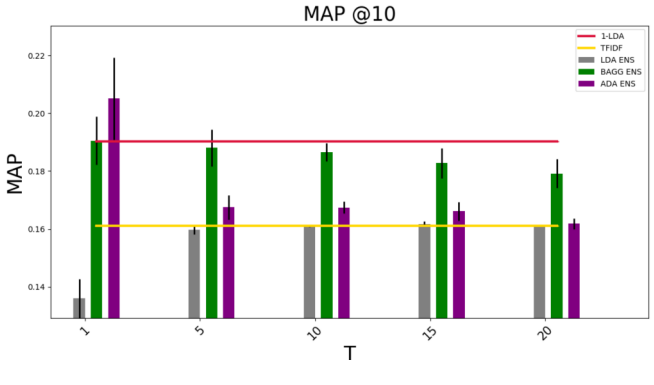


(C) RECALL

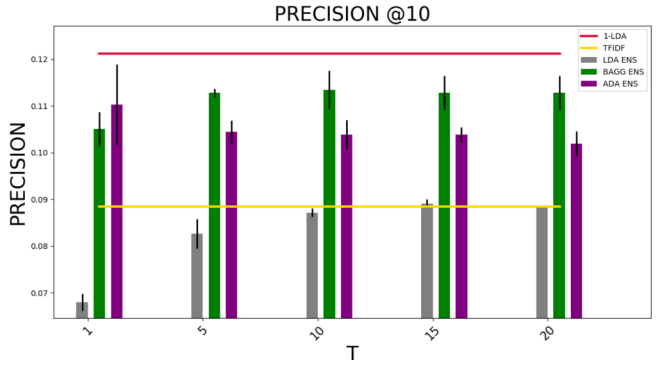


(D) F1

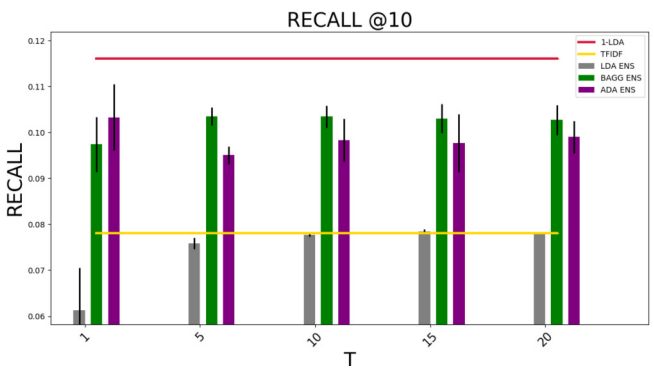
FIGURA A.28. CACM @5



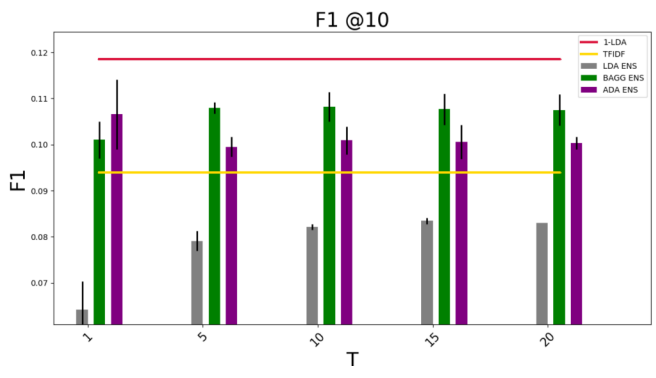
(A) MAP



(B) PRECISION

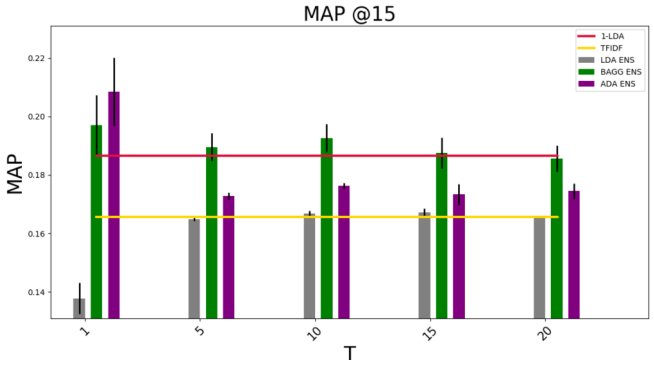


(C) RECALL

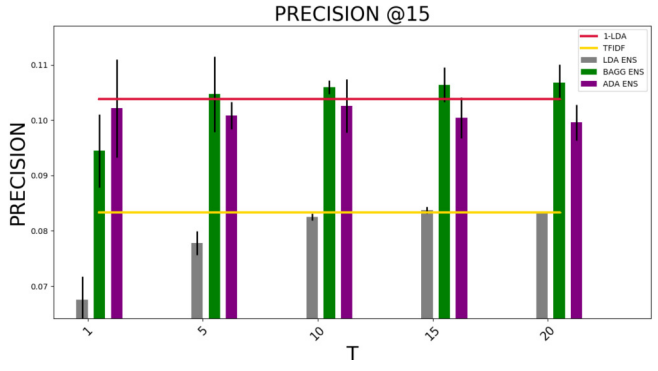


(D) F1

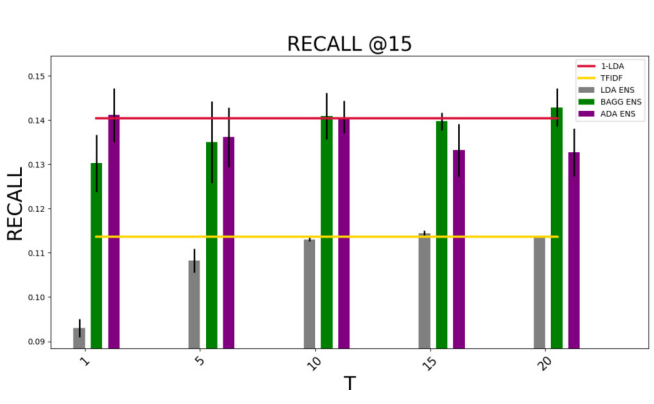
FIGURA A.29. CACM @10



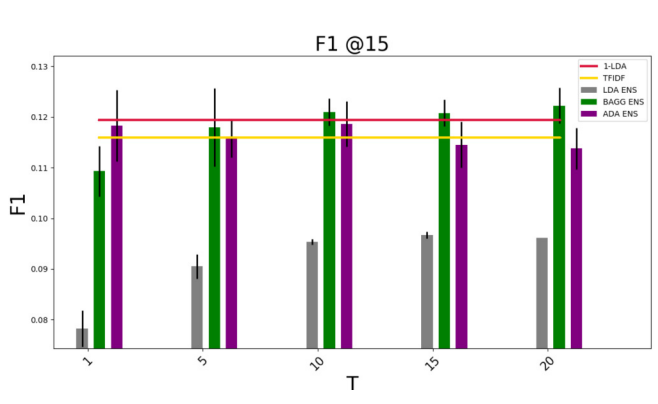
(A) MAP



(B) PRECISION

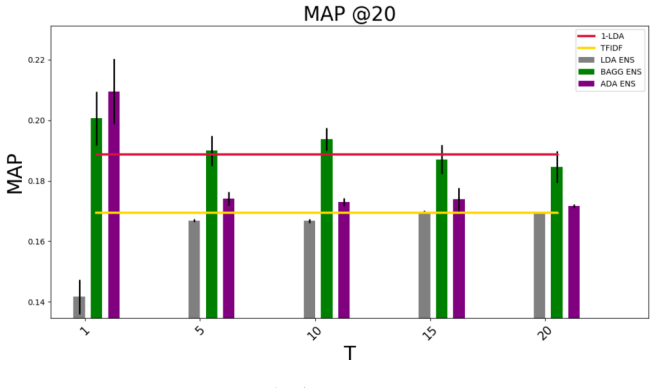


(C) RECALL

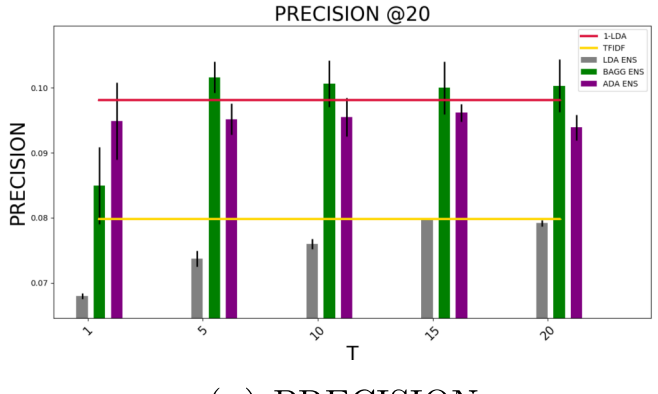


(D) F1

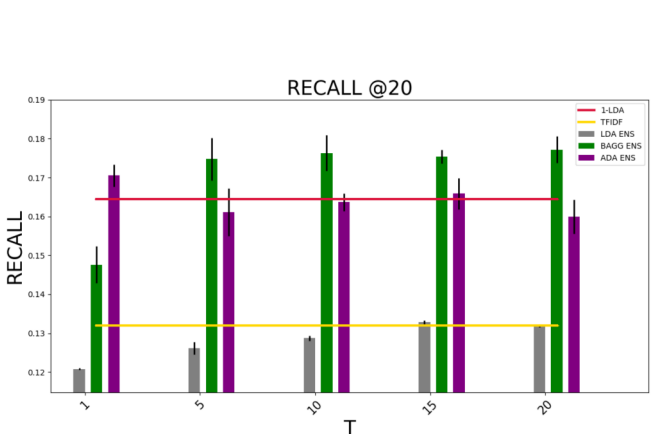
FIGURA A.30. CACM @15



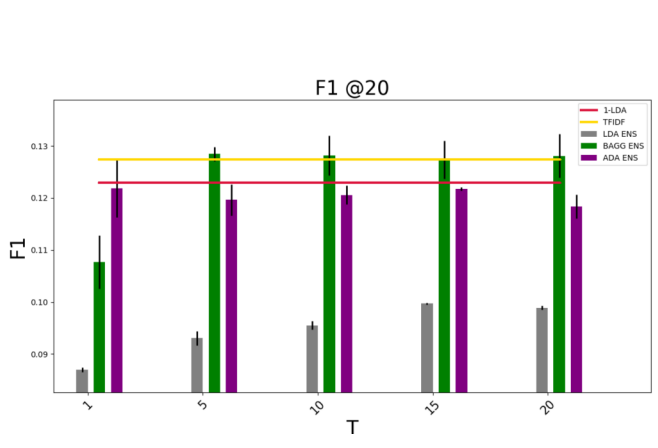
(A) MAP



(B) PRECISION



(C) RECALL



(D) F1

FIGURA A.31. CACM @20

A.11. Tablas CACM

A.11.1. CACM MAP .

1-LDA	0.15865 ± 0.0000	0.15865 ± 0.0000	0.15865 ± 0.0000	0.15865 ± 0.0000	0.15865 ± 0.0000
TFIDF	0.14615 ± 0.0000	0.14615 ± 0.0000	0.14615 ± 0.0000	0.14615 ± 0.0000	0.14615 ± 0.0000
LDA ENS	0.12318 ± 0.00869	0.14509 ± 0.00076	0.14615 ± 0.0000	0.14615 ± 0.0000	0.14615 ± 0.0
BAGG ENS	0.17089 ± 0.00652	0.16885 ± 0.00594	0.16418 ± 0.00203	0.16506 ± 0.00102	0.16116 ± 0.00441
ADA ENS	0.1812 ± 0.00974	0.15577 ± 0.00662	0.15493 ± 0.00591	0.1476 ± 0.00227	0.14952 ± 0.00487
CACM ME MAP P@5					
1-LDA	0.19026 ± 0.0000	0.19026 ± 0.0000	0.19026 ± 0.0000	0.19026 ± 0.0000	0.19026 ± 0.0000
TFIDF	0.16111 ± 0.0000	0.16111 ± 0.0000	0.16111 ± 0.0000	0.16111 ± 0.0000	0.16111 ± 0.0000
LDA ENS	0.13595 ± 0.00669	0.15969 ± 0.00154	0.161 ± 0.0002	0.16176 ± 0.00091	0.16111 ± 0.0
BAGG ENS	0.19057 ± 0.00828	0.18803 ± 0.00643	0.18653 ± 0.0031	0.18278 ± 0.00517	0.17916 ± 0.00496
ADA ENS	0.20515 ± 0.0142	0.16748 ± 0.00427	0.16743 ± 0.0021	0.16614 ± 0.00325	0.16181 ± 0.00184
CACM ME MAP P@10					
1-LDA	0.18666 ± 0.0000	0.18666 ± 0.0000	0.18666 ± 0.0000	0.18666 ± 0.0000	0.18666 ± 0.0000
TFIDF	0.16567 ± 0.0000	0.16567 ± 0.0000	0.16567 ± 0.0000	0.16567 ± 0.0000	0.16567 ± 0.0000
LDA ENS	0.13783 ± 0.00533	0.16487 ± 0.00062	0.16676 ± 0.00096	0.1672 ± 0.00128	0.16567 ± 0.0
BAGG ENS	0.19713 ± 0.01024	0.18953 ± 0.00475	0.19268 ± 0.00469	0.18751 ± 0.00522	0.18564 ± 0.0045
ADA ENS	0.20842 ± 0.01162	0.1728 ± 0.00118	0.17631 ± 0.00101	0.17336 ± 0.00358	0.17449 ± 0.00254
CACM ME MAP P@15					
1-LDA	0.1887 ± 0.0000	0.1887 ± 0.0000	0.1887 ± 0.0000	0.1887 ± 0.0000	0.1887 ± 0.0000
TFIDF	0.1695 ± 0.0000	0.1695 ± 0.0000	0.1695 ± 0.0000	0.1695 ± 0.0000	0.1695 ± 0.0000
LDA ENS	0.14168 ± 0.00576	0.1669 ± 0.00043	0.16673 ± 0.00062	0.16987 ± 0.00026	0.16962 ± 9e-05
BAGG ENS	0.2006 ± 0.00882	0.18995 ± 0.00499	0.19375 ± 0.00367	0.18709 ± 0.00474	0.18457 ± 0.00528
ADA ENS	0.20953 ± 0.01072	0.1741 ± 0.00218	0.17293 ± 0.0013	0.17387 ± 0.00381	0.17175 ± 0.00056
CACM ME MAP P@20					

Figura A.32. CACM MAP

A.11.2. CACM Precision .

1-LDA	0.10769 ± 0.0000	0.10769 ± 0.0000	0.10769 ± 0.0000	0.10769 ± 0.0000	0.10769 ± 0.0000
TFIDF	0.10385 ± 0.0000	0.10385 ± 0.0000	0.10385 ± 0.0000	0.10385 ± 0.0000	0.10385 ± 0.0000
LDA ENS	0.08846 ± 0.01088	0.10256 ± 0.00181	0.10385 ± 0.0	0.10385 ± 0.0	0.10385 ± 0.0
BAGG ENS	0.11026 ± 0.0079	0.12179 ± 0.00363	0.11538 ± 0.00314	0.11538 ± 0.00831	0.11538 ± 0.00544
ADA ENS	0.11667 ± 0.00725	0.11538 ± 0.00314	0.11538 ± 0.00628	0.10897 ± 0.00363	0.10641 ± 0.00181
CACM Precision@5					
1-LDA	0.12115 ± 0.0000	0.12115 ± 0.0000	0.12115 ± 0.0000	0.12115 ± 0.0000	0.12115 ± 0.0000
TFIDF	0.08846 ± 0.0000	0.08846 ± 0.0000	0.08846 ± 0.0000	0.08846 ± 0.0000	0.08846 ± 0.0000
LDA ENS	0.06795 ± 0.00181	0.08269 ± 0.00314	0.08718 ± 0.00091	0.0891 ± 0.00091	0.08846 ± 0.0
BAGG ENS	0.10513 ± 0.00363	0.11282 ± 0.00091	0.11346 ± 0.00415	0.11282 ± 0.00363	0.11282 ± 0.00363
ADA ENS	0.11026 ± 0.00865	0.10449 ± 0.0024	0.10385 ± 0.00314	0.10385 ± 0.00157	0.10192 ± 0.00272
CACM Precision@10					
1-LDA	0.10385 ± 0.0000	0.10385 ± 0.0000	0.10385 ± 0.0000	0.10385 ± 0.0000	0.10385 ± 0.0000
TFIDF	0.08333 ± 0.0000	0.08333 ± 0.0000	0.08333 ± 0.0000	0.08333 ± 0.0000	0.08333 ± 0.0000
LDA ENS	0.06752 ± 0.00423	0.07778 ± 0.00218	0.08248 ± 0.0006	0.08376 ± 0.0006	0.08333 ± 0.0
BAGG ENS	0.09444 ± 0.00665	0.1047 ± 0.00681	0.10598 ± 0.00121	0.10641 ± 0.00314	0.10684 ± 0.0032
ADA ENS	0.10214 ± 0.0089	0.10085 ± 0.00242	0.10256 ± 0.0048	0.10043 ± 0.00368	0.09957 ± 0.0032
CACM Precision@15					
1-LDA	0.09808 ± 0.0000	0.09808 ± 0.0000	0.09808 ± 0.0000	0.09808 ± 0.0000	0.09808 ± 0.0000
TFIDF	0.07981 ± 0.0000	0.07981 ± 0.0000	0.07981 ± 0.0000	0.07981 ± 0.0000	0.07981 ± 0.0000
LDA ENS	0.06795 ± 0.00045	0.07372 ± 0.0012	0.07596 ± 0.00079	0.07981 ± 0.0	0.07917 ± 0.00045
BAGG ENS	0.08494 ± 0.00594	0.1016 ± 0.0024	0.10064 ± 0.00354	0.1 ± 0.00408	0.10032 ± 0.00403
ADA ENS	0.09487 ± 0.00594	0.09519 ± 0.00236	0.09551 ± 0.00297	0.09615 ± 0.00136	0.09391 ± 0.00198
CACM Precision@20					

FIGURA A.33. CACM PRECISION

A.11.3. CACM Recall .

1-LDA	0.03943 ± 0.0000	0.03943 ± 0.0000	0.03943 ± 0.0000	0.03943 ± 0.0000	0.03943 ± 0.0000
TFIDF	0.03849 ± 0.0000	0.03849 ± 0.0000	0.03849 ± 0.0000	0.03849 ± 0.0000	0.03849 ± 0.0000
LDA ENS	0.03208 ± 0.00506	0.03695 ± 0.00218	0.03849 ± 0.0	0.03849 ± 0.0	0.03849 ± 0.0
BAGG ENS	0.05398 ± 0.01097	0.05181 ± 0.00333	0.04767 ± 0.00198	0.04712 ± 0.00663	0.0479 ± 0.0047
ADA ENS	0.05611 ± 0.00422	0.04713 ± 0.0011	0.04677 ± 0.00304	0.04149 ± 0.00295	0.03903 ± 0.00021
CACM Recall@5					
1-LDA	0.11611 ± 0.0000	0.11611 ± 0.0000	0.11611 ± 0.0000	0.11611 ± 0.0000	0.11611 ± 0.0000
TFIDF	0.07813 ± 0.0000	0.07813 ± 0.0000	0.07813 ± 0.0000	0.07813 ± 0.0000	0.07813 ± 0.0000
LDA ENS	0.06125 ± 0.00927	0.07584 ± 0.00122	0.07767 ± 0.00032	0.07846 ± 0.00048	0.07813 ± 0.0
BAGG ENS	0.09741 ± 0.006	0.10345 ± 0.00197	0.10346 ± 0.00242	0.10301 ± 0.00313	0.10268 ± 0.00324
ADA ENS	0.10327 ± 0.00718	0.09503 ± 0.00193	0.09825 ± 0.00469	0.09769 ± 0.00627	0.09899 ± 0.0035
CACM Recall@10					
1-LDA	0.14041 ± 0.0000	0.14041 ± 0.0000	0.14041 ± 0.0000	0.14041 ± 0.0000	0.14041 ± 0.0000
TFIDF	0.11366 ± 0.0000	0.11366 ± 0.0000	0.11366 ± 0.0000	0.11366 ± 0.0000	0.11366 ± 0.0000
LDA ENS	0.09303 ± 0.00204	0.10829 ± 0.00266	0.11302 ± 0.00045	0.11441 ± 0.00066	0.11366 ± 0.0
BAGG ENS	0.13028 ± 0.00647	0.13507 ± 0.00924	0.14101 ± 0.00525	0.13974 ± 0.00201	0.14292 ± 0.00428
ADA ENS	0.14115 ± 0.00608	0.13616 ± 0.0067	0.1407 ± 0.00365	0.13319 ± 0.00592	0.13275 ± 0.00539
CACM Recall@15					
1-LDA	0.16451 ± 0.0000	0.16451 ± 0.0000	0.16451 ± 0.0000	0.16451 ± 0.0000	0.16451 ± 0.0000
TFIDF	0.13204 ± 0.0000	0.13204 ± 0.0000	0.13204 ± 0.0000	0.13204 ± 0.0000	0.13204 ± 0.0000
LDA ENS	0.12081 ± 0.00022	0.12616 ± 0.00161	0.12872 ± 0.00069	0.13281 ± 0.00055	0.13174 ± 0.00021
BAGG ENS	0.14763 ± 0.00472	0.17477 ± 0.00544	0.17634 ± 0.00464	0.17543 ± 0.00176	0.17718 ± 0.0034
ADA ENS	0.17053 ± 0.0028	0.16108 ± 0.00615	0.16371 ± 0.00226	0.16587 ± 0.00395	0.15993 ± 0.00441
CACM Recall@20					

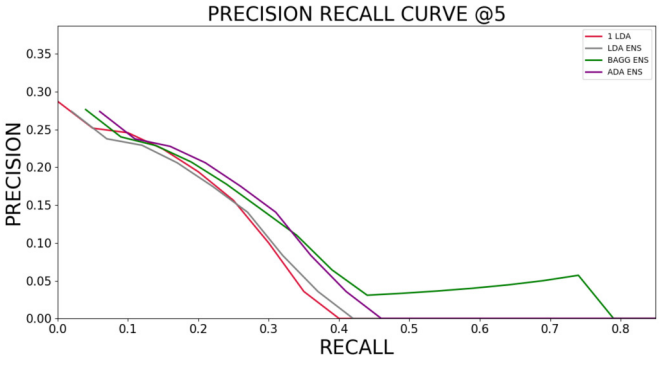
Figura A.34. cacm recall

A.11.4. CACM F1 .

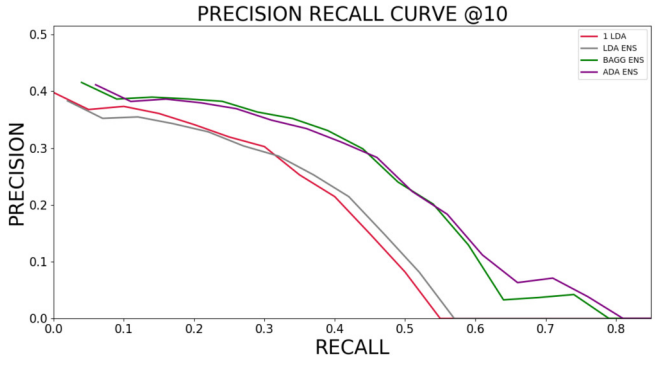
1-LDA	0.05772 ± 0.0000	0.05772 ± 0.0000	0.05772 ± 0.0000	0.05772 ± 0.0000	0.05772 ± 0.0000
TFIDF	0.05274 ± 0.0000	0.05274 ± 0.0000	0.05274 ± 0.0000	0.05274 ± 0.0000	0.05274 ± 0.0000
LDA ENS	0.04707 ± 0.007	0.05431 ± 0.00262	0.05617 ± 0.0	0.05617 ± 0.0	0.05617 ± 0.0
BAGG ENS	0.07175 ± 0.00972	0.07267 ± 0.00388	0.06746 ± 0.0025	0.06684 ± 0.00809	0.06766 ± 0.00559
ADA ENS	0.07551 ± 0.00286	0.06693 ± 0.0016	0.06638 ± 0.00247	0.06006 ± 0.00346	0.05711 ± 0.00048
CACM F1@5					
1-LDA	0.11858 ± 0.0000	0.11858 ± 0.0000	0.11858 ± 0.0000	0.11858 ± 0.0000	0.11858 ± 0.0000
TFIDF	0.09397 ± 0.0000	0.09397 ± 0.0000	0.09397 ± 0.0000	0.09397 ± 0.0000	0.09397 ± 0.0000
LDA ENS	0.06416 ± 0.00616	0.07911 ± 0.0021	0.08215 ± 0.00058	0.08345 ± 0.00067	0.08297 ± 0.0
BAGG ENS	0.10103 ± 0.00402	0.10792 ± 0.00121	0.10822 ± 0.00317	0.10769 ± 0.00336	0.10751 ± 0.00341
ADA ENS	0.10659 ± 0.00755	0.09954 ± 0.00215	0.1009 ± 0.00302	0.10058 ± 0.00373	0.10035 ± 0.00131
CACM F1@10					
1-LDA	0.11939 ± 0.0000	0.11939 ± 0.0000	0.11939 ± 0.0000	0.11939 ± 0.0000	0.11939 ± 0.0000
TFIDF	0.11597 ± 0.0000	0.11597 ± 0.0000	0.11597 ± 0.0000	0.11597 ± 0.0000	0.11597 ± 0.0000
LDA ENS	0.07822 ± 0.00359	0.09053 ± 0.0024	0.09536 ± 0.00057	0.09671 ± 0.00063	0.09616 ± 0.0
BAGG ENS	0.10932 ± 0.00497	0.11795 ± 0.00772	0.12099 ± 0.00271	0.12081 ± 0.00266	0.12227 ± 0.00356
ADA ENS	0.11831 ± 0.00709	0.11585 ± 0.00381	0.11862 ± 0.00446	0.11451 ± 0.00454	0.11379 ± 0.00405
CACM F1@15					
1-LDA	0.12289 ± 0.0000	0.12289 ± 0.0000	0.12289 ± 0.0000	0.12289 ± 0.0000	0.12289 ± 0.0000
TFIDF	0.12736 ± 0.0000	0.12736 ± 0.0000	0.12736 ± 0.0000	0.12736 ± 0.0000	0.12736 ± 0.0000
LDA ENS	0.08698 ± 0.00043	0.09306 ± 0.00138	0.09554 ± 0.00081	0.0997 ± 0.00016	0.0989 ± 0.00041
BAGG ENS	0.10771 ± 0.00513	0.12843 ± 0.0014	0.12813 ± 0.00383	0.12735 ± 0.00361	0.12809 ± 0.00417
ADA ENS	0.12185 ± 0.00557	0.11963 ± 0.00302	0.12058 ± 0.00182	0.1217 ± 0.00031	0.11831 ± 0.00229
CACM F1@20					

FIGURA A.35. CACM F1

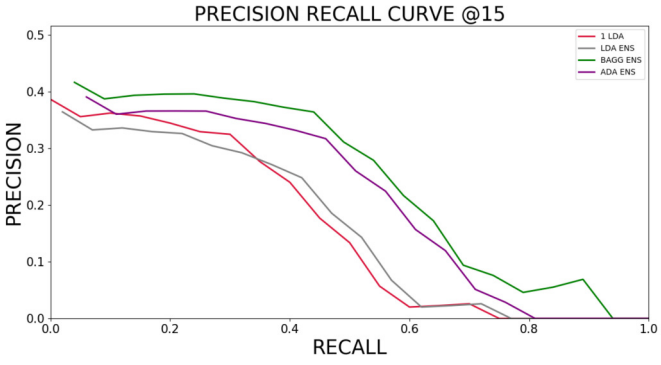
A.12. CURVAS PRECISION RECALL CACM



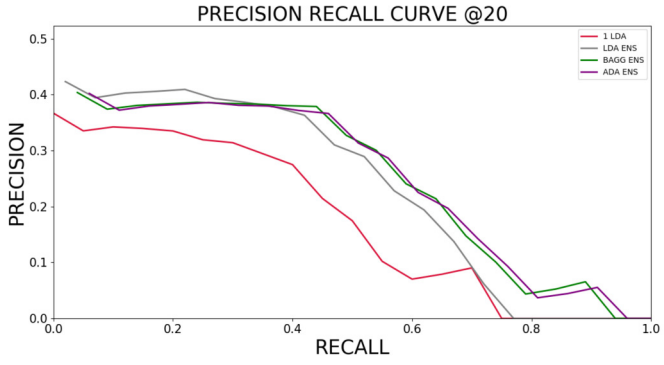
(A) CURVA CACM @5



(B) CURVA CACM @10



(C) CURVA CACM @15



(D) CURVA CACM @20

FIGURA A.36. CURVAS PRECISION RECALL CACM

Bibliografía

[ACD12] Eleftheria Ahtaridis, Christopher Cieri, and Denise DiPersio, *LDC language resource database: Building a bibliographic database*, Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12) (Istanbul, Turkey), European Language Resources Association (ELRA), May 2012, pp. 1723–1728.

[AGv04] L. Azzopardi, M. Girolami, and C. J. van Rijsbergen, *Topic based language models for ad hoc information retrieval*, 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), vol. 4, July 2004, pp. 3281–3286 vol.4.

[AKV16] Monika Arora, Uma Kanjilal, and Dinesh Varshney, *Evaluation of information retrieval: precision and recall*, International Journal of Indian Culture and Business Management **12** (2016), 224.

[Ali96] Kamal Ali, *On the link between error correlation and error reduction in decision tree ensembles.*

[Ama09] Giambattista Amati, *Divergence from randomness models.*

[ASAN15] Bassam Al-Salemi, Mohd Juzaiddin Ab Aziz, and Shahrul Azman Noah, *Lda-adaboost.mh: Accelerated adaboost.mh based on latent dirichlet allocation for text categorization*, Journal of Information Science **41** (2015), no. 1, 27–40 (English).

[ASANA17] B. Al-Salemi, M. Ayob, S. A. M. Noah, and M. J. A. Aziz, *Feature selection based on supervised topic modeling for boosting-based multi-label text categorization*, 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI), Nov 2017, pp. 1–6.

[BB96] Leo Breiman and Leo Breiman, *Bagging predictors*, Machine Learning, 1996, pp. 123–140.

[BB01] P. Baldi and S. Brunak, *Probabilistic graphical models in bioinformatics*, pp. 225–263, 2001.

[BB17] Bhagyashree Barde and Anant Bainwad, *An overview of topic modeling methods and tools*, 06 2017, pp. 745–750.

[BBM16] Stuart Blair, Yaxin Bi, and Maurice Mulvenna, *Increasing topic coherence by aggregating topic models*, vol. 9983, 10 2016, pp. 69–81.

[BBM20] Stuart J. Blair, Yaxin Bi, and Maurice D. Mulvenna, *Aggregated topic models for increasing social media topic coherence*, Applied Intelligence **50** (2020), no. 1, 138–156.

[BCB94] Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew, *Automatic combination of multiple ranked retrieval systems*, Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Berlin, Heidelberg), SIGIR 94, Springer Verlag, 1994, p. 173 181.

[BCC10] Stefan Buttcher, Charles Clarke, and Gordon V. Cormack, *Information retrieval: Implementing and evaluating search engines*, The MIT Press, 2010.

[BHA+19] Christopher Baechle, C. Huang, Ankur Agarwal, Ravi Behara, and Jahyun Goo, *Latent topic ensemble learning for hospital readmission cost optimization*, European Journal of Operational Research **281** (2019).

[Bi12] Yaxin Bi, *The impact of diversity on the accuracy of evidential classifier ensembles*, International Journal of Approximate Reasoning **53** (2012), 584 607.

[BKFS95] N.J. Belkin, P. Kantor, E.A. Fox, and J.A. Shaw, *Combining the evidence of multiple query representations for information retrieval*, Information Processing and Management **31** (1995), no. 3, 431 448, The Second Text Retrieval Conference (TREC-2).

[Ble11] David Blei, *Probabilistic topic models*, vol. 55, 08 2011.

[Ble12] David M. Blei, *Probabilistic topic models*, Commun. ACM **55** (2012), no. 4, 77–84.

[BMNG16] Mark Belford, Brian Mac Namee, and Derek Greene, *Ensemble topic modeling via matrix factorization*, 09 2016.

[BMNG17] ———, *Stability of topic modeling via matrix factorization*, Expert Systems with Applications **91** (2017).

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, *Latent dirichlet allocation*, J. Mach. Learn. Res. **3** (2003), 993–1022.

[BSS16] Jerzy Blaszczynski, Jerzy Stefanowski, and Roman Slowinski, *Consistency driven feature subspace aggregating for ordinal classification*, vol. 9920, 10 2016.

[BWHY05] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao, *Diversity creation methods: A survey and categorisation*, Information Fusion **6** (2005), 5–20.

[BWT05] Gavin Brown, Jeremy Wyatt, and Peter Tino, *Managing diversity in regression ensembles*, Journal of Machine Learning Research **6** (2005), 1621–1650.

[BYRN11] Ricardo Baeza-Yates and Berthier A. Ribeiro-Neto, *Modern information retrieval - the concepts and technology behind search, second edition*, 2011.

[BYRnM+99] Ricardo Baeza-Yates, Berthier Ribeiro-neto, Don Mills, Ontario Bonn, San Juan, Milan Mexico, City Taipei, Addison Wesley, and Longman Limited, *Modern information retrieval*.

[Cal94] James P. Callan, *Passage level evidence in document retrieval*, Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Berlin, Heidelberg), SIGIR 94, Springer Verlag, 1994, p. 302 310.

[CCH+04] Nitesh Chawla, Nitesh Ca, Lawrence Hall, Kevin Bowyer, W Kegelmeyer, and Wpk@ca Gov, *Learning ensembles from bites: A scalable and accurate approach*, Journal of Machine Learning Research **5** (2004), 421–451.

[Col19] Glasgow Test Collection, *Glasgow test collection*, 2019.

[CPL15] Ronan Cummins, Jiaul Paik, and Yuanhua Lv, *A polya urn document language model for improved information retrieval*, ACM Transactions on Information Systems **33** (2015).

[CS95] Philip Chan and Salvatore Stolfo, *Learning arbiter and combiner trees from partitioned data for scaling machine learning*.

[DB75] Lauren B. Doyle and Joseph. Becker, *Information retrieval and processing / lauren b. doyle*, Melville Pub. Co Los Angeles, 1975 (English).

[DB95] Thomas G. Dietterich and Ghulum Bakiri, *Solving multiclass learning problems via error-correcting output codes*, CoRR **cs.AI/9501101** (1995).

[DCT12] Mauro Dragoni, Celia CostaPereira, and Andrea Tettamanzi, *A conceptual representation of documents and queries for information retrieval systems by using light ontologies*, Expert Systems with Applications **39** (2012), 10376 10388.

[Die00] Thomas G. Dietterich, *Ensemble methods in machine learning*, Multiple Classifier Systems (Berlin, Heidelberg), Springer Berlin Heidelberg, 2000, pp. 1–15.

[Don00] David L. Donoho, *High-dimensional data analysis: The curses and blessings of dimensionality*, AMS Conference on math challenges of the 21st century, 2000.

[DRTV13] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir, *A time series forest for classification and feature extraction*, Information Sciences **239** (2013), 142–153.

[DWV99] Harris Drucker, Donghui Wu, and V.N. Vapnik, *Support vector machines for spam categorization*, Neural Networks, IEEE Transactions on **10** (1999), 1048 – 1054.

[EAA15] Emad Elabd, Eissa Alshari, and Hatem M. Abdelkader, *Semantic boolean arabic information retrieval*, CoRR **abs/1512.03167** (2015).

[EAGS16] Haytham Elghazel, Alex Aussem, Ouadie Gharroudi, and Wafa Saadaoui, *Ensemble multi-label text categorization based on rotation forest and latent semantic indexing*, Expert Systems with Applications **57** (2016).

[EL07] Martin Emms and Saturnino Luz, *Machine learning for natural language processing.*

[FS95] Yoav Freund and Robert E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting.*

[FS97] Yoav Freund and Robert E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, J. Comput. Syst. Sci. **55** (1997), no. 1, 119–139.

[GE03] Isabelle Guyon and André Elisseeff, *An introduction to variable and feature selection*, J. Mach. Learn. Res. **3** (2003), 1157–1182.

[GLJG16] Fangyu Gai, Zhiqiang Li, Xinwen Jiang, and Hongchen Guo, *Enhance adaboost algorithm by integrating lda topic model*, 27–37.

[GO17] Kamel Garrouch and Mohamed Nazih Omri, *Bayesian network based information retrieval model*, 2017 International Conference on High Performance Computing Simulation (HPCS), 2017, pp. 193–200.

[HL93] Donna Harman and Mark Liberman, *Tipster complete ldc93t3a*, Web Download. Philadelphia, Linguistic Data Consortium, 1993.

[HSNC18] Shufeng Hao, Chongyang Shi, Zhendong Niu, and Longbing Cao, *Concept coupling learning for improving concept lattice-based document retrieval*, Eng. Appl. Artif. Intell. **69** (2018), 65–75.

[HWWG13] Li Hongmei, Hao Wenning, Gan Wenyan, and Chen Gang, *Survey of probabilistic graphical models*, 11 2013, pp. 275–280.

[IO16] Ayodeji Ibitoye and Olufade Onifade, *Fuzzy latent semantic query expansion model for enhancing information retrieval*, International Journal of modern education and computer science **8** (2016), 49–53.

[JBJ16] B. Jadhav, D. Bhosale, and D. Jadhav, *Pattern based topic model for data mining*, 08 2016, pp. 1–6.

[Jon00] Karen Jones, *Further reflections on trec*, Inf. Process. Manage. **36** (2000), 37–85.

[JR10] Jim Jansen and Soo Young Rieh, *The seventeentheoretical constructs of information searching and information retrieval*, Journal of The American Society for Information Science and Technology - JASIS **61** (2010), 1517–1534.

[Jr.13] Henry E. Kyburg Jr., *Uncertain inferences and uncertain conclusions*, CoRR **abs/1302.3589** (2013).

[JWY+18] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao, *Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey*, Multimedia Tools and Applications **78** (2018).

[KC17] Donald Kraft and Erin Colvin, *Fuzzy information retrieval*, Synthesis Lectures on Information Concepts, Retrieval, and Services **9** (2017), i–63.

[KJSR15] Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush, *Character aware neural language models.*

[KLMH13] Myungjae Kwak, Gondy Leroy, Jesse Martinez, and Jeff Harwell, *Development and evaluation of a biomedical search engine using a predicate-based vector space model.*, Journal of biomedical informatics **33** (2013).

[KMI18] Surya Kallumadi, Bhaskar Mitra, and Tereza Iofciu, *A line in the sand: Recommendation or ad-hoc retrieval?*, 2018.

[KTMA16] Yannis Korkontzelos, Beverley Thomas, Makoto Miwa, and Sophia Ananiadou, *Ensemble classification of grants using LDA-based features*, Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (Portorož, Slovenia), European Language Resources Association (ELRA), May 2016, pp. 1288–1294.

[Kun14a] Ludmila Kuncheva, *Combining pattern classifiers: Methods and algorithms: Second edition*, vol. 47, 01 2014.

[Kun14b] ———, *Combining pattern classifiers: Methods and algorithms: Second edition*, vol. 47, 01 2014.

[KZS+15] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler, *Skip-thought vectors*, CoRR **abs/1506.06726** (2015).

[LC12] Shih-Wei Lin and Shih-Chieh Chen, *Parameter determination and feature selection for c4.5 algorithm using scatter search approach*, Soft Comput. **16** (2012), 63–75.

[LMP+18] Xiaoxu Li, Zhanyu Ma, Pai Peng, Xiaowei Guo, Feiyue Huang, Xiaojie Wang, and Jun Guo, *Supervised latent dirichlet allocation with a mixture of sparse softmax*, Neurocomputing **312** (2018).

[LQQQ] La Lei, Guo Qiao, Cao Qimin, and Li Qitao, *Lda boost classification: boosting by topics*, EURASIP Journal on Advances in Signal Processing **2012**, no. 1 (English).

[LTD+16] Lin Liu, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou, *An overview of topic modeling and its current applications in bioinformatics*, SpringerPlus **5** (2016).

[Mac97] Richard Maclin, *An empirical evaluation of bagging and boosting*, In Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI Press, 1997, pp. 546–551.

[MDK+11] Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukas Burget, and Jan Cernocky, *Empirical evaluation and combination of advanced language modeling techniques*, 01 2011, pp. 605–608.

[ML02] Thomas Minka and John Lafferty, *Expectation-propagation for the generative aspect model*, Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (San Francisco, CA, USA), UAI'02, Morgan Kaufmann Publishers Inc., 2002, pp. 352–359.

[MRGO09] Jesus Maudes, Juan Rodriguez, and Cesar Garcia-Osorio, *Disturbing neighbors diversity for decision forests*, vol. 245, pp. 113–133, 10 2009.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, *Introduction to information retrieval*, Cambridge University Press, USA, 2008.

[Mur12] Kevin P. Murphy, *Machine learning: A probabilistic perspective*, The MIT Press, 2012.

[MWT+] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew Mccallum, *Optimizing semantic coherence in topic models*, 01, pp. 262–272.

[Nil65] Nils J Nilsson, *Learning machines; foundations of trainable pattern-classifying systems*, New York, McGraw-Hill, 1965.

[OM97] D.W. Opitz and R.F. Maclin, *An empirical evaluation of bagging and boosting for artificial neural networks*, Neural Networks,1997., International Conference on, vol. 3, Jun 1997, pp. 1401–1405 vol.3.

[Ona18] Aytug Onan, *Biomedical text categorization based on ensemble pruning and optimized topic modelling*, Computational and Mathematical Methods in Medicine **2018** (2018), 1–22.

[Pai13] Jiaul H. Paik, *A novel tfidf weighting scheme for effective ranking*, Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (2013).

[PBS16] Arnu Pretorius, Surette Bierman, and Sarel Steel, *A bias-variance analysis of ensemble learning for classification*, 12 2016.

[PGCB13] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Y. Bengio, *How to construct deep recurrent neural networks.*, CoRR (2013).

[PKVP06] Sophia Petridou, Vassiliki Koutsonikola, Athena Vakali, and Georgios Papadimitriou, *A divergence-oriented approach for web users clustering*, 05 2006, pp. 1229–1238.

[Pol06] Robi Polikar, *Polikar, r.: Ensemble based systems in decision making. ieee circuit syst. mag. 6, 21-45*, Circuits and Systems Magazine, IEEE **6** (2006), 21 – 45.

[POO18] Mohsen Pourvali, Salvatore Orlando, and Hosna Omidvarborna, *Topic models and fusion methods: a union to improve text clustering and cluster labeling*, International Journal of Interactive Multimedia and Artificial Intelligence **InPress** (2018), 7.

[QLYL18] Jipeng Qiang, Yun Li, Yun-Hao Yuan, and Wei Liu, *Snapshot ensembles of non-negative matrix factorization for stability of topic modeling*, Applied Intelligence **48** (2018).

[QWCZ14] Abdulhakim Qahtan, Suojin Wang, Raymond Carroll, and Xiangliang Zhang, *A new study of two divergence metrics for change detection in data streams*, vol. 263, 08 2014.

[RC13] Andrew K. Rider and Nitesh V. Chawla, *An ensemble topic model for sharing healthcare data and predicting disease risk*, Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (New York, NY, USA), BCB'13, ACM, 2013, pp. 333:333–333:340.

[Rij79] C. J. Van Rijsbergen, *Information retrieval*, 2nd ed., Butterworth-Heinemann, USA, 1979.

[Rok08] Lior Rokach, *Rokach, l.: Genetic algorithm-based feature set partitioning for classification problems. pattern recognition letters 41, 1676-1700*, Pattern Recognition **41** (2008), 1676–1700.

[Rok09] ———, *Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography*, Computational Statistics and Data Analysis **53** (2009), 4046–4072.

[RW12] Venkatesh Ramanathan and Harry Wechsler, *Phishing website detection using latent dirichlet allocation and adaboost*, ISI'12, 2012, pp. 102–107.

[RZGSS04] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth, *The author-topic model for authors and documents*, Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, UAI '04, AUAI Press, 2004, pp. 487–494.

[SB88] Gerard Salton and Chris Buckley, *Term-weighting approaches in automatic text retrieval.*, Inf. Process. Manag. **24** (1988), no. 5, 513–523.

[SBYM12] Mohammed Shatnawi, Muneer Bani Yassein, and Reem Mahafza, *A framework for retrieving arabic documents based on queries written in arabic slang language*, Journal of Information Science **38** (2012), 350–365.

[Sch90] Robert E. Schapire, *The strength of weak learnability*, Mach. Learn. **5** (1990), no. 2, 197–227.

[SD12] Jitendra Nath Singh and Sanjay Kumar Dwivedi, *Analysis of vector space model in information retrieval*, 2012.

[SG01] Amit Singhal and I. Google, *Modern information retrieval: A brief overview*, IEEE Data Engineering Bulletin **24** (2001).

[SHG+14] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Gregoire Mesnil, *A latent semantic model with convolutional-pooling structure for information retrieval*, CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management (2014), 101–110.

[SLM09] Liangcai Shu, Bo Long, and Weiyi Meng, *A latent topic model for complete entity resolution*, 03 2009, pp. 880–891.

[SLSB12] Igor Santos, Carlos Laorden, Borja Sanz, and Pablo Bringas, *Enhanced topic-based vector space model for semantics-aware spam filtering*, Expert Systems with Applications **39** (2012), 437–444.

[SLYS10] Zhiyong Shen, Ping Luo, Shengwen Yang, and Xukun Shen, *Topic modeling ensembles.*, ICDM (Geoffrey I. Webb, Bing Liu 0001, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu, eds.), IEEE Computer Society, July 2010, pp. 1031–1036.

[SNS15] Martin Sundermeyer, Hermann Ney, and Ralf Schluter, *From feedforward to recurrent lstm neural networks for language modeling*, Audio, Speech, and Language Processing, IEEE/ACM Transactions on **23** (2015), 517–529.

[SORN17] Ghulam Sarwar, Colm O' Riordan, and John Newell, *Passage level evidence for effective document level retrieval*, 01 2017, pp. 83–90.

[SS00] Robert E. Schapire and Yoram Singer, *Boostexter: A boosting-based system for text categorization*, Machine Learning, 2000, pp. 135–168.

[STGCL12] Yan-Tao Zheng Sheng Tang, Yong-Dong Zhang Gang Cao, and Jin-Tao Li, *Ensemble learning with lda topic models for visual concept detection*, Trans. Multi. (2012), 175–200.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc Le, *Sequence to sequence learning with neural networks*, Advances in Neural Information Processing Systems **4** (2014).

[TP10] Peter Turney and Patrick Pantel, *From frequency to meaning: Vector space models of semantics*, Journal of Artificial Intelligence Research **37** (2010).

[TWJS14] Tabea Tietz, Jörg Waitelonis, Joscha Jäger, and Harald Sack, *Smart media navigator: Visualizing recommendations based on linked data*, International Semantic Web Conference, 2014.

[TWT+11] Kai Ting, Jonathan Wells, Swee Chuan Tan, Shyh Teng, and Geoffrey Webb, *Feature-subspace aggregating: Ensembles for stable and unstable learners*, Machine Learning **82** (2011), 375–397.

[UR15] Ricardo Usbeck and Michael Roder, *Gerbil general entity annotator benchmarking framework*, 05 2015.

[VM02] Giorgio Valentini and Francesco Masulli, *Ensembles of learning machines*, vol. 2486, 05 2002, pp. 3–22.

[Voo04] Ellen Voorhees, *Overview of the trec 2004 robust track.*, 01 2004.

[WBM07] Shengli Wu, Yaxin Bi, and Sally McClean, *Applying statistical principles to data fusion in information retrieval*, 11 2007, pp. 313–319.

[WC06a] Xing Wei and W. Croft, *Lda-based document models for ad-hoc retrieval*, 01 2006, pp. 178–185.

[WC06b] Xing Wei and W. Bruce Croft, *Lda-based document models for ad-hoc retrieval*, Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '06, ACM, 2006, pp. 178–185.

[WES15] Jorg Waitelonis, Claudia Exeler, and Harald Sack, *Linked data enabled generalized vector space model to improve document retrieval*, 10 2015.

[WG14] Y. Wang and Q. Guo, *Multi-lda hybrid topic model with boosting strategy and its application in text classification*, Proceedings of the 33rd Chinese Control Conference, July 2014, pp. 4802–4806.

[WH13] Ben Wise and Max Henrion, *A framework for comparing uncertain inference systems to probability.*

[WLC16] Yanshan Wang, Jaesung Lee, and In Chan Choi, *Indexing by latent dirichlet allocation and ensemble model*, Journal of the Association for Information Science and Technology **67** (2016), 1736–1750.

[Wu18] Qiuyi Wu, *Topic modeling with lda tutorial*, 02 2018.

[XZL11] Deyi Xiong, Min Zhang, and Haizhou Li, *Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers.*, vol. 1, 01 2011, pp. 1288–1297.

[ZC13] Guoqiang Zhong and Mohamed Cheriet, *Adaptive error-correcting output codes*, 08 2013, pp. 1932–1938.

[Zei12] Matthew Zeiler, *Adadelta: An adaptive learning rate method.*

[ZJX+15] Shiliang Zhang, Hui Jiang, Mingbin Xu, JunFeng Hou, and LiRong Dai, *The fixed-size ordinally-forgetting encoding method for neural network language models*, 495–500.

[ZL01] Chengxiang Zhai and John Lafferty, *A study of smoothing methods for language models applied to ad hoc information retrieval*, Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '01, ACM, 2001, pp. 334–342.

[ZSCH16] Simon Zhao, Salman Salloum, Yeshou Cai, and Joshua Huang, *Ensemble subspace clustering of text data using two-level features*, International Journal of Machine Learning and Cybernetics **8** (2016).

[ZThM16] Joey Zhou, Ivor Tsang, Shen Shyang ho, and Klaus-Robert Muller, *N-ary error correcting coding scheme*.

[ZZZ15] Yanfei Zhong, Qiqi Zhu, and Liangpei Zhang, *Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery*, IEEE Transactions on Geoscience and Remote Sensing **53** (2015), 6207–6222.