INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN COMPUTACION

TESIS:

# “Clasificación temática de textos basada en transformaciones semánticas”

PARA OBTENER EL GRADO DE:
DOCTORADO EN CIENCIAS DE LA COMPUTACION

PRESENTA:
M. en C. JORGE VICTOR CARRERA TREJO

DIRECTOR(ES) DE TESIS:
DR. GRIGORI SIDOROV
DR. MARCO A. MORENO IBARRA

MEXICO, D.F., 2015

SIP-14-BIS

# INSTITUTO POLITÉCNICO NACIONAL
## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 16:00 horas del día 28 del mes de abril de 2015 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

***Centro de Investigación en Computación***

para examinar la tesis titulada:

**"Clasificación temática de textos basada en transformaciones semánticas"**

Presentada por el alumno:

| CARRERA | TREJO | JORGE VÍCTOR |
|---|---|---|
| Apellido paterno | Apellido materno | Nombre(s) |

Con registro: B 1 1 0 8 7 1

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

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

LA COMISIÓN REVISORA

Directores de tesis

Dr. Marco Antonio Moreno Ibarra

Dr. Grigori Sidorov

Dr. Oleksiy Pogrebnyak

Dr. Alexander Gelbukh

Dr. Miguel Jesús Torres Ruiz

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
DIRECCIÓN

*INSTITUTO POLITÉCNICO NACIONAL*

*SECRETARÍA DE INVESTIGACIÓN Y POSGRADO*

## *CARTA CESIÓN DE DERECHOS*

En la Ciudad de México el día 19 del mes de Junio del año 2015, el (la) que suscribe M. en C. Jorge Víctor Carrera Trejo alumno (a) del Programa de Doctorado en Ciencias de la Computación con número de registro B110871, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Grigori Sidorov y Dr. Marco Antonio Moreno Ibarra y cede los derechos del trabajo intitulado Clasificación temática de textos basada en transformaciones semánticas, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

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

M. en C. Jorge Víctor Carrera Trejo

# Resumen

La información contenida en un documento puede hacer referencia a uno o más tópicos, los cuáles son identificados durante el proceso de clasificación. Para realizar esta clasificación, es necesario definir un conjunto de características del documento e introducirlas en un clasificador, es decir, construir un modelo espacio vectorial y aplicarlo a un clasificador supervisado. Un problema en este tipo de clasificación es encontrar el conjunto de características "correcto", i.e., el conjunto que permita obtener los mejores resultados en la clasificación de un conjunto de documentos. Las características que pueden ser utilizadas son: palabras, N-gramas, N-gramas sintácticos, etc., o una combinación entre ellas. Hay varios tipos de N-gramas dependiendo del tipo de elementos que los componen, como son POS tags, relaciones de dependencia, etc. Este trabajo propone complementar los conjuntos de características tradicionales con los resultados de la aplicación de los algoritmos: latent Dirichlet allocation (LDA) y/o latent semantic analysis (LSA). Se realizaron experimentos utilizando los conjuntos de características propuestos en un corpus multi etiquetado de documentos del corpus Reuters-21578. Clasificando estos conjuntos con el algoritmo R*a*kel1 (con Naïve Bayes como algoritmo base de multi clasificación), comparando los resultados obtenidos con otros conjuntos de características usando la medida F1. En los experimentos, se muestra que los mejores resultados son obtenidos utilizando los conjuntos de características que contienen el complemento semántico (LDA, LSA).

# Abstract

The information contained in a document can refer to one or more topics, which are identified during the classification process. To perform this classification, it is necessary to define a set of features of the document and feed them into a classifier, i.e., to construct a Vector Space Model and apply it with a supervised classifier. A problem in this type of classification is to find a "correct" set of features, i.e., the set that allows obtaining the best results in the classification of the documents. The features that can be used are: words, n-grams, syntactic n-grams, etc. There are n-grmas of various types depending on the types of elements they are composed from, such as POS tags, dependency relations, etc., or a combination of these. This work proposes to complement traditional sets of features with the results of application of the following algorithms: latent Dirichlet allocation (LDA) and/or latent semantic analysis (LSA). We made experiments using the new sets of features for a multi labeled corpus of documents from Reuters-21578 collection. We classified them with the algorithm Ra*k*el1 (Naïve Bayes based multi classification algorithm), comparing the results obtained with other feature sets using the F1 measure. In the experiments, we show that the best results are obtained using the sets of features with the semantic complement (LDA, LSA).

# Agradecimientos

A Dios, por permitirme llegar a este momento de mi vida…

A mis Padres, por enseñarme el camino…

A Xochitl, por ayudarme a seguir adelante, por su tiempo y por todo su apoyo …

A mis asesores, Dr. Grigori Sidorov y Dr. Marco A. Moreno Ibarra, por todo su apoyo, paciencia, conocimientos y por creer en mi trabajo...

Al Dr. Evgueni Polchkov†, muchas gracias, dónde quiera que esté…

A todos mis amigos, por hacer el ambiente divertido y competitivo, pero sobre por aguantarme todo este tiempo…

A todos mis profesores por brindarme sus conocimientos, su amistad y su apoyo…

Y al Instituto Politécnico Nacional, por enseñarme lo que significa tener el corazón guinda y blanco…

# Contenido

# LISTA DE FIGURAS

Página

# LISTA DE TABLAS

Página

# 1. Introducción

Tradicionalmente en el campo de la clasificación, se considera a un elemento como perteneciente a una sola categoría a partir de un conjunto de ellas.

No obstante, en el campo del lenguaje natural un documento, considerando su contenido, es posible que sea multi etiquetado, es decir, asignado dentro de una o varias categorías en un espacio predefinido de clases o tópicos.

Por ejemplo, se puede observar un texto cuyo contenido este relacionado con información de alguna región agrícola, por su naturaleza, dicho texto puede ser incluido dentro de un tópico económico, por lo que se puede suponer que pertenece solamente a dicho tópico, lo que se puede determinar utilizando un esquema de clasificación tradicional, aunque, si se observan más a detalle las palabras que contiene el documento y sus relaciones, se podría determinar alguna pertenencia con algún otro tópico, como puede ser del tipo geográfico, financiero, ecológico, entre otros, por lo que se considera que el documento puede ser multi clasificado, como se observa en la figura 1.1.

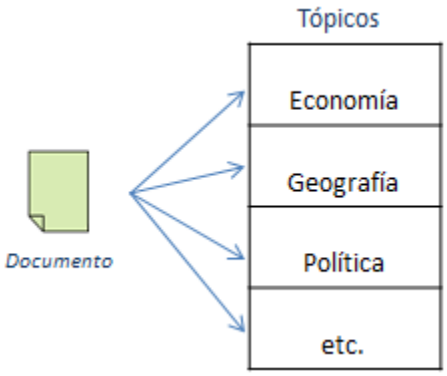


Figura 1.1. Documento multi clasificado.

Por otro lado, debido al rápido crecimiento de los dispositivos informáticos, actualmente, se tiene una gran necesidad de información que alimente a estos dispositivos en forma puntual y específica, identificando tópicos y textos que puedan aportar esa información,

considerando además que la información aportada por un documento puede aplicarse en diversos tópicos.

Por lo que se hace de vital importancia contar con métodos que permitan manejar y organizar la información de forma automática y óptima, construyendo para ello diversos clasificadores basados en el contenido de un conjunto o corpus de documentos.

Los datos de entrada de estos clasificadores basan su representación en modelos de representación vectorial con las siguientes características [1].

- Alta dimensionalidad del espacio vectorial. Los documentos contienen una gran cantidad de palabras, las cuáles engloban sus características y determinan la dimensión de los vectores.

- Pocas características irrelevantes. Un documento cuenta con muy pocas palabras que pueden ser consideradas como irrelevantes y que pueden ser eliminadas.

Los modelos vectoriales se crean utilizando principalmente el modelo *tf –idf*, term frequency –inverse document frequency, el cual tiene como características o componentes de los vectores a las palabras que contienen los documentos, utilizando todas ellas o bien eliminando aquellas que se denominan *stop words* u otras.

Actualmente, se proponen utilizar otro tipo de características basadas en las palabras, como pueden ser los lemas o stemms, o bien, si se consideran enunciados o frases se pueden utilizar bigramas, trigramas, etc, conocidos formalmente como N-gramas.

Si se toman en cuenta las relaciones entre las palabras se pueden utilizar N-gramas sintácticos en forma continua [2] o no [3]. O bien, finalmente se puede hacer uso de una combinación entre diversas características.

El desarrollo de esta tesis se enfoca en proponer un conjunto de características de un corpus de documentos con base en un modelo de espacio vectorial, con el objetivo que sean

clasificadas utilizando algún algoritmo de multi etiquetado comparando los resultados obtenidos con el modelo tradicional de palabras.

## 1.1. Planteamiento del problema

El presenta trabajo propone una método de caracterización para un corpus de documentos multi etiquetado que permita mejorar la clasificación de dicho corpus.

Este esquema se basa en el modelo espacio vectorial utilizando *tf –idf* para su construcción, como características se proponen palabras junto con los resultados de los algoritmos de reducción espacial latent Dirichlet allocation (LDA) y latent semantic analysis (LSA).

Para evaluar los resultados obtenidos, estos se compararán con aquellos mostrados utilizando solamente *tf –idf*, en el cual se utilizan únicamente las palabras de los documentos como características, como corpus de prueba se emplea el corpus Reuters-21578, todo esto en un ambiente supervisado.

El método propuesto se muestra en el siguiente ejemplo, consideremos un corpus multietiquetado de documentos, en el cual se identifica un conjunto de 4 tópicos, por lo que se observa que cada uno de los documentos pertenece a más de un tópico.

Cada uno de los documentos se representa mediante un par de vectores, que incluyen las palabras que contienen utilizando valores numéricos, para ello se utiliza en un vector el algoritmo *tf –idf*, el cual es el más utilizado en tareas de clasificación, en el otro se utilizará la caracterización propuesta en el presente trabajo que contiene el modelo *tf –idf* complementado con su reducción LDA o LSA, finalmente a ambos vectores se les agregará las clases o etiquetas a las que pertenece el documento.

Consideremos un documento *d*, que puede ser representado mediante un vector *v* utilizando:

- Algoritmo *tf –idf*, $v_d$ = {**0.1 0.5 0.5 0.3 1 0 0 1**}.
- Propuesta desarrollada, $v_d$ = {**0.1 0.5 0.5 0.3** 0.2 0.5 **1 0 0 1**}.

Dónde,

- Los 4 primeros valores de ambos vectores, representan los resultados arrojados por el algoritmos *tf –idf* y los últimos 4 valores indican cada una de las clases a las que pertenece el documento, por lo que en este caso se observa que el documento *d* pertenece a los tópicos 1 y 4.
- Los valores 5 y 6 del vector basado en la propuesta desarrollada identifican al modelo LDA, basado en dos tópicos al ser únicamente dos valores.

Generando cada uno de los vectores $v_d$ para cada uno de los documentos $d_n$, en el corpus de documentos se crean dos espacios vectoriales, supongamos que nuestro corpus está conformado por tres documentos, en este caso el espacio vectorial $e_v$ basado únicamente en *tf –idf* se encontrará formado por tres vectores como pueden ser los que se muestran en la tabla 1.1.

Tabla 1.1. Modelo de espacio vectorial basado en *tf –idf*.

$$e_v = \begin{matrix} \{0.1\ 0.5\ 0.5\ 0.3\ 1\ 0\ 0\ 1\} \\ \{0.3\ 0.0\ 0.6\ 0.2\ 1\ 0\ 1\ 0\} \\ \{0.4\ 0.2\ 0.1\ 0.7\ 1\ 1\ 0\ 1\} \end{matrix}$$

Mientras que utilizando el modelo propuesto a desarrollar en el presente trabajo, se puede crear un espacio vectorial como el que se muestra en la tabla 1.2.

Tabla 1.2. Modelo de espacio vectorial basado en el modelo propuesto.

$$e_v = \begin{matrix} \{0.1\ 0.5\ 0.5\ 0.3\ 0.2\ 0.5\ 1\ 0\ 0\ 1\} \\ \{0.3\ 0.0\ 0.6\ 0.2\ 0.1\ 0.8\ 1\ 0\ 1\ 0\} \\ \{0.4\ 0.2\ 0.1\ 0.7\ 0.7\ 0.9\ 1\ 1\ 0\ 1\} \end{matrix}$$

Ambos espacios vectoriales servirán como entrada a un algoritmo de clasificación, para finalmente, con los resultados obtenidos calcular alguna medida, como puede ser F1, comparando los resultados de cada espacio vectorial, esperando observar que el modelo de caracterización desarrollado en este trabajo mejore los resultados del modelo obtenido utilizando *tf –idf*.

## 1.2. Hipótesis y objetivos

### 1.2.1. Hipótesis

Existe un conjunto de características semánticas que permiten mejorar la clasificación de un corpus de documentos en un entorno multi etiquetado.

### 1.2.2. Objetivo general

Identificar un conjunto de características construido a partir de un corpus multi etiquetado de documentos que permita mejorar la clasificación de dicho corpus con respecto a otros conjuntos de características usualmente más utilizadas.

### 1.2.3. Objetivos particulares

Los objetivos particulares que se persiguen en el desarrollo de esta tesis se enumeran a continuación.

- Diseñar e implementar métodos que permitan caracterizar un corpus de documentos en un espacio vectorial para los métodos de caracterización identificados en el estado del arte.
- Diseñar e implementar métodos que permitan complementar un espacio vectorial basado en un corpus de documentos con otro vectorial perteneciente al mismo corpus de documentos.
- Diseñar e implementar métodos que permitan comparar los diversos resultados de multi clasificación para los distintos modelos espacios vectoriales basados en diferentes características.
- Describir y desarrollar un caso de estudio.
- Definir complementos para los diferentes conjuntos de características de un corpus de documentos que permita mejorar la multi clasificación.

- Definir un conjunto de características que sea un caso base de comparación, con base en la literatura consultada.

## 1.3. Aportaciones científicas esperadas

Alguna de las aportaciones que se obtendrán con el desarrollo de este trabajo son las siguientes:

- Nuevos modelos espacio vectoriales, VSM, basados en la combinación del VSM de un conjunto de características y el VSM resultante de aplicarle un algoritmo de agrupación o reducción dimensional, para el mejoramiento de la clasificación de dicho conjunto.
- Una metodología que funcione como una heurística que identifique el mejor complemento aplicado a un VSM para mejorar su clasificación.

## 1.4. Justificación

En la organización de la información de un corpus de documentos, es importante identificar los diversos tópicos a los que pueden pertenecer dichos documentos, esto debido a las diversas palabras y a las relaciones que contienen.

Con lo cual se pueden reducir los tiempos de búsqueda de la información y a su vez identificar documentos que pueden aportar información en un espacio de búsqueda determinado, aún cuando considerando el origen del documento no pareciere esto posible, pero su contenido permite que esto sea posible.

Por lo que es importante desarrollar metodos basados en el contenido de los documentos que permitan identificar estos tópicos y organizar al corpus en dichos tópicos.

Por lo que este trabajo propone un esquema de caracterización utilizando un modelo espacio vectorial que permite clasificar un corpus multi etiquetado de documentos mejorando los esquemas tradicionales de caracterización basados en *tf –idf*.

## 1.5. Organización de la tesis

El resto de la tesis está organizada como sigue: el capítulo 2 describe los trabajos relacionados con conceptos básicos necesarios para entender el trabajo presentado. En el capítulo 3 se presentan los trabajos relacionados con el desarrollo de la tesis. El método propuesto se presenta en el capítulo 4, detallando el caso de estudio. En el capítulo 5 aparecen los resultados experimentales, mientras que las conclusiones y la propuesta de trabajo futuro se presentan en el capítulo 6.

# 2. Marco teórico

## 2.1. Representación de la información

La clasificación automática de textos es un problema que se refiere a asignar en forma automática un conjunto de documentos dentro de un conjunto de categorías predefinidas, entre los problemas que intenta resolver se encuentra el que se refiere a la gran cantidad de información online con la que se cuenta actualmente, en el cual se hace necesario contar el día de hoy con un resumen y una buena indexación del documento [4].

Actualmente se utilizan principalmente técnicas estadísticas para realizar esta clasificación, entre las que se pueden incluir modelos de regresión multi variada, clasificación basada en el vecino más cercano, probabilidad de Bayes, árboles de decisión, redes neuronales, aprendizaje basado en reglas y algoritmos de aprendizaje inductivo.

Una característica o dificultad de los problemas de clasificación de textos, es la alta dimensionalidad del espacio de características [1]. El conjunto de características base, usualmente utilizadas consiste de términos únicos, palabras o frases que se encuentran en los documentos, los cuáles pueden ser cientos o miles de términos relacionados con un corpus de documentos, lo cual hace difícil el manejo de los datos por algunos algoritmos de clasificación, como pueden ser las redes neuronales, dónde cada característica es una entrada de la red neuronal, por lo que pareciera deseable que se redujera este espacio dimensional, sin embargo, este espacio no se puede reducir drásticamente, ya se sacrifica el porcentaje de documentos correctamente clasificados, por lo que elegir correctamente la forma de representación de los documentos para así seleccionar sus características se vuelve una tarea importante.

En el desarrollo de esta tesis se observará en forma general a un documento como un modelo de bolsa de palabras ordenada en forma secuencial [5], dentro de la cual la representación más común utilizada [6], [7] es el modelo espacio-vectorial (VSM), en el

cual cada objeto es representado como vectores cuyas componentes son las características de los objetos, cada característica corresponde a una dimensión del vector y la dimensión total del vector está dada por el número total de características [8], dónde el valor de cada característica será calculada utilizando alguno de los siguientes trabajos.

## 2.1.1. Frecuencia de palabras (*tf*)

Para conocer la importancia de una palabra dentro de un documento se puede utilizar la frecuencia de palabras como medida o peso, esta medida se puede definir como la frecuencia $tf_{w,d}$, de la palabra $w$ en el documento $d$, es decir, el número de veces que se repite $w$ en el documento $d$, dónde $tf_{w,d}$ es siempre un valor positivo y real, lo cual se puede representar de acuerdo a la ecuación 2.1.1.

$$tf_{w,d} = frecuencia(w,d), \tag{2.1.1}$$

Sin embargo, no todas las palabras en un documento son igualmente importantes, es decir, algunas pueden aportar más información que otras con respecto al documento, por lo que al momento de realizar este cálculo se debe considerar que palabras se deben incluir y cuáles no, usualmente este conjunto de palabras de palabras lo forman aquellas con una muy alta o que son comunes a un gran número de documentos que se encuentren en un conjunto de estos.

Utilizando esta medida se ignora el orden en el cuál se encuentran los términos enfocándose en el número de ocurrencias de estos, por lo que este valor es un valor cuantitativo del documento, por ejemplo utilizando esta medida la frase "*Rosa es más bonita que Martha*" es idéntica a la frase "*Martha es más bonita que Rosa*", lo cual desde una perspectiva qualitativa es erróneo, no obstante, se puede observar que ambas frases manejan un concepto similar.

Un error que presenta la medida *tf* es considerar a todas las palabras igual de importantes al momento de realizar una búsqueda dentro de un conjunto de documentos o corpus, es decir,

utilizando la medida *tf* no se observa claramente la importancia de un término dentro de un corpus.

## 2.1.2. Indice invertido de documentos (*idf*)

La medida frecuencia de documentos, $df_t$, se refiere al número de documentos que contienen el término $t$ en un corpus $c$ o conjunto de estos determinado, esto se observa en la ecuación 2.1.2.

$$df_t = numero\_documentos(t, frecuencia(w, c)), \tag{2.1.2}$$

Al involucrarse el corpus, si el término se encuentra en un gran número de documentos, su $df_t$ será alto, lo que significa que el término se encuentra en una gran cantidad de documentos, en consecuencia si el término tiene un valor bajo significa que son pocos los documentos que lo contienen, por lo que es necesario escalar el valor $df_t$ con base en el tamaño o número de documentos que conforman el corpus y resaltar aquellos términos que no se encuentran usualmente, la frecuencia inversa de documentos ó $idf_t$ es una medida que cumple con lo antes planteado.

La definición de $idf_t$ se plantea a continuación, considerando un conjunto de documentos de tamaño $N$, el $idf_t$ para el término $t$ se calcula como se muestra en la ecuación 2.1.3.

$$idf_t = log\frac{N}{df_t}, \tag{2.1.3}$$

Utilizando esta ecuación, las palabras que tengan una frecuencia baja tendrán un valor de $idf_t$ alto, mientras que aquellas que tengan una frecuencia alta su $idf_t$ será bajo, además de encontrarse escalado el valor con respecto al tamaño del corpus.

## 2.1.3. Medida *tf-idf*

La medida $tf$- $idf$ es una medida que combina a las medidas $tf$ o frecuencia de términos, con $idf$ o índice invertido de documentos, dando como resultado un valor que corresponde

a cada término en cada documento. Es una medida numérica que indica el grado de relevancia de una palabra para un documento con respecto a un corpus.

El valor *tf –idf* aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero esto es compensado por la frecuencia de la palabra el corpus, lo que permite reconocer aquellas palabras que son generalmente más comunes que otras.

Esta medida se calcula como se muestra en la ecuación 2.1.4.

$$tf\text{-}idf_{t,d} = tf_{t,d} \text{ x } idf_{t,} \qquad (2.1.4)$$

El valor de esta medida es.

- Alto, cuando el término se encuentra muchas veces en un número pequeño de documentos, permitiendo asi discriminar estos documentos.
- Bajo, cuando el término aparece pocas veces en un documento o en muchos documentos, indicando asi que el término no es tan importante.
- Muy bajo, cuando el término se encuentra en prácticamente en todos los documentos.

## 2.1.4. N-gramas

Otro tipo de características basadas en las palabras que contiene un documento, son las conocidas como N-gramas, dentro de estos conceptos podemos encontrar los tradicionales y los sintácticos, los cuales se definen a continuación.

### 2.1.4.1. N-gramas tradicionales

Los N-gramas tradicionales se pueden definir como secuencias de elementos como aparecen en un documento [9].

La letra *N* indica cuántos elementos forman parte de dicha secuencia, por lo que define la longitud de esta, es decir, del N-grama, por ejemplo, si $N = 1$, se refiere a un 1-grama ó unigrama, Si $N = 2$ se refiere a un 2-grama ó bigrama.

En la tabla 2.1.1 se da un enunciado de ejemplo y algunos de los N-gramas que se pueden construir tomando como base este.

Tabla 2.1.1. Ejemplos de N-gramas.

| Enunciado | Los forasteros son 4 artistas | | | | |
|---|---|---|---|---|---|
| ***1***-Grama, Unigrama | Los | forasteros | son | 4 | artistas |
| ***2***-Gramas, Bigrama | Los forasteros | forasteros son | son 4 | 4 artistas | |
| ***3***-Gramas, Trigrama | Los forasteros son | forasteros son 4 | son 4 artistas | | |

Como se puede observar en la tabla 2.1.1, cuando se utilizan unigramas, es decir, aquellos N-gramas con longitud 1, se están utilizando palabras.

Una características de los N-gramas es el grado de libertad del tipo de elementos que pueden formar el este, estos pueden ser lemas ó palabras, pero también pueden ser etiquetas de clases gramaticales tales como sustantivos, verbos, etc.

Los N-gramas tradicionales representan la información sintagmática ignorando información sintáctica y son ampliamente utilizados en varias tareas de la lingüística tradicional con buenos resultados [5].

## 2.1.4.2. N-gramas sintácticos

Los N-gramas sintácticos es un modelo de N-gramas que permite involucrar información sintáctica, ya que con estos se logra evitar el ruido introducido por la estructura superficial del lenguaje, debido a que en este nivel las palabras no relacionadas sintácticamente pueden aparecer juntas, no obstante, dos palabras que se encuentran relacionadas sintácticamente no necesariamente son vecinas inmediatas.

Estos N-gramas se construyen siguiendo la secuencia de la ruta del árbol sintáctico correspondiente al enunciado a partir del cuál se generan los N-gramas buscados, por ejemplo, para el siguiente enunciado, se puede construir el árbol sintáctico que se muestra en la figura 2.1.1.

"*El doctor Ferguson se ocupaba desde hacia mucho tiempo de todos los pormenores de su expedición.*"

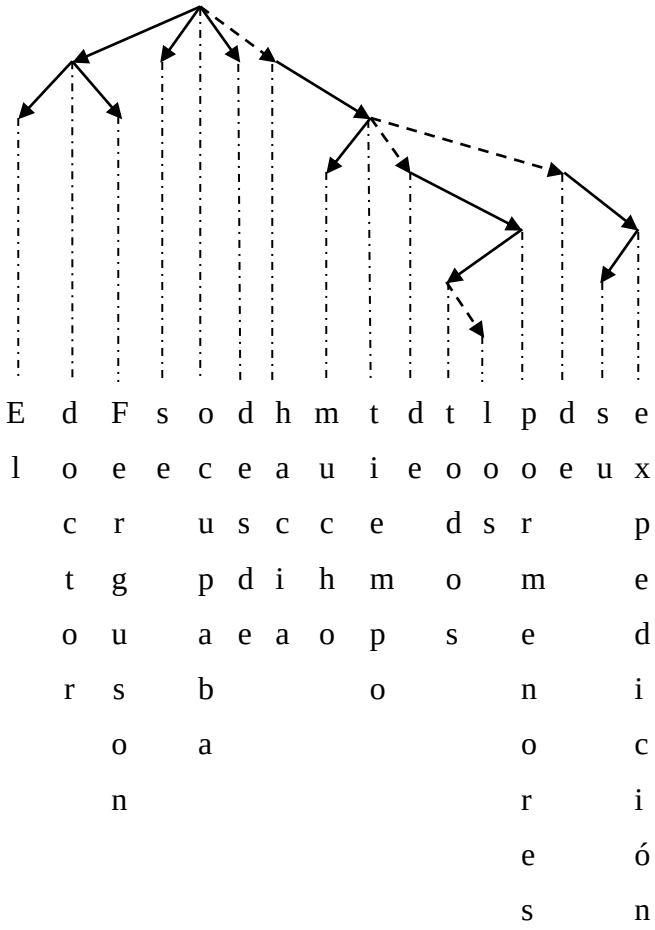


Figura 2.1.1. Ejemplo de árbol sintáctico.

Existen dos tipos de N-gramas sintácticos.

- Continuos, son aquellos N-gramas que para su construcción siguen una ruta continua utilizando el árbol sintáctico, es decir, las que se encuentran en nodos padre forman el N-grama con las palabras que se encuentran en sus nodos hijo, por ejemplo, utilizando el árbol de la figura 2.1.1, se pueden obtener los siguientes 2-gramas: (ocupaba, se), (ocupaba, desde), (tiemp, mucho), (expedición, su), (doctor, el), (todos, los), etc.
- No continuos, son aquellos N-gramas dónde siguiendo la ruta del árbol sintáctico se pueden utilizar bifurcaciones, es decir, las palabras que se encuentran en nodos padres forman el n-grama con aquellas palabras que no son descendientes directos, por ejemplo, para utilizando el árbol sintáctico de la figura 2.1.1 se pueden formar los

siguientes 2-gramas no continuos: (ocupaba, el), (ocupaba, ferguson), (hacía, de), (tiempo, pormenores).

## 2.1.5. Modelo espacio vectorial

Utilizando diversos valores de caracterización para un documento, es claro ver que este puede ser representado como un vector, cuyas componentes estan definidas por cada uno de los términos que componen al documento; para el caso de un conjunto de documentos, se puede generar un diccionario que consiste en todas las palabras que forman parte de todos los documentos, contruyendo con ellas un vector general. La representación documento – vector debe capturar la importancia de cada uno de las palabras.

Para el caso en particular de cada uno de los documentos, utilizando el vector generado a partir del diccionario, si alguna palabra no se encuentra en el documento, el peso o valor de dicha componente será 0.

Finalmente se denomina modelo espacial vectorial o VSM al conjunto de vectores que representan a cada uno de los documentos que pertenecen a un corpus de estos, el cual sirve como base para realizar operaciones de clasificación ó agrupamiento.

## 2.1.6. Discusión

En este capítulo se muestran las formas de caracterización de la información que son las más ampliamente utilizadas en trabajos de clasificación, así como, nuevas metodologías de caracterización que ofrecen buenos resultados en la clasificación de documentos más no son utilizadas en gran medida en la multi clasificación, si no que en este ámbito se utilizan principalmente las características tradicionales como son *unigramas* o *n-gramas*.

Cabe mencionar que los modelos de clasificación basados en modelos espacio vectorial se basan principalmente en el uso de *tf –idf.*

## 2.2. Agrupamiento temático de la información

Anteriormente, cuando se realizaba la búsqueda de algunos términos o palabras dentro de un documento, se identificaban exactamente las palabras o términos en el mismo orden en el que fueron introducidos, posteriormente haciendo uso de operaciones booleanas, se realizaba su búsqueda eliminando el orden, rankeando a los documentos según el resultado de la búsqueda.

Sin embargo, este tipo de métodos son inadecuados, ya que no pueden expresar conceptos tales como sinonimia dentro del proceso de búsqueda, sumado al hecho en el cual muchas palabras pueden expresar diferentes conceptos, lo cual es conocido como polisema, por lo que muchos documentos que puediesen ser relevantes para la búsqueda son ignorados, ya que al no contener exactamente los términos a buscar son considerados como irrelevantes [12].

Una solución al problema antes mencionado dentro de la recuperación de información se refiere a conceptualizar un documento considerando su temática ó significado [12] permitiendo organizar la información utilizando sus estructuras semánticas [12], [13].

En este capítulo se muestran dos de los métodos principales que permiten categorizar un corpus de documentos identificando los tópicos a los que pueden pertenecer, el primero de ellos conocido como *latent Dirichlet allocation* (LDA), el cual esta basado en fundamentos probabilísticos, siendo el segundo el método conocido como *latent semantic indexing* (LSI), también llamado *latent semantic analysis* (LSA), basado en fundamentes estadísticos.

LSI al igual que LDA son también conocidos por ser métodos de reducción dimensional, en los cuales, con un base en un espacio de representación vectorial de dimensión $K$, también considerado de alta dimensionalidad se reduce a un espacio de dimensión $k$, o de menor dimensionalidad, de tal manera que $K \gg k$. El valor de $k$ es un valor que se desea como dimensión del espacio de representación usualmente es escogido entre 100 y 150 [13], [14], [15], por lo que cada vector en el espacio de representación tendrá este valor en su dimensión.

Este valor $k$ es considerado un valor de tópicos, en el cual se agrupa un corpus de documentos en estos $k$ tópicos.

## 2.2.1. Indexación semántica latente (*Latent Semantic Indexing*)

### 2.2.1.1. Descripción

El modelo de indexación semántica latente se fundamenta en la aplicación de una técnica matemática denominada *Singular Value Decomposition* o *SVD*, la cual se basa en la construcción de una matriz de palabras – documento y que se muestra en la figura 2.2.1, esta técnica esta estrechamente relacionada con la descomposición en *Eigenvectores* o vectores propios y factores de análisis [15]. *SVD* permite identificar patrones de asociación entre los datos reflejados en la matriz [14]. Como resultado identifica aquellos términos que sean más cercanos a los documentos, aún cuando cuando pueden o no aparecer en ellos.

LSI es una metodología que proyecta a un corpus de documentos en un espacio de dimensiones semánticas latentes, en un espacio semántico latente, una búsqueda en particular y un documento pueden ser similares aún cuando no compartan términos o palabras en común, pero que contengan términos que sean semánticamente similares.

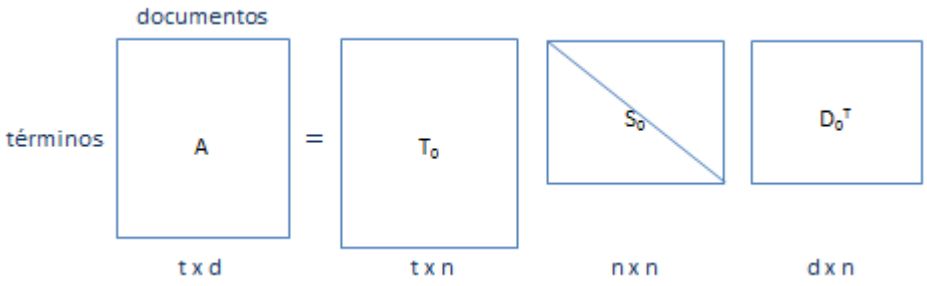


Figura 2.2.1. Esquema gráfico de *Singular Value Decomposition* o *SVD*.

## 2.2.1.2. Algoritmo

El algoritmo que sigue el modelo LSI es el siguiente.

- Considerando un corpus de documentos se debe construir una matriz $A_{txd}$, de términos $(t)$ - documentos $(d)$, que servirá de entrada al algoritmo LSI, dónde los términos representan las características del corpus de documentos, para su calculo se utiliza alguna técnica de caracterización de la información como puede ser *tf –idf*.
- Utilizando *SVD*, se descompone la matiz de entrada en una proyección que corresponde al producto de tres matrices $T_{txn}$, $S_{nxn}$ y $D_{dxn}$, como se muestra en la ecuación 2.2.1.

$$A_{txd} = T_{txn} S_{nxn} (D_{dxn})^T, \qquad (2.2.1)$$

dónde $t$ es el número de términos, $d$ el número de documentos y $n = \min(t, d)$, es la dimensión del espacio vectorial de representación, $T$ y $D$ contienen columnas ortonormales, es decir, cumplen con la ecuación 2.2.2.

$$TT^T = D^T D = I, \qquad (2.2.2)$$

- Utilizando las matrices construidas durante el proceso de proyección, se construye una nueva matriz $A'_{txd}$, de acuerdo a la ecuación 2.2.3, realizando el producto entre las matrices.

$$A'_{txd} = T_{txm} S_{mxm} (D_{dxm})^T, \qquad (2.2.3)$$

Observando la ecuación 2.2.3 se puede observar la agrupación del corpus de documentos en este caso en $m$ tópicos, siendo $n \gg m$.

- Finalmente utilizando la matriz $A'_{txd}$ se identifican las relaciones semánticas de términos siendo estos agrupados en $m$ conjuntos que pertenecen a cada uno de los tópicos y un conjunto de valores de pertenencia a cada uno de los tópicos por cada uno de los documentos.

## 2.2.2. Asignación de Dirichlet latente (*Latent Dirichlet Allocation*)

### 2.2.2.1. Descripción

El modelo de asignación de Dirichlet latente (LDA) se refiere a un modelo generativo probabilístico de un corpus, para el cual los documentos que componen este corpus, pueden representarse como una combinación vectorial de las probabilidades de pertenencia de estos a cada uno de los elementos de un conjunto de tópicos, siendo cada tópico caracterizado por una distribución probabilística de un conjunto de palabras del corpus, por lo que es necesario al utilizar el modelo LDA indicar el número de tópicos en el cuál se quiere dividir el corpus.

El cálculo del modelo LDA toma como entrada el modelo *tf –idf* de un corpus documentos y un número n de tópicos a partir de los cuáles el modelo generativo determinará las probabilidades de pertenencia de cada documento, obteniéndose así un modelo vectorial del corpus, esto se muestra en la figura 2.2.2.

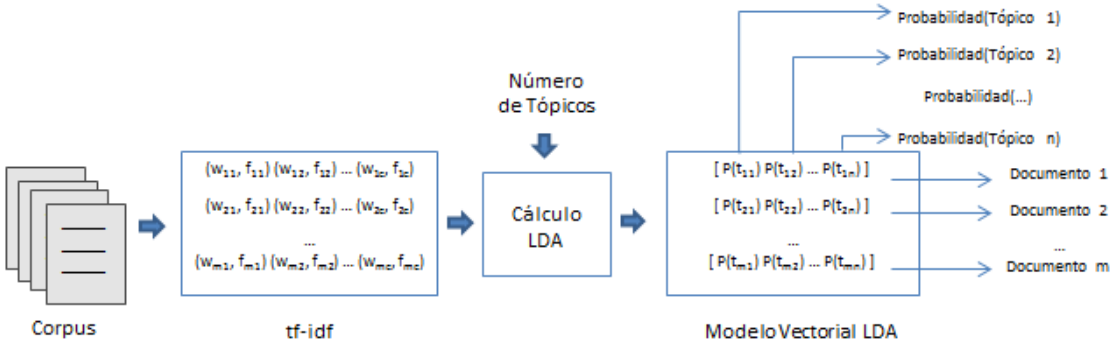


Figura 2.2.2. Latent Dirichlet Allocation.

### 2.2.2.2. Algoritmo

El algoritmo que sigue el modelo LDA es el siguiente.

- Considerando un corpus de documentos se debe construir una matriz $A_{dxt}$, de documentos $(d)$ - términos $(t)$, que servirá de entrada al algoritmo LDA junto con el número de tópicos a los cuáles se quiere conocer la pertenencia de los distintos documentos.

  Los términos representan las características del corpus de documentos, para su calculo se utiliza alguna técnica de caracterización de la información como puede ser *tf –idf*.
- Considerando cada documento $d$ perteneciente al corpus y la matriz $A_{dxt}$, se calcula mediante una distribución de Poisson el número de palabras o términos $t$ que contiene cada documento $d$.
- Haciendo uso de una distribución de Dirichlet se calcula la probabilidad $P$ de pertenencia de este documento a cada uno de los tópicos de entrada.
- Mediante un proceso iterativo, por cada palabra $w_i$ en un documento $d$ para un tópico $t_n$, utilizando una distribución multinomial, se calcula su probabilidad multinomial condicionada $P(w_i/t_n)$ de la palabra $w_i$ con respecto al tópico $t_n$ [10], [11].
- Finalmente, se devuelve un conjunto de palabras por cada uno de los tópicos y un conjunto de probabilidades de pertenencia a cada uno de los tópicos por cada uno de los documentos.

## 2.2.3. Discusión

Es este capítulo se muestran las metodologías latent Dirichlet allocation y latent semantic indexing, o también conocida como latent semantic analisys, que permiten realizar una reducción dimensional, aunque una característica que se quiere resaltar es que permiten generar un conjunto de características basadas en la agrupación por tópicos, la cual, puede ser utilizada en la clasificación en varias clases, por lo que en este trabajo se busca explotar dicha característica.

Dicho conjunto de características son arrojadas en forma de un modelo de espacio vectorial lo que permite compararlo o complementarlo con otro tipo de características utilizando el mismo modelo.

## 2.3. Clasificación y multi clasificación

La clasificación de documentos es una tarea fundamental dentro de los sistemas de información, ya que permite conocer los distintos contextos o temáticas a las que pertenecen las fuentes de información, que pueden ser documentos, imágenes, videos, etc, con base en los distintos metadatos, descripciones, palabras y/o frases que contienen, adquiriendo estas, un significado distinto dependiendo de su relación semántica con otras palabras y frases dentro de un contexto particular [16].

Tradicionalmente se ha conceptualizado la clasificación con el paradigma de asociar un elemento con solamente una clase de un conjunto de clases a las que puede pertenecer, observando a esta tarea como una binaria, es decir, el elemento pertenece o no a la clase. No obstante, en los últimos años en aplicaciones relacionadas con *machine learning* e *information retrieval* se ha puesto en manifiesto la importancia de asociar las fuentes de información no solamente con una clase, considerando a este problema no como un problema binario, dado que un valor no binario puede pertenecer a un subconjunto del conjunto de clases, lo que se conoce como multi etiquetado o multi clasificación.

En este capítulo se observarán ambos conceptos, tanto el de clasificación como el de multi clasificación.

### 2.3.1. Clasificación

Dentro de *machine learning*, considerando un contexto de aprendizaje supervisado, dónde se tiene un conjunto de objetos y un conjunto de clases a las que pertenecen los objetos, se puede definir como clasificación al uso técnicas de aprendizaje supervisado, en las cuáles un objeto cualesquiera es representado por un vector de características es asociado con alguna categoría [18].

Lo anterior se puede representar de la siguiente manera.

Sea $X$ un conjunto de objetos y $Y$ un conjunto de categorías a las que pertenecen los objetos, el objetivo del aprendizaje supervisado es encontrar una función $f: X \to Y$ a partir de un conjunto de entrenamiento o aprendizaje $\{ (x_i, y_i) \mid 1 \leq i \leq m \}$ , dónde $x_i \in X$ es una instancia caracterizada de un objeto y $y_i \in Y$ se refiere a la categoría a la que pertenece el objeto según la semántica de su caracterización.

Por ejemplo consideremos un documento que describe un pelicula y un conjunto de clases, entre las que se encuentran acción, suspenso, comedia, drama, etc. Al ejecutar un clasificador, este determina de acuerdo a sus reglas que pertenece a la clase de suspenso, ignorando en que medida puede pertenecer a las otras clases. Esto se muestra en la figura 2.3.1.

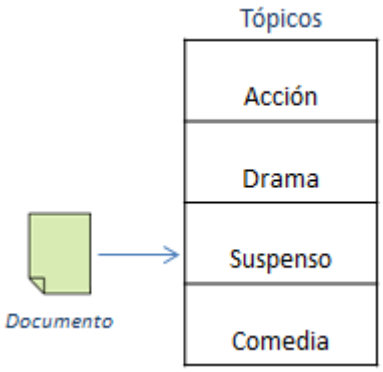


Figura 2.3.1. Clasificación en solamente un tópico.

## 2.3.1.1. Algoritmos

Los clasificadores de aprendizaje supervisado utilizan métodos de aprendizaje inductivo, que permiten clasificar información a partir de un conjunto de aprendizaje, entre los más utilizados se encuentran [35].

- Árboles de decisión.
- Naïve Bayes.
- Redes de Bayes.
- Máquinas de Soporte Vectorial – *SVM* –.

Estos algoritmos se presentan a continuación.

### 2.3.1.1.1. Árboles de decisión

Los árboles de decisión también son conocidos como árboles de clasificación y se refieren a un método que permite generar reglas de clasificación en forma automática, normalmente conocidas como reglas decisión, utilizando una estructura tipo árbol [36].

Tabla 2.3.1. Datos para la construcción de un árbol de decisión.

| Outlook | Temp. (ºF) | Humidity (%) | Windy | Class |
|---|---|---|---|---|
| sunny | 75 | 70 | true | play |
| sunny | 80 | 90 | true | don't play |
| sunny | 85 | 85 | false | don't play |
| sunny | 72 | 95 | false | don't play |
| sunny | 69 | 70 | false | play |
| overcast | 72 | 90 | true | play |
| overcast | 83 | 78 | false | play |
| overcast | 64 | 65 | true | play |
| overcast | 81 | 75 | false | play |
| rain | 71 | 80 | true | don't play |
| rain | 65 | 70 | true | don't play |
| rain | 75 | 80 | false | play |
| rain | 68 | 80 | false | play |
| rain | 70 | 96 | false | play |

Las reglas de decisión se generan con base en un conjunto de datos de aprendizaje, por ejemplo consideremos los datos mostrados en la tabla 2.3.1, el árbol de decisión generado con estos datos, indicando las reglas de decisión fuera de los recuadros, se muestra en la figura 2.3.2.

Los árboles de decisión es uno de los métodos de clasificación más sencillos y fáciles de implementar, toma de entrada un objeto o situación descrita por un conjunto de atributos y regresa una decisión "true/false".

En general pueden tener un rango más amplio de decisiones, no solamente las booleanas, cada nodo interno corresponde a una prueba en el valor de uno de los atributos y las ramas están etiquetadas con los posibles valores de la prueba. Finalmente, las hojas especifican la clasificación, en el ejemplo, se refieren a una decisión.

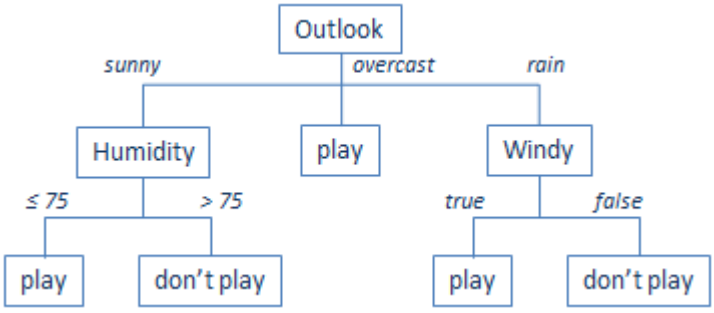


Figura 2.3.2. Árbol de decisión.

## 2.3.1.1.2. Naïve Bayes

Este tipo de clasificadores se basa en un área de las matemáticas denominada teoría de la probabilidad, siendo su objetivo clasificar un elemento en la clase a la que sea más parecido, i.e., cuya probabilidad de pertenencia sea más alta [36].

Este algoritmo utiliza probabilidad condicional permitiendo observar la importancia de un atributo de una instancia en particular independientemente de sus demás atributos dentro de la clasificación, siendo uno de los clasificadores que mejores resultados arroja.

Su algoritmo se basa en lo siguiente.

Dado un conjunto de $k$ clases $k = \{c_1, c_2, \dots, c_n\}$ mutuamente excluyentes cuyas probabilidades son $P(c_1), P(c_2), \dots, P(c_k)$ y un conjunto de $w$ instancias de $n$ atributos, cuyos valores son $v_1, v_2, \dots, v_n$.

La probabilidad de que la j-ésima instancia $w_j$, pertenezca a la m-ésima clase $c_m$ se calcula de acuerdo a la ecuación 2.3.1.

$$P(c_m) x P(v_1 \; y \; v_2 \; y \dots y \; v_n | c_m), \qquad (2.3.1)$$

Asumiendo que todos los atributos son independientes, el valor de la ecuación 2.3.1 puede ser calculado de acuerdo a la ecuación 2.3.2.

$$P(c_m)xP(v_1|c_m)xP(v_2|c_m)x \ldots xP(v_n|c_m), \qquad (2.3.2)$$

Utilizando la ecuación anterior se calcula el valor para todas las clases y se coloca a la instancia en la clase cuya probabilidad sea más alta.

### 2.3.1.1.3. Redes de Bayes

Una red bayesiana es un modelo gráfico que codifica relaciones probabilísticas entre las variables de interés, en la actualidad se han convertido en la representación más popular de conocimiento en los sistemas expertos y se han diseñado métodos que permiten el aprendizaje a las redes bayesianas a partir de un conjunto de datos de aprendizaje [37].

Cuando se usa en conjunción con técnicas estadísticas, el modelo gráfico tiene varias ventajas para el análisis de datos.

- El modelo codifica dependencias entre todas las variables por lo que el modelo es capaz de manejar variables con datos faltantes.
- Una red bayesiana se puede utilizar para aprender las relaciones entre los datos de aprendizaje, por lo que se puede utilizar para entender el dominio sobre el cual trabaja y así poder realizar la clasificación de datos de prueba.
- Dado que el modelo puede aprender e identifica las relaciones entre sus entradas, semántica probabilística, es una representación ideal para combinar el conocimiento previo ó datos de aprendizaje y datos de prueba.
- Los métodos estadísticos bayesianos en conjunto con redes bayesianas ofrecen un enfoque eficiente y de principios para evitar el sobreajuste de datos.

Formalmente, las redes bayesianas son grafos dirigidos acíclicos cuyos nodos representan variables aleatorias en el sentido de Naïve Bayes: las mismas pueden ser cantidades observables, variables latentes, parámetros desconocidos o hipótesis. Las aristas representan dependencias condicionales, los nodos que no se encuentran conectados representan variables las cuales son condicionalmente independientes de las otras. Cada

nodo tiene asociado una función de probabilidad que toma como entrada un conjunto particular de valores de las variables iniciales del nodo y devuelve la probabilidad de la variable representada por el nodo.

Ideas similares pueden ser aplicadas a grafos no dirigidos y posiblemente cíclicos, como son las llamadas redes de Markov.

En la figura 2.3.3 se muestra una red bayesiana simple en la que se indica la influencia de la lluvia sobre un rociador de agua y de este a su vez sobre la hierba húmeda.

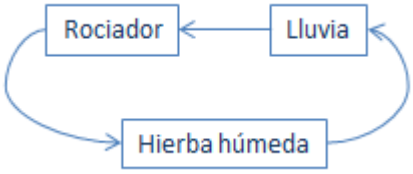


Figura 2.3.3. Red Bayesiana simple.

### 2.3.1.1.4. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial se basan en el principio de minimización del riesgo estructural – *structural risk minimization* – de la teoría del aprendizaje computacional. La idea detrás de esta teoría es: "*Encontrar una hipótesis* $h$ *para la cuál se garantice el mínimo error verdadero*" [38].

El error verdadero de $h$ es la probabilidad de clasificar erróneamente un ejemplo de prueba elegido al azar, se define un límite superior que permita relacionar la hipótesis $h$ y la complejidad del espacio de trabajo $H$, definido por la dimensión del vector de características $VC$, el cual es el espacio de trabajo de la hipótesis $h$. Las máquinas de soporte vectorial encuentran la la hipótesis $h$ el cuál minimiza el límite del error verdadero controlando la dimensión de $VC$ de $H$ [1].

Las SVM's son aprendices universales, ya que en su forma básica, aprenden utilizando una función lineal definiendo un umbral, aunque, se pueden utilizar otras funciones de

aprendizaje tales como funciones polinomiales, redes neuronales de base radial (RBF) y perceptrones multicapa.

Una propiedad importante de la SVM's es que su capacidad de aprender es independiente de la dimensión del espacio de características, es decir, la complejidad de la hipótesis se basa en la distancia de separación de las clases a clasificar, no del número de características, esto significa que la hipótesis se puede generalizar siempre que se cuente con un gran número de características que hagan que las clases sean separables utilizando funciones del espacio de hipótesis.

## 2.3.2. Multi clasificación

Este tipo de tareas se incluyen principalemnto dentro un área denominada categorización o clasificación de textos [17] la cual se refiere a ordenar un corpus de documentos dentro de ninguno, uno o varios conjuntos definidos como categorías, temáticas, clases o etiquetas, lo cual es conocido como multi etiquetado [17], [18], [11], [22], siendo esta área una combinación entre *información retrieval* y *machine learning* [17], [19], [20], [21].

El multi etiquetado difiere de la clasificación tradicional en que en esta última, cada uno de los elementos a clasificar es relacionado con solamente una clase o tópico, mientras que en el multietiquetado pueden ser relacionados con más de un tópico [18], esta relación también es conocida como etiquetado,.

Por ejemplo, en el párrafo anterior 0 se menciona un documento que es clasificado o etiquetado solamente en el tópico *Suspenso*, no obstante, si consideramos los valores del documento con las otras clases, podemos definir una relación de pertenencia del documento con ellas. En la figura 2.3.4 se muestra un documento que es etiquetado en varias clases.

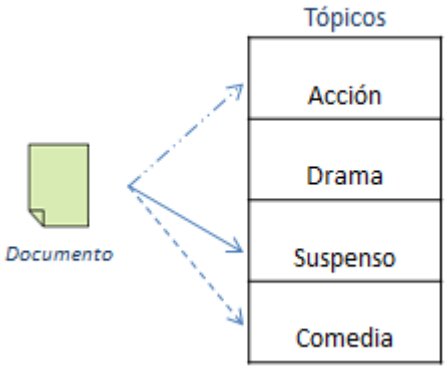


Figura 2.3.4. Multi clasificación de un documento.

## 2.3.2.1. Métodos de multi clasificación

Los métodos de multi clasificación se pueden agrupar en dos categorías principalmente [18], [22].

- Métodos de transformación de problemas.
- Métodos de adaptación de algoritmos.

### 2.3.2.1.1. Métodos de transformación de problemas

Este tipo de métodos se refieren a transformar el problema de multi clasificación en uno o más problemas de clasificación o bien trata el problema como un problema de regresión.

Este tipo de métodos se pueden ejemplicar de la siguiente manera.

En la tabla 2.3.2 se observan 4 documentos que pertenecen a uno o más de las siguientes clases: deportes, religión, ciencia y política.

Tabla 2.3.2. Ejemplo de un conjunto de datos multi etiquetados.

| Documento | Deportes | Religión | Ciencia | Política |
|---|---|---|---|---|
| 1 | X | | | X |
| 2 | | | X | X |
| 3 | X | | | |

| Documento | Deportes | Religión | Ciencia | Política |
|---|---|---|---|---|
| 4 | | X | X | |

Utilizando este tipo de métodos, el problema lo podemos tranformar en cuatro tipos, de acuerdo a [22], [23], los cuáles se listan a continuación.

- En la primera transformación o TF1, se selecciona en forma aleatoria sólo una de las clases de los elementos que pertenecen a más de una, esto se muestra en la tabla 2.3.3.

Tabla 2.3.3. Transformación TF1 del conjunto de datos multietiquetados.

| Documento | Deportes | Religión | Ciencia | Política |
|---|---|---|---|---|
| 1 | X | | | |
| 2 | | | | X |
| 3 | X | | | |
| 4 | | | X | |

Como se observa, se elimina una clase a los documentos 1, 2 y 4, el documento 3 esta relacionado solamente con una clase, por lo que no se le elimina ninguna clase.

- En la segunda transformación o TF2, se omiten aquellas instancias que no sean multi etiquetadas, lo que se puede observar en la tabla 2.3.4.

Tabla 2.3.4. Transformación TF2 del conjunto de datos multietiquetados.

| Documento | Deportes | Religión | Ciencia | Política |
|---|---|---|---|---|
| 3 | X | | | |

Como se puede ver el único documento no multi etiquetado es el número 3 por lo que es el único que no se elimina.

- Un tercer método de transformación o TF3, considera cada subconjunto de etiquetas como si se tratara solamente de una, para calcular todo el conjunto de combinaciones de etiquetas posibles se utiliza para ello el conjunto potencia del conjunto [23], [24], en la tabla 2.3.5 se puede observar un ejemplo de este nuevo conjunto de clases creado, tomando como referencia la información mostrada en la tabla 2.3.2.

- Existen un cuarto método de transformación o TF4, el cual es el más utilizado [23], [25], [26], [27], es aquel que plantea generar tantos clasificadores binarios como clases se tengan, que permitan categorizar a todos los elementos del conjunto a clasificar.
  El clasificador final será la unión de todos los clasificadores para cada clase. En la tabla 2.3.6 se muestran los clasificadores generados para nuestras clases de ejemplo.

Tabla 2.3.5. Transformación TF3 del conjunto de datos multietiquetados.

| Documento | Deportes | Deportes y Religión | Ciencia y Política | Ciencia y Religión |
|---|---|---|---|---|
| 1 | | X | | |
| 2 | | | X | |
| 3 | X | | | |
| 4 | | | | X |

Tabla 2.3.6. Transformación TF4 del conjunto de datos multietiquetados.

| Docto. | Deportes | ¬Deportes |
|---|---|---|
| 1 | X | |
| 2 | | X |
| 3 | X | |
| 4 | | X |

(a)

| Docto. | Política | ¬Política |
|---|---|---|
| 1 | X | |
| 2 | X | |
| 3 | | X |
| 4 | | X |

(b)

| Docto. | Religión | ¬Religión |
|---|---|---|
| 1 | X | |
| 2 | X | |
| 3 | | X |
| 4 | | X |

(c)

| Docto. | Ciencia | ¬Ciencia |
|---|---|---|
| 1 | | X |
| 2 | X | |
| 3 | | X |
| 4 | X | |

(d)

### 2.3.2.1.2. Métodos de adaptación de algoritmos

Los métodos de adaptación de algoritmos, se refieren a adaptar aquellos algoritmos de clasificación en un contexto de multi clasificación.

Algunos trabajos que presentación adaptación de algoritmos son.

- En [28] se adapta el algoritmo C4.5, para permitir el manejo de múltiples etiquetas en las hojas de los árboles.
- Adaboost.MH y Adaboost.MR [29] son dos extensiones del algoritmo Adaboost [31] para la clasificación multietiquetada.
- ML-*k*NN [30] es una adaptación del algoritmo *k*NN utilizando datos multietiquetados. Este método sigue la metodología TF4, dónde a diferencia del algoritmo *k*NN, ML-*k*NN tiene la capacidad de producir una lista de etiquetas a las que puede pertenecer un elemento.
- En [32] se presenta un modelo probabilístico generativo en el cual cada etiqueta esta generada por diferentes palabras, este modelo supone que cada documento es producido por una combinación de palabras de distintas etiquetas, cuando se presentaba un nuevo documento se utilizan reglas de Bayes para determinar su multi clasificación.
- En [33] presenta un algoritmo de ranking para la multi clasificación, utilizando la filosofía de las máquinas de soporte vectorial, *SVM*, sin embargo este trabajo no puede manejar conjuntos de etiquetas como clases al ser un algoritmo del tipo ranking.
- En [34] se presentan dos mejoras para los clasificadores basados en *SVM* utilizando la idea de los métodos de transformación de problemas TF4, en la primera de ellas se observa a cualquier algoritmo de clasificación como una extensión del método TF4, dónde la idea principal es complementar el conjunto de características con la salida de un clasificador binario, convirtiéndose este conjunto en la entrada de otro clasificador, resolviendo el problema con una combinación de clasificadores.

  La segunda mejora se refiere a eliminar las instancias que presenten dentro de los márgenes del funcionamiento específico de la *SVM* similitudes negativas similares o bien similitudes positivas y negativas similares.

## 2.3.3. Algoritmo RA*k*EL

Dentro de los métodos de transformación de problemas, se encuentran los algoritmos de multi clasificación en los cuáles se considera a cada subconjunto de etiquetas como solamente una etiqueta, este tipo de métodos se denominan métodos de aprendizaje *Label Powerset* (LP), una de sus ventajas es considerar las correlaciones entre las etiquetas y una desventaja es el gran número de subclases que se generan, dónde muchas de ellas tienen asociados un pequeño número de ejemplos [43], [44].

El algoritmo RA*k*EL o random *k*-labelsets propone un enfoque basado en un conjunto de clasificadores LP. Dónde cada clasificador LP es entrenado utilizando un pequeño subconjunto tomado en forma aleatoria del conjunto de etiquetas. Este enfoque tiene como objetivo considerar las correlaciones entre las etiquetas y al mismo tiempo evitar los problemas antes mencionados de los algoritmos LP.

### 2.3.3.1. Algoritmo

Para entender el algoritmo que sigue este método, se hace necesario definir el concepto de *k-labelsets*, sea $L = \{\lambda_i\}$, dónde $i = 1 \ldots |L|$ el conjunto de etiquetas en un dominio multi clasificado. Un conjunto $Y \subseteq L$con $k = |Y|$ es llamado *k-labelset*. También se usará el término $L^k$ para indicar todos los posibles conjuntos con *k-labelset* en $L$. El tamaño de $L^k$ esta dado por el coeficiente binomial $|L^k| = \binom{|L|}{k}$.

El algoritmo RA*k*EL construye en forma iterativa un conjunto de $m$ clasificadores Label Powerset (LP). En cada iteración $i = 1 \ldots m$, el algoritmo elige en forma aleatoria un *k-labelset* $Y_i$ de $L^k$ sin reemplazarlo. Este le permite al algoritmo aprender a partir de un clasificador LP $h_i: X \rightarrow P(Y_i)$. El pseudocódigo seria el siguiente.

**Entrada.** Número de modelos $m$, tamaño del labelset $k$, conjunto de etiquetas $L$, conjunto de entrenamiento $D$.
**Salida.** Un conjunto de clasificadores LP $h_i$ con sus correspondientes *k-labelsets* $Y_i$ $R \leftarrow L^k$.

En forma iterativa $i \leftarrow 1$ hasta $\min(m, |L^k|)$

$Y_i \leftarrow$ Un *k-labelset* seleccionado en forma aleatoria de $R$.

Entrenar un clasificador LP $h_i: X \rightarrow P(Y_i)$ utilizando $D$.

$R \leftarrow R \backslash \{Y_i\}$

El número de iteraciones $m$ es definido en un rango de valores entre $1$ y $|L^k|$ y el valor de $k$ se encuentra entre $2$ y $|L| - 1$, Para el caso en que $k = 1$ y $m = |L|$ el algoritmo construye un conjunto de clasificadores basados en el método *Binary Relevance* (*BR*), mientras que para $k = |L|$ y $m = 1$ se obtiene un clasificador para solamente una clase basado en el método LP.

RA*k*EL se basa en la hipótesis que al usar conjuntos de etiquetas de tamaño pequeño y un número adecuado de iteraciones, las correlaciones entre las etiquetas se manejarán correctamente.

## 2.3.4. Medidas de evaluación

La evaluación del rendimiento de los algoritmos de clasificación en el problema multi etiquetado es mucho más difícil que en el caso de la clasificación con una sola etiqueta [22], [42], [40]. Cuando un algoritmo asigna un conjunto de etiquetas, estas pueden ser muy pocas, por lo que faltarán algunas etiquetas correctas que deberían haber sido asignadas o bien se pueden asignar demasiadas, añadiendo algunas etiquetas que sean irrelevantes. Para cualquier etiqueta dada, fácilmente se puede determinar si es correcta o incorrecta su asignación, sin embargo para todo el conjunto de etiquetas asignadas es más complicado ya que es necesario evaluar todo el conjunto, es decir, todo el rendimiento, por lo que las métricas de evaluación normalmente deben ser promediadas. Desafortunadamente no existe una única medida o bien un conjunto de ellas que sean universalmente utilizadas [40].

En la evaluación de los resultados se utiliza principalmente la medida F1, la cual se calcula como un promedio armónico de las medidas precisión y recall [22], [39].

- Precision, la precisión es la proporción entre el número de documentos correctamente clasificados entre el número total de documentos clasificados, esta medida se define de acuerdo a la ecuación 2.3.3.

$$P = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Z_i|}, \qquad (2.3.3)$$

Dónde,

  - $Z_i$, son los documentos que fueron clasificados.
  - $Y_i$, son los documentos de entrenamientos de los cuales se conocen sus clases.

Entre más se acerque el valor de la métrica a 0, mayor será el número de documentos recuperados que no se clasificaron correctamente, mientras que si el valor de la precisión es igual a uno, todos los documentos recuperados son relevantes.

- Recall, recall es la proporción de documentos relevantes recuperados, comparado con el total de los documentos que son relevantes existentes en el corpus, recall se calcula como lo indica la ecuación 2.3.4.

$$R = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Y_i|}, \qquad (2.3.4)$$

Dónde,

  - $Z_i$, son los documentos que fueron clasificados.
  - $Y_i$, son los documentos de entrenamientos de los cuales se conocen sus clases.

Si el resultado de la ecuación 2.3.4 es igual a 1, todos los documentos fueron correctamente clasificados, por el contrario en el caso que el valor sea igual a cero, se tiene que los documentos obtenidos fueron erróneamente clasificados.

- Medida F1, la medida F1 es una combinación de las medidas precisión, *P*, y recall, *R*, se define como el promedio armónico de las dos métricas y es calculada de acuerdo a la ecuación 2.3.5.

$$F_1 = \frac{1}{m}\frac{2 * P * R}{P + R} = \frac{1}{m}\sum_{i=1}^{m}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|}, \qquad (2.3.5)$$

Si el resultado de la ecuación 2.3.5 es 1 se tiene una buena clasificación, por el contrario si es igual a 0 la clasificación es mala, cabe hacer notar que no se encuentra definida una proporción correcta, ya que ese concepto no se define de forma única en el ámbito de la multi clasificación [40].

## 2.3.5. Discusión

En este capítulo se pueden observar las diferencias principales entre clasificación orientada solamente a una clase y aquella orientada a múltiples clases o multi clasificación, resaltando una gran diferencia en sus respectivos algoritmos de clasificación, ya que aquellos que se refieren a la clasificación de clases únicas o uniclase son los algoritmos base de los algoritmos de multi clasificación, es decir, los multi clasificadores se orientan principalmente a descomponer el problema en clasificación uniclase, por lo que se hace importante conocer el funcionamiento de este tipo de clasificadores.

Finalmente es importante mencionar al algoritmo RA*k*EL dada que actualmente ha mostrado muy buenos resultados en el ámbito de la multi clasificación.

# 3. Antecedentes

Como se ha mencionado anteriormente la clasificación de documentos multi etiquetados ha recibido un especial interés [1], [11], [16], [17], [22], enfocándose principalmente en la búsqueda de métodos automáticos de clasificación con el objetivo que estos reemplacen a aquellos que son asistidos por medios humanos. Estos nuevos métodos se basan principalmente en varios tipos de características y utilizan diferentes algoritmos de clasificación.

Principalmente los algoritmos de multi clasificación utilizan modelos de espacio vectorial basados en diferentes características y *tf –idf* para calcular las relaciones numéricas de estas características, y finalmente utilizan como algoritmos base de clasificación Naïve Bayes, árboles de decisión, redes neuronales, máquinas de soporte vectorial, etc [35], [46].

En este capítulo se mostrarán algunos de los trabajos relacionados con esta tesis, en los cuales se proponen diferentes ideas para mejorar los resultados en la multi clasificación de documentos o bien proponen ideas que permitan realizar esta operación.

## 3.1. Modelos de características para un corpus de documentos

En este trabajo, titulado en inglés “Novel Unsupervised Features for Czech Multi-label Document Classification” [47], se presenta una propuesta de caracterización de un corpus de documentos multietiquetado en idioma checo que permita multi clasificarlos.

Esta propuesta de caracterización se basa en la utilización de un stemmer no supervisado, el algoritmo latent Dirichlet allocation (LDA) y en espacios semánticos, denominados HAL y COALS, como corpus de pruebas se utiliza el de la Agencia Checa de Noticias (CTK), en el cual los documentos pueden pertenecer a diferentes tópicos como son política, deportes, cultura, negocios entre otros.

Este trabajo compara sus resultados de clasificación con.

- Como caso base propone el uso de palabras, utilizando *tf –idf* para la construcción del modelo de espacio vectorial.
- Otro esquema es el uso de los stems de las palabras, lo cual permite el manejo del lenguaje checo más fácilmente, de igual manera con *tf –idf*.
- El uso de LDA utilizando el modelo de palabras.

La importancia de este trabajo radica en la propuesta del uso de espacios semánticos HAL – *Hyperspace Analogue to Language* – [48] y COALS – *Correlated Occurrence Analogue to Lexical Semantic* – [49] en la construcción del VSM de características, un espacio semántico es un modelo de distribución semántica en la cual las palabras o stems se agrupan en clusters de acuerdo a la distancia semántica entre las distintas palabras o stems, siendo estos clusters caractizados por una palabra o stem, por lo que las distintas palabras o stems en estos grupos se sustituyen por la palabra o stem que caracteriza al grupo.

Utilizando los espacios semánticos se propone la creación de dos VSM's más de características, uno basado en el espacio semántico HAL y el otro en el espacio COALS.

Finalmente, utilizando el esquema basado en los stems de las palabras y el algoritmo LDA, se crea un conjunto más de características denominado S-LDA (Stem-LDA) y diversos conjuntos basados en la combinación de diferentes VSM's.

El multi clasificador utilizado para desarrollar las pruebas es un clasificador $n$ binario, i.e., se utilizan diferentes clasificadores binarios, en los cuáles un documento se asigna a una clase si pertenece a ella, en caso contrario no se asigna. Este clasificador se encuentra implementado en el software MALLET [50].

Como medidas de evaluación se utiliza precision (P), recall (R) y la medida F1, como procedimiento de evaluación se utiliza *k-fold cross validation* con $k = 5$, i.e., se utiliza el 20 % de los datos como conjunto de prueba.

En la tabla 3.1.1 se muestran los resultados obtenidos durante la clasificación y comparación de los mejores VSM's, incluyendo el caso base.

Tabla 3.1.1. Resultados utilizando diferentes combinaciones.

| VSM | P (%) | R (%) | F1 (%) | Mejora F1 (%) |
|---|---|---|---|---|
| palabras | 88.1 | 72.7 | 79.7 | |
| stems | 86.4 | 75.0 | 80.3 | +0.7 |
| palabras + stems | 88.3 | 74.8 | 81.0 | +1.3 |
| palabras + HAL | 88.4 | 72.8 | 79.9 | +0.2 |
| palabras + COALS | 88.5 | 72.8 | 79.9 | +0.2 |
| palabras + S-LDA | 89.2 | 74.6 | 81.2 | +1.6 |
| palabras + stems + S-LDA | 88.8 | 75.5 | 81.6 | +1.9 |
| palabras + stems + S-LDA + COALS | 89.0 | 75.6 | 81.7 | +2.1 |

En el caso de los espacios semánticos se utilizaron.

- En el caso de COALS la combinación de 4 modelos, estos son 100, 500, 1000 y 5000 clusters.
- En el caso de HAL todos los modelos en el espacio semántico.

Para la creación del modelo S-LDA se utilizó la combinación de los VSM's S-LDA para 100 y 400 tópicos.

# 3.2. LDA en un ambiente multi etiquetado

Otro trabajo a considerar, es el denominado "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora" [11], en este trabajo se propone un método de selección de tags, dónde un tag se considera una palabra descriptiva de un documento, para un documento en un corpus multietiquetado, basado en el algoritmo latent Dirichlet allocation o LDA, denominado LDA etiquetado – *Labeled LDA* – o L-LDA.

En esta metodología se considera que LDA no es método apropiado para la generación de tópicos en un corpus multietiquetado, ya que LDA es una metodología que modela a un documento como una mezcla de tópicos pero que no es supervisada, estos tópicos son generados en forma automática como una combinación de palabras basándose en una distribución probabilística, pero se desconoce el significado de dichos tópicos, por lo que al tratarse de un corpus multietiquetado es complicado relacionar los distintos tópicos

generados con las distintas clases o etiquetas del corpus, por lo que propone una metodología que se base en un modelo generativo que sea supervisado para así generar una relación entre los tópicos generados y las distintas etiquetas del corpus.

L-LDA es un modelo que se basa en el modelo LDA, diferenciándose de este en el hecho de que en L-LDA se conocen los tópicos a los que pertenecen cada uno de los documentos por lo que al construir el modelo LDA se buscan las palabras que formarán parte de los tópicos ya conocidos, y en el modelo multinomial de Naïve Bayes al utilizarlo para buscar la probabilidad de pertenencia de cada uno de los documentos a sus etiquetas o clases.

Tabla 3.2.1. Comparación de L-LDA con SVM.

| Etiqueta | %MacroF1 | | %MicroF1 | |
|---|---|---|---|---|
| | L-LDA | SVM | L-LDA | SVM |
| Arts | 30.70 | 23.23 | 39.81 | 48.42 |
| Business | 30.81 | 22.82 | 67.00 | 72.15 |
| Computers | 27.55 | 18.29 | 48.95 | 61.97 |
| Education | 33.78 | 36.03 | 41.19 | 59.45 |
| Entertainment | 39.42 | 43.22 | 47.71 | 62.89 |
| Health | 45.36 | 47.86 | 58.13 | 72.21 |
| Recreation | 37.63 | 33.77 | 43.71 | 59.15 |
| Society | 27.32 | 23.89 | 42.98 | 52.29 |

Como corpus de prueba se utiliza un conjunto de documentos del sitio web de Yahoo clasificados en 8 clases, como son: *Arts, Business, Computers, Education, Entertainment, Health, Recreation, Society,* como características se utiliza los modelos más utilizados principalmente el modelo *tf* y *tf –idf* [51], [52], tanto en el caso base como en L-LDA, como clasificador para el caso base se utiliza un conjunto de clasificadores binarios basados en SVM, uno por clase, como medidas de evaluación se utilizaron *MacroF1* y *MicroF1*.

Dónde *MacroF1* se refiere al promedio de los valores de precisión y recall por clase o etiqueta y *MicroF1* a la suma por separado de los valores de falsos positivos, falsos negativos y verdaderos positivos obtenidos por cada clase y se realiza el calculo de la medida utilizando la fórmula de la medida F1.

En la tabla 3.2.1 se pueden ver los resultados de los clasificadores L-LDA y el conjunto de clasificadores binarios SVM, como se observa a nivel macro L-LDA ofrece mejores resultados, fallando a nivel micro.

## 3.3. Clasificador RA*k*EL

El clasificador denominado *RAndom k-labELsets* o RA*k*EL [43], [44] esta orientado al problema de multi etiquetado basado en la construcción de un conjunto de clasificadores label powerset.

En [44] se presenta una comparación de RA*k*EL contra otros clasificadores, considerando 8 corpus multietiquetados como corpus de pruebas los cuáles se muestran en tabla 3.3.1.

Tabla 3.3.1. Corpus de pruebas.

| Corpus | Documentos | Etiquetas |
|---|---|---|
| scene | 2407 | 6 |
| yeast | 2417 | 14 |
| tmc2007 | 28596 | 22 |
| medical | 978 | 45 |
| enron | 1702 | 53 |
| mediamill | 43907 | 101 |
| reuters (rcv1) | 6000 | 101 |
| bibtex | 7395 | 159 |

Como medidas de evaluación se basa en la utilización de las medidas F1 micro y macro, como algoritmos de comparación se utilizan.

- Binary Relevance.
- Label Powerset.
- ML*k*NN, multi label *k*NN.
- BPMLL, que se refiere al perceptrón multicapa entrenado con el algoritmo de retropropagación del error [55].
- CLR, método de comparación de pares denominado "*Calibrated Label Ranking*" [54].

En la tabla 3.3.2 se muestra un concentrado de los mejores clasificadores considerando la medida F1 micro y en la tabla 3.3.3 considerando la medida F1 macro, se consideran dos tipos del algoritmo RA*k*EL, diferenciándose en sus parámetros de entrada.

Tabla 3.3.2. Concentrado de la medida F1 micro.

| Corpus | BR | LP | RA*k*EL$_d$ | RA*k*EL$_o$ | ML*k*NN | BPMLL | CLR |
|---|---|---|---|---|---|---|---|
| scene | 4 | 5 | 6 | 2 | 1 | 7 | 3 |
| yeast | 5 | 7 | 6 | 3 | 1 | 2 | 4 |
| tmc2007 | 4 | 7 | 5 | 1 | 6 | 2 | 3 |
| medical | 1 | 5 | 3 | 2 | 6 | 7 | 4 |
| enron | 4 | 7 | 5 | 2 | 6 | 1 | 3 |
| mediamill | 4 | 7 | 5 | 1 | 3 | 6 | 2 |
| reuters (rcv1) | 2 | 5 | 3 | 1 | 6 | 7 | 4 |
| bibtex | 3 | 6 | 4 | 2 | 7 | 1 | 5 |
| average rank | 3.38 | 6.13 | 4.63 | 1.75 | 4.50 | 4.13 | 3.50 |

Tabla 3.3.3. Concentrado de la medida F1 macro.

| Corpus | BR | LP | RA*k*EL$_d$ | RA*k*EL$_o$ | ML*k*NN | BPMLL | CLR |
|---|---|---|---|---|---|---|---|
| scene | 4 | 5 | 6 | 2 | 1 | 7 | 3 |
| yeast | 5 | 6 | 3 | 2 | 7 | 1 | 4 |
| tmc2007 | 4 | 6 | 5 | 2 | 7 | 1 | 3 |
| medical | 2 | 5 | 3 | 1 | 7 | 6 | 4 |
| enron | 4 | 5 | 3 | 2 | 7 | 1 | 6 |
| mediamill | 3 | 4 | 2 | 1 | 5 | 7 | 6 |
| reuters (rcv1) | 1 | 4 | 3 | 2 | 6 | 7 | 5 |
| bibtex | 2 | 5 | 4 | 3 | 7 | 1 | 6 |
| average rank | 3.13 | 5.00 | 3.63 | 1.88 | 5.88 | 3.88 | 4.63 |

En [44] se pueden verificar más a detalle los resultados, en este trabajo se muestran solo los más representativos, como se puede observar en el ámbito de multi clasificación el algoritmo RA*k*EL es el que a nivel global ofrece mejores resultados [53].

## 3.4. Representación de la información en la clasificación de documentos

En el trabajo denominado "Rich document representation and classification: an analysis" [57] se destacan tres factores importantes dentro de la clasificación de documentos.

- Modelo de clasificación, que se refiere al clasificador utilizado.
- Medida de similitud.
- Modelo de representación del documento.

El desarrollo del trabajo se enfoca en demostrar que la representación del documento, es decir, el conjunto de características utilizadas, tienen un alto impacto en la calidad de la clasificación.

Como modelo de clasificación se utiliza un clasificador de documentos basado en el centroide, ya que se considera un clasificador simple pero que ofrece buenos resultados en comparación con otros clasificadores como son *k*NN o SVM, sin embargo, al utilizar este método se hace necesario conocer a priori el conjunto de las diferentes clases en las que se van a clasificar los documentos para poder así calcular los centroides de las mismas, por lo que como medida de similitud se utiliza la distancia a los diferentes centroides de las clases.

Como modelos de representación utilizando un modelo de espacio vectorial, se utilizan n-gramas, palabras, frases, una representación basada en la lógica del documento denominada RDR y la combinación entre ellos.

La representación RDR se construye mediante procesamiento lingüístico, utilizando una representación basada en lógica de primer orden. Por ejemplo en la frase "…*operating systems for personal computers*…" es posible identificar que la palabra *for* relaciona a las palabras "*operating systems*" y "*personal computers*", por lo que es posible representar el enunciado utilizando lógica de primer orden como *operating_system(personal_computers)*, por lo que se considera que RDR provee una representación semántica del documento.

Como medida de evaluación se utiliza la precisión.

Finalmente, como corpus de prueba se utiliza el corpus Reuters-21578, utilizando solamente aquellos documentos que pertenecen a un clase.

Con base en sus resultados, se observa que RDR es mejor que los N-gramas, palabras y frases, no obstante, también proponen el uso de combinaciones, ya que estas mejoran las combinaciones pero el modelo se vuelve más complejo.

## 3.5. Clasificación basada en una representación multi palabra

El trabajo denominado "Text classification based on multi-word with support vector machine" [58], plantea mejorar las características que forman el modelo de espacio vectorial a clasificar para así mejorar los resultados de esta operación o bien en su caso el de agrupamientar la información.

Se propone un método basado inicialmente en la extracción de palabras de un documento utilizando para ello una estructura sintáctica, para posteriormente representar a los documentos usando una estructura multipalabra utilizando para ello las palabras que fueron extraidas anteriormente.

Por ejemplo, en primer lugar, utilizando diversas expresiones regulares se extraen diferentes frases de los enunciados que componen a los documentos, por ejemplo, del enunciado.

"*The U.S. agriculture department last December slashed its 12 month of 1987 sugar import quota.*"

Es posible extraer.

- *U.S. agriculture department.*
- *U.S. agriculture.*
- *agriculture department.*
- *last December.*

- *sugar import quota.*
- *entre otras.*

Posteriormente se identifican por medios de heurísticas aquellas expresiones que definan un solo concepto utilizando un número mínimo de palabras, es importante mencionar que el concepto identificado sustituye a las a expresiones, es así que para las expresiones *U.S. agriculture department, U.S. agricultura* y *agriculture department* es posible utilizar los términos *U.S. agriculture* o *agriculture department.*

Como caso base de comparación se utilizan palabras, como clasificador se utilizan máquinas de soporte vectorial, una de ellas de kernel lineal y la otra con kernel no lineal.

El corpus de prueba esta basado en el corpus Reuters-21578, utilizando solamente 4 tópicos *grain, crude, trade* e *interest,* considerando 252 documentos para *grain,* 208 para *crude*, 133 para *interest* y 171 para *trade,* a este corpus se le realiza un preprocesamiento eliminando *stop words,* como medida de evaluación se utiliza el promedio.

En los resultados obtenidos se puede observar que la representación multipalabra, ofrece mejores resultados que el caso base.

# 3.6. Características de la información

Otros trabajos en los cuales se hace mención de la importancia de los conjuntos de características que sirven como entrada a los distintos algoritmos de multi etiquetado o de clasificación.

Por ejemplo en [1] se hace mención de la importancia que pueden tener todas las palabras como características en el proceso de clasificación, resaltando el problema que se tiene al eliminar palabras de un corpus, ya que de acuerdo a este trabajo en un corpus se cuentan con muy pocas palabras irrelevantes, ejemplifica un proceso de clasificación utilizando el corpus de Reuters no multi-etiquetado y una una máquina de soporte vectorial como clasificador, comparando sus resultados, los cuales son mejores, con respecto a otros clasificadores tales como Bayes, *k*NN ó C4.5.

En [8] se resalta la importancia de identificar relaciones entre las palabras utilizando diferentes medidas como es la distancia de Levenshtein como características, para posteriormente complementar el conjunto de características original, este trabajo se refiere a la clasificación utilizando un clasificador basado en la medida coseno.

Otros trabajos como [59] propone una métrica para evaluar las características en los documentos eliminando aquellas que sean redundantes, denominando esta métrica ganancia de información global – GIG –, proponiendo además un nuevo método de selección de características denominado maximización de la ganancia de información global – MGIG –, el cual comparan con 4 algoritmos de selección de características siendo el principal de ellos ganancia de información – IG –, el cual consideran como caso base, utilizan 6 corpus de pruebas entre el que se encuentra Reuters-21578, utilizando documentos uniclase, como medida de evaluación se utiliza F1 y como clasificadores Naïve Bayes y SVM, con base en sus resultados MGIG obtiene las mejores clasificiones.

Finalmente, en [35] se resalta la importancia de utilizar conjuntos de características como representaciones de los documentos que sean simples, ya que estas ofrecen mejores resultados que otras caracterizaciones que se basan en transformaciones más complejas como aquellas que utilizan análisis sintácico ó morfológico, como corpus de prueba se utiliza el corpus de Reuters-21578, como clasificadores se muestran Naïve Bayes, redes bayesianas, árboles de decisión y máquinas de soporte vectorial.

Es importante resaltar que en todos los trabajos mencionados se utiliza caracterizaciones basadas en modelos espacio vectorial y *tf*, *tf –idf* o *idf* para generar los valores numéricos de las VSM.

# 3.7. Discusión

En este capítulo, como se pueden observar en los diferentes trabajos presentados, los modelos de espacio vectorial que sirven como entrada a los diferentes algoritmos de clasificación, se construyen principalmente usando características tales como palabras o *unigramas*, *N-gramas* o en nuevas propuestas como son los *N-gramas sintácticos*, y con algoritmos como *tf, idf, tf –idf* para calcular los valores numéricos de estas características, para posteriormente realizar su clasificación, no obstante, algunos trabajos mencionan que los mejores resultados de clasificación no se obtienen utilizando solamente un tipo de características, si no que es posible combinar estas para que estos puedan mejorar.

Así también se comienzan a utilizar modelos tipo LDA o LSA como modelos de agrupamiento de documentos, identificando diferente tipos de tópicos, se propone el uso de estos modelos como características, pero no combinados, no obstante, algunos trabajos construyen nuevos modelos basados en LDA o LSA que ofrecen mejores resultados que los tradicionales LDA o LSA.

# 4. Método propuesto

Como se puede observar en varios de los trabajos presentados en el desarrollo de esta tésis, el modelo de espacio vectorial es el modelo generalmente utilizado para caracterizar a un corpus de documentos. También se observa que normalmente se utilizan como características las palabras dentro de un documento u otros elementos como los N-gramas, no obstante, en el ámbito de la multi clasificación, los mejores resultados obtenidos en operaciones de clasificación son aquellos que utilizan combinaciones de características.

Por lo que el método propuesto se basa en la sencilla idea de la combinación de características, como se observa en la figura 4.1, cumpliendo con el objetivo general planteado, en el que se busca identificar un conjunto de características que mejoren los resultados de clasificación utilizando un modelo de espacio vectorial, considerando un conjunto de características base como línea de comparación, en este caso aquellas basadas en modelos tradicionales.

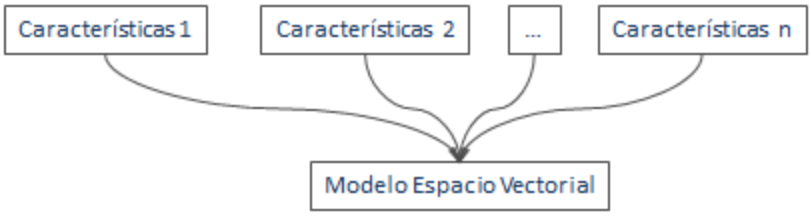


Figura 4.1. Modelo propuesto.

No obstante, al utilizar combinaciones de características se debe observar que el modelo de espacio vectorial se vuelve más complejo y en ocasiones inmanejable por los multi clasificadores, debido a la cada vez más alta dimensión de los vectores que lo conforman.

Por lo que en el modelo a implementar, se busca que sea procesable por el clasificador en el cual se implemente.

En los trabajos relacionados son utilizadas las combinaciones de características como palabras, N-gramas, y otros como N-gramas sintácticos, implementadas en diferentes

corpus, estas formas de caracterización se pueden considerar que guardan relaciones estadísticas, no obstante, también es posible considerar que tienen relaciones semánticas, dado que en su conjunto conforman a cada uno de los documentos y a su vez cada uno de ellos ofrecen un significado que sin las palabras que los componen cambiaría.

Otros modelos utilizan espacios semánticos combinados con las características antes mencionadas.

Observando el funcionamiento de los algoritmos latent dirichlet allocation y latent semantic indexing se puede identificar el uso que se puede hacer de ellos no utilizándolos para una reducción dimensional de un espacio vectorial, como se muestra en la figura 2.2.2 para el caso de LDA, lo mismo aplica para el caso de LSA, como algoritmos de agrupamiento.

Como se menciona en el apartado 3.2 y se comprueba en el apartado de resultados, utilizando solamente los modelos LDA y LSA no ofrecen buenos resultados en el ámbito de la multi clasificación, no obstante, en este trabajo se considera que los resultados arrojados por estos modelos ofrecen un buen complemento que mejore la clasificación, como se comprueba en el apartado de resultados.

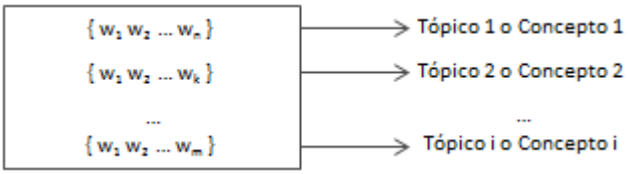


Figura 4.2. Modelo de tópicos LDA o LSA.

Este trabajo considera a este complemento como semántico ya que cada tópico es constituido por un conjunto de palabras relacionadas entre sí, que pueden describir un concepto, en la figura 4.2 se muestra un ejemplo del modelo de tópicos arrojado por un modelo LDA o LSA.

Cabe recordar que los modelos LDA y LSA construyen un modelo de espacio vectorial considerando un modelo de espacio vectorial construido con base en todo el corpus de documentos, dentro de este LDA-VSM o LSA-VSM, cada una de sus columnas pertenece a un tópico, y cada fila a un documento, dónde la relación documento –tópico indica el grado

de pertenecencia del documento al tópico, en la figura 4.3 se muestra el VSM generado por el algoritmo LDA.

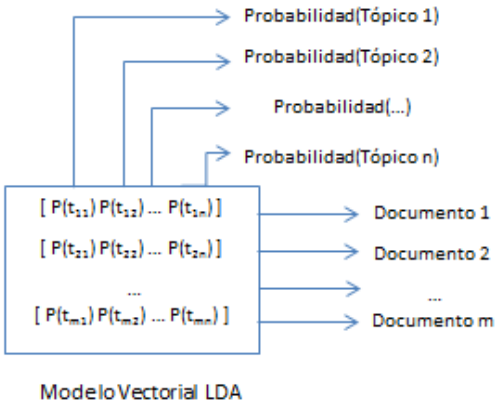


Figura 4.3. Modelo LDA-VSM.

Se propone utilizar los resultados de estos algoritmos como complemento de algún VSM, bajo la premisa en la cual suponemos que: si dos documentos son similares, entonces pertenecen a los mismos tópicos, por lo que sus respectivos vectores LDA y LSA son similares, por lo que cual al complementar el VSM original con su modelo LDA o LSA, la clasificación mejorará por consequencia.

Bajo esta premisa el método propuesto en este trabajo, que en el papel se basa en una idea sencilla, propone la combinación del modelo de espacio vectorial generado utilizando características básicas o alguna combinación de ellas, que caractericen a un corpus de documentos, con su modelo basado en los algoritmos de reducción de dimensionalidad, latent Dirichlet allocation LDA o latent semantic analisys LSA, con el objetivo de crear un nuevo modelo de espacio vectorial, que al utilizar como complemento los modelos LDA o LSA, mejore la clasificación del modelo complementado, y se obtenga por consecuencia un nuevo conjunto de características, cuyos resultados de clasificación sean el mejor de los casos.

Para verificar la idea anteriormente planteada, se utiliza como corpus de prueba, un subconjunto del corpus de documentos multi etiquetados Reuters-21578, construyendo con este, modelos de espacio vectorial que se utilizarán como línea base y que estarán basados

en características tradicionales y en algunos casos por combinaciones de ellas, proponiéndose en forma general el siguiente método que verifique la hipótesis propuesta.

- Inicialmente, dentro de este método, los modelos base se clasificarán para observar cuáles de ellos muestran los mejores resultados en cuanto a la medida F1.
- Posteriormente se aplicará a estos modelos algún algoritmo LDA o LSA.
- Los VSM's resultado de la aplicación de dichos algoritmos se complementarán con los modelos que les dieron origen y se clasificarán observando el resultado de su medida F1.
- Para realizar la multi clasificación del corpus se plantea la utilización del algoritmo denominado RA*k*EL, implementado en el software Meka [45], el cuál de acuerdo a la bibliografía consultada es un algoritmo que ofrece buenos resultados en multi clasificación.

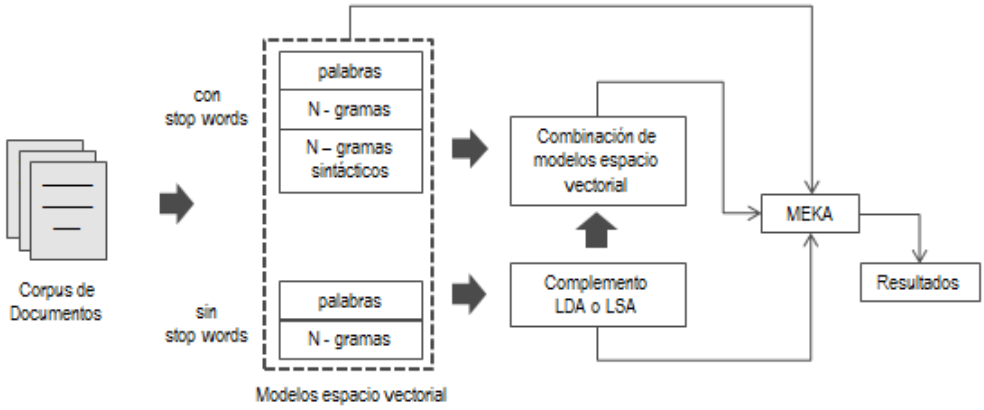


Figura 4.4. Método propuesto.

El método mencionado anteriormente se muestra en la figura 4.4, construyo este sobre 4 etapas, las cuáles se enumeran a continuación.

1. Construcción de los modelos base espacio vectorial.
2. Combinación de modelos VSM.
3. Construcción y combinación de modelos LDA y LSA.
4. Multi clasificación de los VSM's.

En los siguientes apartados se describen cada una de estas etapas.

# 4.1. Construcción de los modelos de espacio vectorial base

En esta etapa se construyen diferentes modelos espacio vectorial, considerando las características más utilizadas en diversos trabajos de clasificación que se han mencionado en el apartado 3.

Las características que se han considerado son.

- Lemas de palabras incluyendo *stop words*, para generar estas características se identifican las palabras que contienen los diferentes documentos y en el corpus, no se elimina ninguna palabra.

  Identificadas las palabras se identifican los lemas de cada una de ellas para así generar un nuevo conjunto de documentos que los contengan y un diccionario que contiene todos los lemas de palabras en el corpus.
- Lemas de palabras eliminando *stop words*, este conjunto de características se genera como el caso anterior identificando las palabras que contienen los diferentes documentos, pero en este caso se eliminan las palabras denominadas como *stop words*, la lista de estas palabras es proporcionada por el corpus Reuters-21578, posteriormente se calculan los lemas de las palabras, generándose así un nuevo conjunto de documentos y un diccionario de palabras del corpus.
- Bi-gramas que incluyen *stop words*. En este conjunto de características se identifican los enunciados que contienen los documentos.

  Posteriormente se identifican todos los bi-gramas o pares de palabras que conforman dichos enunciados, con estos pares de palabras se construye un nuevo conjunto de documentos y un diccionario de palabras del corpus.
- Bi-gramas que no incluyen *stop words*. En este conjunto de características se identifican los enunciados que contienen los documentos, eliminando aquellas palabras que se encuentran en la lista de *stop words*.

Posteriormente se identifican todos los bi-gramas o pares de palabras que conforman dichos enunciados, con estos pares de palabras se construye un nuevo conjunto de documentos y un diccionario de palabras del corpus.

- N-gramas sintácticos. Se identifican los enunciados que conforman cada uno de los documentos, utilizando estos enunciados se construyen sus correspondientes árboles sintácticos.

  Con estos árboles sintácticos, se identifican todos los bi-gramas o pares de palabras que los conforman, con estos pares de palabras se construye un nuevo conjunto de documentos y un diccionario de palabras del corpus.

Finalmente con cada uno de los conjuntos de documentos y sus respectivos diccionarios de palabras del corpus, haciendo uso del algoritmo *tf –idf* se construyen los modelos espacio vectorial identificándolos por las características que les dieron origen.

# 4.2. Combinación de modelos VSM

Esta etapa se enfoca en combinar dos modelos espacio vectorial distintos, obteniendo un nuevo modelo de espacio vectorial dónde cada fila de este nuevo modelo es la combinación de la una fila de un VSM combinada con la fila del otro VSM, dónde cada una de las filas representan al mismo documento, en forma general, esto se puede observar en la figura 4.5.

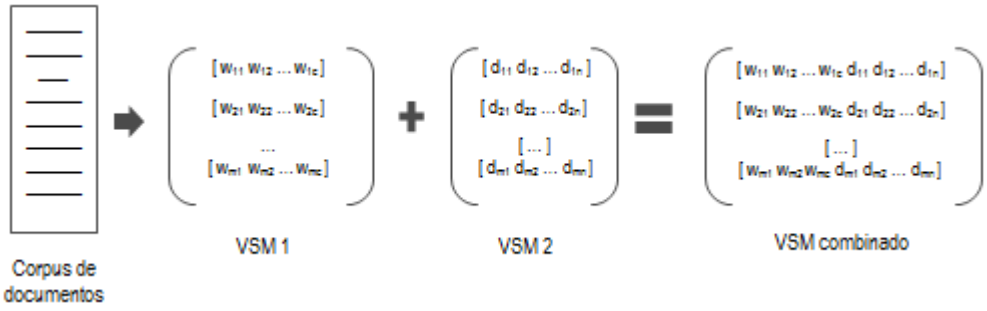


Figura 4.5. Combinación de VSM's.

Con el objetivo de asegurar dicha correspondencia, la combinación realiza a un nivel de conjuntos, es decir, considerando dos conjuntos de documentos generados a partir de distintas características, se toma un documento de un conjunto y el mismo documento de

otro conjunto, generando un nuevo documento que incluye al primer documento concatenado con el segundo documento, finalmente los diccionarios de palabras de ambos corpus de documentos se concatenarán de la misma manera que los documentos, generándose así un nuevo conjunto de documentos.

Finalmente se aplica el algoritmo *tf –idf* a este nuevo conjunto de documentos y su diccionario de palabras, construyendose el nuevo modelo de espacio vectorial identificándolo por las características que le dieron origen.

## 4.3. Construcción y combinación de modelos LDA y LSA

Utilizando cualquiera de los VSM's generados en los apartados 4.1 o 4.2, se aplica alguno de los algoritmos LDA o LSA, utilizando para su ejecución diferentes números de tópicos, generando así los correspondientes modelos espacio vectorial LDA o LSA.

En este trabajo se propone valores de tópicos que se encuentran en el siguiente rango 4, 5, 6, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550 y 600.

El número de tópicos indica el número de columnas de los modelos LDA-VSM y LDA-VSM, por lo que como se puede observar al combinar estos modelos con otros VSM el número de características no crece en forma importante.

Es importante mencionar que en cada uno de los modelos espacio vectorial generado por los algoritmos LDA o LSA y el número de tópicos, se identifican los documentos a los que corresponden cada una de las filas que en el modelo se encuentran.

Conociendo esta relación se combinan un modelo de espacio vectorial LDA o LSA con el VSM con base en el cual fue generado, a diferencia de la combinanción anteriormente explicada de dos distintos VSM's, en este caso, haciendo uso de un proceso iterativo, se toma cada una de las filas del VSM calculado con base en el algoritmo *tf –idf* y se complementa con su correspondiente fila del VSM LDA o LSA, asegurando que ambas

filas correspondan al mismo documento, obteniéndose así un nuevo modelo de espacio vectorial que contiene su correspondiente complemento LDA o LSA.

## 4.4. Clasificación de los VSM's

Finalmente, en esta etapa se realiza la clasificación de los distintos modelos espacio vectorial, obtenidos a partir de diferentes características y complementos LDA o LSA construidos en las etapas anteriores.

Es importante mencionar que este método funciona bajo un esquema supervisado, por lo que en cada uno de los modelos construidos se indica en cada una de las filas que caracterizan a los documentos se indican los tópicos a los que pertenecen estos.

Por ejemplo, en la tabla 4.1 se muestra el extracto de un vector de un documento, en el cual los últimos 3 valores indican que el documento pertenece a los tópicos 1 y 3.

Tabla 4.1. Ejemplo de clases a las que pertenece un documento.

| |
|---|
| { $V_1$ $V_2$ $V_3$ … $C_n$ **1 0 1** } |

Por lo que antes de realizar la operación de clasificación, en cada una de las filas del VSM a clasificar se le agrega un pequeño vector que indique a que clases pertenece el documento, de acuerdo al formato que el clasificador indique.

Finalmente, para realizar la clasificación multietiquetada se utiliza el algoritmo RA*k*EL1 [43], [44] implementado en el software MEKA [45], el cual se esta basado en el lenguaje de programación Java, haciendo uso del software WEKA [60] como proveedor del clasificador base, cada uno de los VSM's generados en las etapas anteriores es clasificado.

El software MEKA entrega un reporte dónde indica el valor de las medidas precision y recall, haciando uso de estas medidas se realiza el cálculo de cada una de las medidas F1 de cada uno de los VSM´s clasificados para así posteriormente ser comparadas y definir los mejores resultados de clasificación.

### 4.4.1. Metodología de clasificación

Utilizando cada uno de los reportes generados por Meka, se identifica el conjunto de características representado por un modelo espacio vectorial que ha obtenido los mejores resultados de clasificación mediante la siguiente metodología:

1. Por cada archivo de resultados se identifica:
    a. El conjunto de características que componen el VSM clasificado.
    b. Los valores de precisión y recall.
    c. Se calcula el valor de la medida F1.
2. Comparando todo el conjunto de medidas F1 se listan de mayor a menor, indicando los conjuntos de características correspondientes.
3. Se devuelve la lista de resultados.

## 4.5. Discusión

Como se observa dentro de este capítulo, la idea central detrás de trabajo, es en su concepto simple, basándo la propuesta del trabajo en considerar a los modelos LDA y LSA, como un conjunto de características que se pudiesen combinar con otros, esperando así mejorar los resultados de clasificación de los diferentes conjuntos de características, antes de ser complementados.

En el capítulo 5, con base en los resultados obtenidos se podrá observar si dicha afirmación anteriormente mencionada es correcta, de ser así, se podrá considerar a los modelos LDA y LSA como buenos complementos, como trabajo futuro en este capítulo se podrá verificar la premisa en la cual si dos documentos son similares sus correspondientes vectores LDA y LSA serán similares también y su efecto en la clasificación.

Por otro lado, como se mencionó anteriormente los modelos LDA y LSA dependen del número de tópicos que se indiquen al momento de generarlos, dicho número representa la dimensión del vector de tópicos, no obstante, agregar un modelo con un número pequeño o

grande de tópicos puede presentar mejores resultados que un modelo con un gran número grande o pequeño, i.e., no se puede recomendar un número óptimo.

Cabe mencionar que la dimensión del nuevo modelo de espacio vectorial generado con base en la combinación de algún LDA-VSM o LSA-VSM no difiere en gran medida con respecto al modelo de espacio vectorial que no contiene la combinación.

Finalmente, la metodología de clasificación propuesta se basa en comparar los diferentes resultados de las medidas F1 obtenidos al clasificar los distintos modelos espacio vectorial, realizando una búsqueda de los mejores valores, con el objetivo de identificar el mejor modelo de características propuesto.

# 5. Resultados obtenidos

En este capítulo se presentarán los resultados obtenidos a partir de la aplicación del método presentado en el capítulo 4.

## 5.1. Aplicaciones

Las aplicaciones para crear los distintos VSM's, para el cálculo de los algoritmos LDA y LSA, así como el cálculo de lemas, eliminación de stopwords, etc., estas aplicaciones fueron desarrolladas utilizando:

- Software Python v2.7, utilizado como lenguaje de programación para el desarrollo de aplicaciones.
- Librería *nltk* de Python para el cálculo de *tf, tf –idf, idf* y la eliminación de las stopwords, principalmente.
- Librería *gensim* de Python para el cálculo de los algoritmos latent Dirichlet allocation y latent semantic indexing.
- Software Core NLP de la Universidad de Stanford, utilizado para el cálculo de *n*-gramas sintácticos y los lemas de las palabras.
- Meka 1.3 como software multi clasificador.

La aplicación desarrollada es constituida por diferentes módulos, entre los que se pueden mencionar.

### 5.1.1. Caracterización de los documentos

Este módulo toma como entrada un corpus de documentos, y se encarga de:

- Eliminar *stop words*, símbolos, etc de los distintos documentos.
- Generación de los árboles sintácticos de los documentos.

- Generar nuevos corpus de documentos caracterizados en.
    - Palabras.
    - N-gramas.
    - N-gramas sintácticos.

## 5.1.2. Creación de los modelos *tf –idf*

Este módulo toma como entrada un corpus de documentos previamente caracterizado, y se encarga de:

- Crear los modelos *idf*, *tf* y finalmente *tf –idf* para cada documento.
- Crear un diccionario de palabras del corpus de documentos.

## 5.1.3. Combinación de los modelos *tf –idf*

Este módulo toma como entrada dos corpus de documentos, encargándose de realizar la combinación entre ellos.

Esta combinación se realiza archivo por archivo, por lo que se crea un nuevo corpus de documentos.

## 5.1.4. Aplicación de los algoritmos LDA y LSA

Este módulo toma como entrada un VSM y un número de tópicos, aplicando el algoritmo LDA o LSA a dicho VSM.

Como resultado se obtiene un modelo VSM que contiene el vector LDA o LSA de cada documento.

Finalmente, se contruye un nuevo archivo por documento, generándose así un nuevo corpus.

### 5.1.5. Construcción de los modelos de espacio vectorial

Este módulo se encarga de construir los distintos VSM's en un archivo con formato arff para que sea procesado por el software Meka.

Toma como entrada un corpus de documentos, con base en cada documento se construye el vector característico de este, anexando las etiquetas de las diferentes clases a las que pertenece el documento construyendo así el VSM correspondiente.

### 5.1.6. Clasificador del VSM multi etiquetado

Este módulo se encarga de tomar cada uno de los VSM y enviarlos vía comando al software Meka para su clasificación, dentro de este comando es necesario indicar el multi clasifcador con el que se va a trabajar el VSM y el clasificador base, necesario para el multi clasificador, así como los parámetros propios del multi clasificador y clasificador base.

Entrega como resultado un archivo txt que contiene entre otros valores, los de las medidas precisión y recall, necesarios para el cálculo de la medida F1.

Finalmente, con los diferentes archivos de resultados se realiza el cálculo de la F1 para cada uno de los archivos y se construye un archivo de Excel con el cual se realiza la comparación de las distintas medidas F1.

## 5.2. Corpus de prueba

Como corpus de prueba se utilizo un extracto del corpus denominado Reuters-21578, el cual se descargó utilizando la librería *nltk* de Python.

El corpus Reuters-21578 se encuentra constituido por:

- Conjunto de documentos de aprendizaje.

- Conjunto de documentos de prueba.

En este trabajo se considero parte del conjunto de documentos de aprendizaje, este conjunto se encuentra constituido por 7,769 documentos, de los cuáles 6,577 archivos que representan el 84.65 % del conjunto pertenecen solamente a una temática o tópico, i.e., cada documento se identifica únicamente con una sola etiqueta, teniendo que el 15.35 % restante del conjunto de documentos de entrenamiento que representa alrededor de 1,192 archivos se encuentra multietiquetado, i.e., cada documento pertenece a más de una temática o tópico, es decir, cada documento se identifica con más de una etiqueta.

Analizando el conjunto de documentos multietiquetados se puede observar el siguiente comportamiento.

Los 1,192 archivos con los que cuenta se encuentran clasificados en 385 multi clases de la siguiente manera.

1. Las multi clases "*interest money-fx*", "*wheat grain*", "*grain corn*" y "*money-fx dlr*", agrupan alrededor de 346 archivos, alrededor del 29 % del total de archivos.
2. 289 archivos, que representan alrededor del 24.25 % del total de archivos, pertenecen a 17 multi clases, a cada una de las cuáles le pertenecen entre 11 y 32 archivos.
3. Finalmente, 557 archivos, i.e., el 46.75 % del total de archivos se agrupan en 364 multi clases, no obstante, es importante hacer notar que 289 multi clases cuentan con sólo 1 archivo en su conjunto, 35 con 2 y el resto entre 3 y 9.

Como se puede observar el extracto de corpus de entrenamiento que contiene los documentos multietiquetados no se encuentra balanceado, es por esta razón que se escogio un subconjunto de estos documentos que permita contar con multi clases que en su conjunto se encuentren lo más balanceadas posibles, es decir, se esogio como corpus de prueba las clases del punto 1, las distribución de archivos de estas clases se muestra en la tabla 5.1.

Tabla 5.1. Distribución de archivos del corpus de prueba.

| Multi clases | Número de archivos |
|---|---|
| interest money-fx | 112 |
| wheat grain | 102 |
| corn grain | 67 |
| dlr money-fx | 65 |

## 5.3. Construcción de los modelos de espacio vectorial

Para construir los modelos de espacio vectorial cada uno de los archivos es sometido a un proceso de extracción de características, para su posterior evaluación. En este trabajo se construyen los siguientes VSM's basados en las siguientes características.

a. Lemas de palabras, con este conjunto de características se crean dos conjuntos de documentos en uno de ellos se consideran todos los lemas y en el otro se eliminan aquellos que pertenecen a las denominadas *stop words*, para ello se sigue el procedimiento que a continuación se menciona.
    a.1. Utilizando el software Stanford Parser, cada uno de los documentos pertenecientes al corpus de prueba se someten a un proceso de tokenización con el objetivo de identificar las palabras que los conforman, generando un conjunto de documentos tokenizados.
    a.2. Tomando el conjunto de documentos generados en el punto a.1, cada uno de ellos se someten de nueva cuenta al Stanford Parser para el calculo de los lemas de las palabras, a este conjunto de documentos se le denominará únicamente *unigramas*.
    a.3. Volviendo a tomar el conjunto de documentos generados en el punto a.1, se eliminan de ellos las palabras que son *stop words*, para posteriormente someterlos de nueva cuenta al Stanford Parser para el calculo de los lemas de las palabras que no fueron eliminadas, a este conjunto de documentos se le denominará *unigramas sin stop words* o *unigramas ssw*.

b. *N*-gramas, tomando los conjuntos de documentos creados en los puntos a.a.2 y a.a.3 se procede a crear los diferentes conjuntos de *n*-gramas que los conforman, es así que se crean conjuntos de documentos denominados por ejemplo, dependiendo del *n*-grama que se quiera crear y del conjunto de documentos elegido, *2gramas* si consideramos el conjunto del punto a.a.2 o *2gramas ssw* si consideramos el conjunto del punto a.a.3, el que no contiene *stop words*.
c. *N*-gramas sintácticos, tomando los documentos del corpus de pruebas, se someten los diferentes archivos al Stanford Parser con el objetivo de crear los árboles sintácticos de lo diferentes enunciados que contienen, una vez identificados estos árboles por archivo, con un programa desarrollado en Python se procede a crear los diferentes conjuntos de *n*-gramas sintácticos que los conforman, es así que se crean conjuntos de documentos denominados *2s-gramas*, *3s-gramas*, etc. Es importante mencionar que es necesario incluir las *stop words* para la creación de los diferentes árboles sintácticos.

Finalmente a los distintos conjuntos de documentos creados en los puntos anteriores:

- Punto a.a.2, *unigramas*.
- Punto a.a.3, *unigramas ssw*.
- Punto b, *n-gramas* o *n-gramas ssw*.
- Punto c, *ns-gramas*.

Se les aplica el algoritmo *tf –idf* con lo que se construyen diferentes modelos de espacio vectorial, los cuáles se denotan con el nombre de los conjuntos que los crearon, por ejemplo si el VSM esta basado en *unigramas* se domina *unigramas*.

En la tabla 5.2 se indican algunos ejemplos de VSM's creados así como el número de características en cada uno de ellos.

Tabla 5.2. Número de características de algunos VSM's creados.

| Modelo VSM | Características |
|---|---|
| unigramas ssw | 3,776 |
| unigramas | 4,067 |
| 2gramas ssw | 18,534 |

| | |
|---|---|
| 2gramas | 23,667 |
| 2s-gramas | 26,164 |

Es importante mencionar que en este trabajo se considerarán dos casos bases contra los cuáles se compararán los resultados de la propuesta de este trabajo, estos son *unigramas* y *unigramas ssw*.

Tabla 5.3. Ejemplo de un vector de características en un VSM.

{ (70 2.2380), (288 2.2380), … , (16821 0.8488), (18274 2.2380) }

En la tabla 5.3 se muestran un extracto de un vector de características para algún documento en algún VSM, en este vector el primer número corresponde al número de la característica y el segundo es su correspondiente valor *tf –idf*, cabe mencionar que aquellas características cuyo valor *tf –idf* es cero no se incluyen en el vector.

# 5.4. Combinación de los VSM's

Con ayuda de un software desarrollado en Python, se combinan algunos VSM's generados en el apartado 5.3, cabe aclarar que un VSM entre mayor sea su número de características mayor es el tiempo de procesamiento durante la fase de clasificación, algunos VSM's generados a partir de esta combinaciones junto con su número de características se muestran en la tabla 5.4, el número de características es la suma del número de características de los VSM's combinados.

Tabla 5.4. Número de características de algunos VSM's combinados.

| Modelo VSM | Características |
|---|---|
| unigramas ssw + 2gramas ssw | 22,310 |
| unigramas + 2gramas ssw | 22,601 |
| unigramas ssw + 2gramas | 27,443 |
| unigramas + 2gramas | 27,734 |
| unigramas ssw + 2s-gramas | 29,940 |
| unigramas + 2s-gramas | 30,231 |

# 5.5. Construcción y combinación de los complementos LDA y LSA

Por cada VSM generado en el apartado 5.3 y las combinaciones construidas en el inciso 5.4 se generan sus VSM's basados en los modelos LDA y LSA, para posteriormente combinarlas con los VSM's que los generaron.

Para la generación de los modelos LDA y LSA se utilizaron como números de tópicos 6, 50, 100, 500 y 1000.

El número de características de cada VSM basado en estos modelos corresponde con el número de tópicos con base en los cuáles se generaron, es decir, si para generar un modelo LDA de un VSM en particular se indicaron 6 tópicos como parámetro de entrada, el LDA-VSM generado tendrá 6 características.

En la tabla 5.5 se muestra un ejemplo de un vector LDA, en el cuál se utilizaró como parámetro de entrada como número de tópicos igual a 6, se puede observar que este documento pertenece solamente al primer tópico solamente.

Tabla 5.5. Ejemplo de un vector LDA.

{ Topic0 0.9934, Topic1 0.0, Topic2 0.0, Topic3 0.0, Topic4 0.0, Topic5 0.0 }

En la tabla 5.6 se muestran algunos ejemplos de VSM's con su respectivo número de características, que han sido complementados con sus modelos LDA.

Tabla 5.6. Algunos VSM's combinados con sus modelos LDA.

| Modelo VSM | Características |
|---|---|
| unigrams ssw | 3,776 |
| unigrams ssw + lda_6 | 3,780 |
| unigrams ssw + lda_50 | 3,826 |
| unigrams ssw + lda_100 | 3,876 |
| unigrams ssw + lda_500 | 4,276 |
| unigrams ssw + lda_1000 | 4,776 |

| Modelo VSM | Características |
|---|---|
| 2gramas ssw | 18,534 |
| 2gramas ssw + lda_6 | 18,540 |
| 2gramas ssw + lda_50 | 18,584 |
| 2gramas ssw + lda_100 | 18,634 |
| 2gramas ssw + lda_500 | 19,034 |
| 2gramas ssw + lda_1000 | 19,534 |
| unigrams ssw + 2gramas ssw | 22,310 |
| unigrams ssw + 2gramas ssw + lda_6 | 22,316 |
| unigrams ssw + 2gramas ssw + lda_50 | 22,360 |
| unigrams ssw + 2gramas ssw + lda_100 | 22,410 |
| unigrams ssw + 2gramas ssw + lda_500 | 22,810 |
| unigrams ssw + 2gramas ssw + lda_1000 | 23,310 |

Finalmente, en la tabla 5.7 se muestra un vector combinado con su modelo LDA.

Tabla 5.7. Ejemplo de un vector con su complemento LDA.

{ (70 2.2380), (288 2.2380), …, (16821 0.8488), (18274 2.2380), (18533 0.0), (18534 0.9934) }

# 5.6. Parámetros de clasificación

Para realizar la clasificación de los VSM's se configuró principalmente el software Meka con los parámetros mostrados en la tabla 5.8, en el Anexo A se muestran otros experimentos junto con los parámetros utilizados.

Tabla 5.8. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | Rakel1 |
| Clasificador base | Naïve Bayes |
| Método de valudación | k-Fold cross with k = 10 |

# 5.7. Experimentos

En esta sección se muestran los diferentes experimentos realizados, aplicando a los resultados obtenidos la metodología de clasificación desarrollada, para así finalmente identificar los mejores modelos de características clasificados y comparar diferentes valores de la medida F1 con los casos base propuestos.

## 5.7.1. Comparación entre casos

Considerando los VSM's construidos en el apartado 5.3 que incluyen a los *unigramas*, *n-gramas* y *n-gramas sintácticos* junto con los parámetros para el clasificador mostrados en el punto 5.6, se realiza la clasificación de estos VSM's obteniéndose los siguientes resultados mostrados en la tabla 5.9.

Tabla 5.9. Medidas F1 de VSM's tradicionales.

| Modelo VSM | Medida F1 |
|---|---|
| 2gramas ssw | 0.8970 |
| unigrams ssw | 0.8955 |
| unigrams | 0.8935 |
| 2gramas | 0.8800 |
| 2s-gramas | 0.8405 |

Como se menciono en el capítulo 3 de Antecedentes, tradicionalmente se considera que los resultados arrojados durante la clasificación utilizando modelos de espacio vectorial basados en unigramas ofrecen resultados competitivos, si no es que los mejores, no obstante, como se puede observar en la tabla 5.9, en este experimento el VSM mejor clasificado es el basado en bigramas que en el que no se incluyen las *stop words.*

El valor de la medida F1 de los bigramas sin *stop words* es de 0.8970, la cual mejora en 0.0015 a la arrojada por el VSM basado en unigramas sin *stop words* que es de 0.8955.

En la figura 5.1 se puede observar un comparativo gráfico de la tabla 5.9.

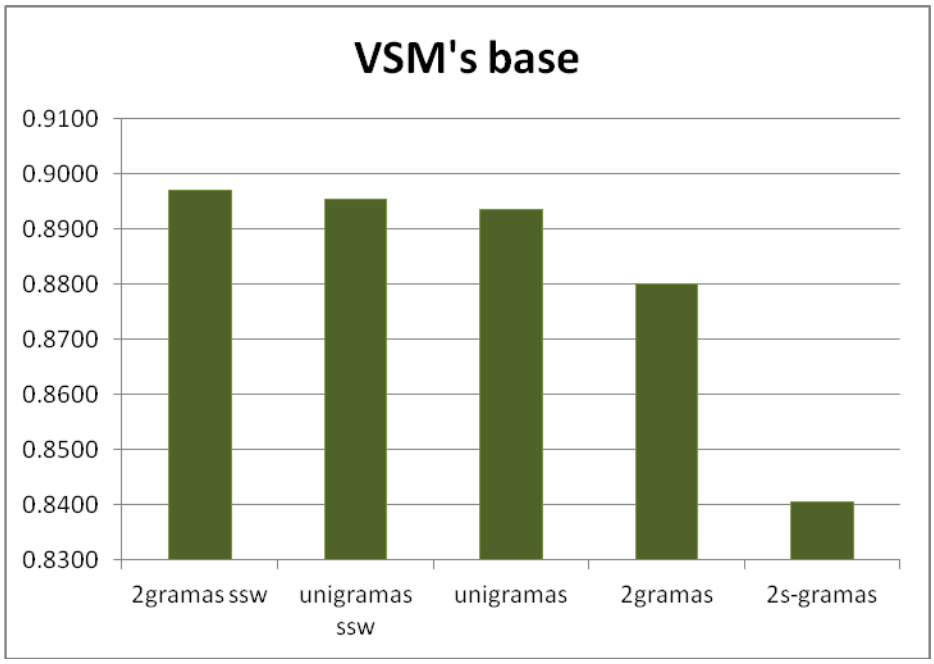


Figura 5.1. Representación gráfica de las medidas F1 de VSM's tradicionales.

En la tabla 5.10 se muestran los resultados de clasificar los VSM's basados en los modelos LDA de los modelos espacio vectorial empleados en el experimento anterior, es estas clasificaciones se utilizaron los modelos LDA puros, es decir, no se han combinado, con el objetivo de observar los resultados de clasificar estos modelos.

En la columna del centro se indican el número de tópicos utilizado para generar el modelo LDA correspondiente, por otro lado, los valores indicados de la medida F1 se listan del mejor de los casos al peor de ellos. Como se puede observar el mejor caso corresponde al LDA-VSM de los *unigramas ssw* cuyo valor es de 0.6065, el cual tiene una diferencia de 0.2905 con respecto al mejor de los casos mostrados en la tabla 5.9, cuya medida F1 es de 0.8970, por lo que estos modelos VSM –LDA por sí solos no mejoran la clasificación.

Tabla 5.10. Clasificación de los modelos LDA de los VSM's tradicionales.

| Modelo VSM –LDA | Tópicos | Medida F1 |
|---|---|---|
| unigramas ssw | 100 | 0.6065 |
| unigramas ssw | 50 | 0.5940 |
| unigramas ssw | 100 | 0.5260 |
| unigramas ssw | 6 | 0.5125 |

| Modelo VSM –LDA | Tópicos | Medida F1 |
|---|---|---|
| 2gramas ssw | 100 | 0.5025 |
| unigramas ssw | 500 | 0.4435 |
| 2gramas ssw | 50 | 0.4405 |
| 2gramas ssw | 6 | 0.4265 |
| 2gramas ssw | 500 | 0.4055 |
| 2gramas ssw | 1000 | 0.3920 |

En la tabla 5.11 se muestran los resultados de clasificar algunos VSM's creados a partir de la combinación de los modelos construidos en el apartado 5.3, como se puede observar algunas de estas combinaciones si mejoran la medida F1 de clasificación. El mejor de los casos hasta este momento es 0.8970 el cual se obtuvo utilizandoel VSM basado en 2gramas ssw, no obstante, clasificando estos nuevos modelos se observa que el mejor de los casos resulta de la clasificación del modelo que combina unigramas ssw y 2gramas ssw, con un valor F1 de 0.9180, es decir, se obtiene una mejora de 0.021.

Tabla 5.11. Clasificación de los modelos VSM's tradicionales combinados.

| VSM combinado | Medida F1 |
|---|---|
| unigramas ssw + 2gramas ssw | 0.9180 |
| unigramas ssw + 2gramas | 0.9120 |
| unigramas + 2gramas | 0.9120 |
| unigramas + 2gramas | 0.9035 |
| unigramas ssw + 2s-gramas | 0.9025 |
| unigramas + 2s-gramas | 0.8975 |

Por otro lado, si comparamos este nuevo mejor caso con el mejor caso de los casos base, es decir, con unigramas ssw, cuyo valor es 0.8955 se tiene una mejora de 0.0225.

En la figura 5.2 se muestran una comparación gráfica de los VSM's listados en la tabla 5.11.

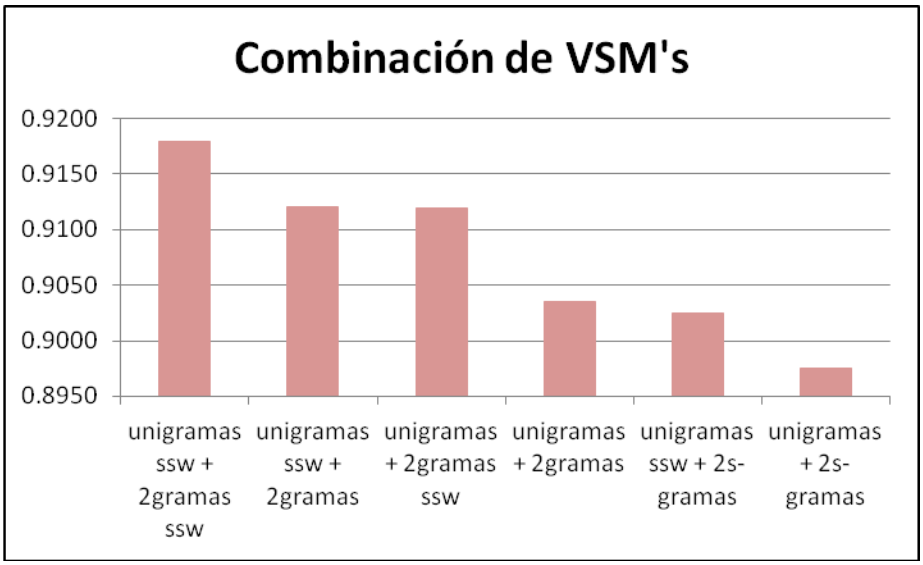


Figura 5.2. Representación gráfica de las medidas F1 de VSM's combinados.

En la tabla 5.12 se muestran algunos de los resultados de clasificar los modelos LDA generados a partir de los VSM's combinados, al igual como se observó en la tabla 5.10 no se logra una mejora en los resultados.

Tabla 5.12. Clasificación de los modelos LDA de los VSM's combinados.

| Modelo VSM –LDA | Tópicos | Medida F1 |
|---|---|---|
| unigramas ssw + 2gramas ssw | 50 | 0.5870 |
| unigramas ssw + 2gramas ssw | 100 | 0.4605 |
| unigramas ssw + 2gramas ssw | 500 | 0.4525 |
| unigramas ssw + 2gramas ssw | 6 | 0.4355 |
| unigramas ssw + 2gramas ssw | 1,000 | 0.3750 |

Finalmente, utilizando la idea propuesta en el desarrollo de esta tesis de combinar los modelos LDA-VSM o LSA-VSM con sus respectivos VSM's a partir de los cuáles se generan, se construyen nuevos modelos VSM's, y así se procede a su clasificación.

En la tabla 5.13 se muestran los mejores resultados de clasificar estos nuevos VSM's, ordenados del mejor al peor de los casos, observando que el mejor de los casos el cual es el VSM que combina unigramas ssw y 2gramas ssw, cuyo valor F1 es de 0.9180, en este caso es mejorado por el modelo VSM combinado de unigramas ssw y 2gramas ssw, combinado

con su complemento LDA a 100 tópicos, cuya medida F1 es de 0.9240, es decir, se obtiene una mejora de 0.006.

Tabla 5.13. Clasificación de los VSM's combinados con sus modelos LDA.

| Modelo | Medida F1 |
|---|---|
| unigramas ssw + 2gramas ssw + lda_100 | 0.9240 |
| unigramas ssw + 2gramas ssw + lda_6 | 0.9235 |
| unigramas ssw + lda_1000 | 0.9030 |
| unigramas ssw + lda_500 | 0.9010 |
| 2gramas ssw + lda_6 | 0.8990 |

En la figura 5.3 se muestra una comparación gráfica de los VSM's listados en la tabla 5.13.

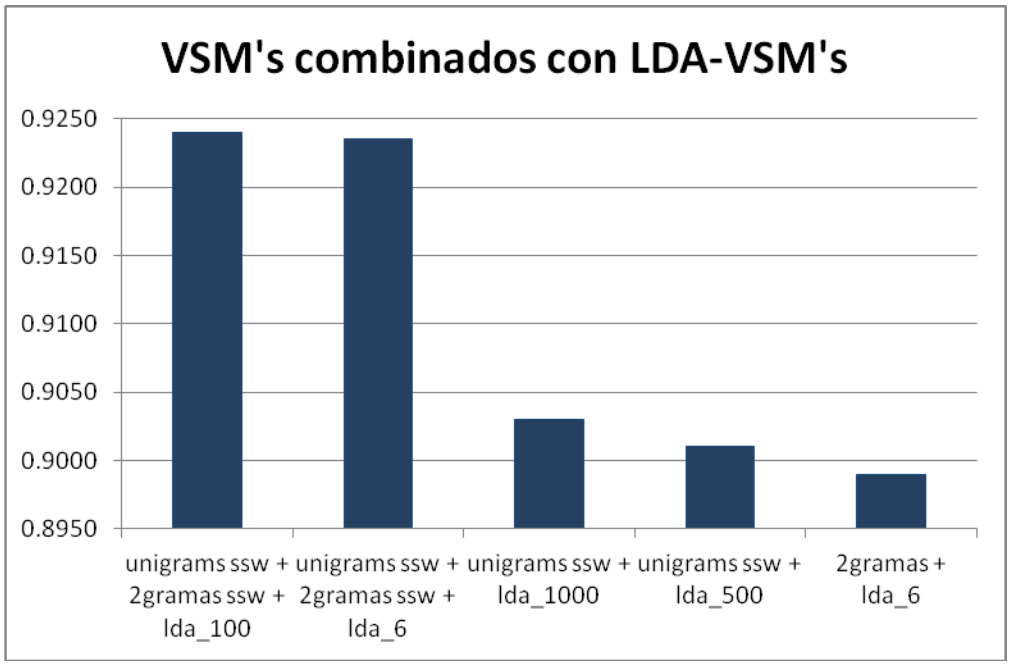


Figura 5.3. Gráfica de los resultados de VSM's y LDA-VSM's combinados.

Finalmente en la tabla 5.14 se muestra un resumen con los mejores casos de los experimentos realizados ordenados del mejor al peor de los casos, como se puede observar el mejor de los casos es una combinación de un modelo VSM generado a partir de otra combinación de unigramas ssw con bigramas ssw para finalmente combinar este VSM con su respectivo modelo LDA-VSM.

Solo uno de los dos casos base, propuestos en esta tesis se ubica en esta lista, en la posición 13, mientras que los modelos propuestos en este trabajo, basados en la combinación de un VSM con su correspodiente modelo LDA ocupan los primeros dos lugares.

Tabla 5.14. Resumen de todos los experimentos realizados.

| | Modelo | Medida F1 |
|---|---|---|
| 1 | unigramas ssw + 2gramas ssw + lda_100 | 0.9240 |
| 2 | unigramas ssw + 2gramas ssw + lda_6 | 0.9235 |
| 3 | unigramas ssw + 2gramas ssw | 0.9180 |
| 4 | unigramas ssw + 2gramas ssw | 0.9120 |
| 5 | unigramas + 2gramas ssw | 0.9120 |
| 6 | unigramas + 2gramas | 0.9035 |
| 7 | unigramas ssw + lda_1000 | 0.9030 |
| 8 | unigramas ssw + 2s-gramas | 0.9025 |
| 9 | unigramas ssw + lda_500 | 0.9010 |
| 10 | 2gramas ssw + lda_6 | 0.8990 |
| 11 | unigramas + 2s-gramas | 0.8975 |
| 12 | 2gramas | 0.8970 |
| 13 | unigramas ssw (caso base) | 0.8955 |
| 14 | unigramas | 0.8935 |
| 15 | 2gramas | 0.8800 |
| 16 | 2s-gramas | 0.8405 |

Finalmente, en la figura 5.4 se muestra un comparativo gráfico de los valores de la medida F1 mostrados en la tabla 5.14.

En el Anexo A se muestran resultados obtenidos utilizando otros multi clasificadores.

## 5.7.2. Experimentos particulares

Como se ha podido observar en la sección 5.7.1, cuando un VSM es complementado o combinado con algún complemento LDA es posible mejorar su clasificación, por lo que en los siguientes experimentos se muestran los resultados de combinar diferentes modelos de espacio vectorial, VSM con sus modelos latent Dirichlett allocation, LDA o latent semantic

indexing, LSA, dentro de estos experimentos también se realiza la combinación de un VSM no solamente con uno de modelos LDA o LSA, si no con ambos modelos, con el objetivo de observar el comportamiento de la clasificación.

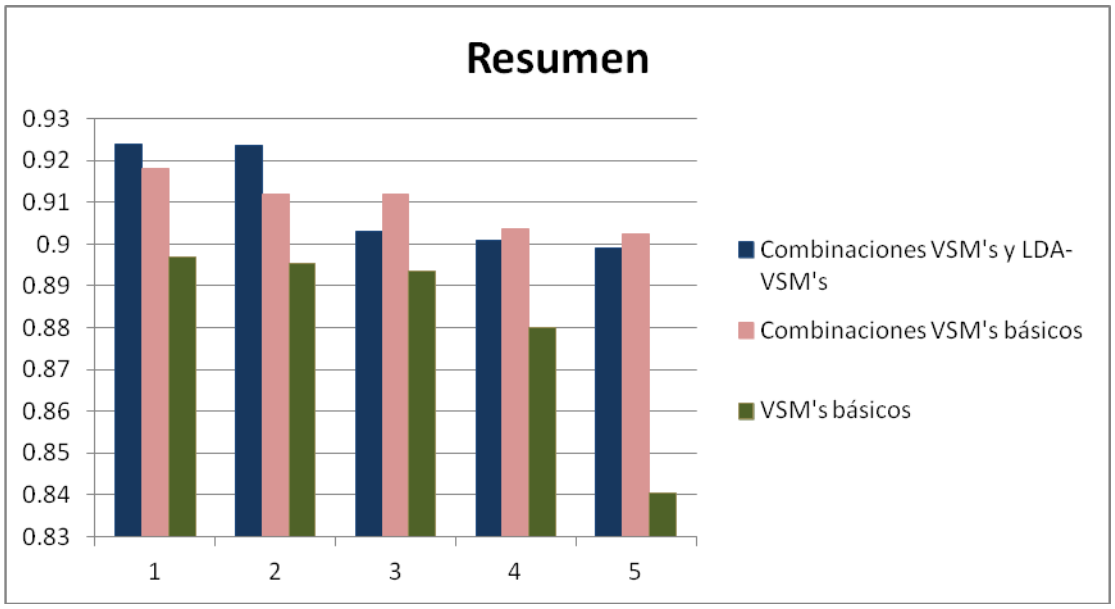


Figura 5.4. Comparativo gráfico de los mejores casos de los resultados obtenidos.

Es importante mencionar que en los experimentos que a continuación se presentarán los VSM's fueron normalizados con respecto al valor mayor en cada una de las columnas en el VSM. Además que se consideraron para su desarrollo los casos base propuestos y el mejor caso obtenido en 5.7.1.

Los modelos LDA y LSA generados se crearon considerando 4, 5, 6, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550 y 600 tópicos.

Los modelos que se construyen con base en la combinación de un VSM con sus respectivos modelos LDA y LSA se indican con el término + *ldsa*, si se utilizan únicamente el modelo LDA o LSA sel término es + *lda* o + *lsa* y finalmente si se muestra solamente el modelo LDA ó LSA sin combinar, se omite el símbolo +.

En la tabla 5.15 se muestran los resultados para el caso del VSM basado en *unigramas*, como se puede observar el caso base consigue un valor de 0.8955, pero es mejorado por la

combinación de este con los modelos LDA a 6 tópicos, identificado por *unigramas + lda_6*, con una medida F1 de 0.8970, otros modelos como son:

- *unigramas + lsa_30*,
- *unigramas + lsa_4*

Consiguen el mismo valor de medida F1 de 0.8955.

Tabla 5.15. Resultados unigramas.

| | Modelo | Medida F1 |
|---|---|---|
| 1 | unigramas + lda_6 | 0.8970 |
| 2 | unigramas + lsa_30 | 0.8955 |
| 3 | unigramas | 0.8955 |
| 4 | unigramas + lsa_4 | 0.8955 |
| 5 | unigramas + lsa_60 | 0.8950 |
| 6 | unigramas + lda_4 | 0.8950 |
| 7 | unigramas + ldsa_20 | 0.8950 |
| 8 | unigramas + ldsa_4 | 0.8950 |
| 9 | unigramas + ldsa_5 | 0.8950 |
| 10 | unigramas + lsa_5 | 0.8950 |
| 11 | unigramas lda_550 | 0.7505 |
| 12 | unigramas lsa_20 | 0.7505 |
| 13 | unigramas lda_600 | 0.7310 |
| 14 | unigramas lda_500 | 0.7055 |
| 15 | unigramas lda_250 | 0.7040 |
| 16 | unigramas lsa_40 | 0.6685 |
| 17 | unigramas lsa_30 | 0.6675 |
| 18 | unigramas lsa_10 | 0.6565 |
| 19 | unigramas lda_350 | 0.6355 |

En la tabla 5.16 se muestra el modelo de espacio vectorial basado en *unigramas ssw* en este caso la medida F1 del modelo base de 0.8915 es mejorada por diferentes modelos combinados con sus modelos LDA, LSA o LDSA. El mejor de los casos es el modelo *unigramas ssw + ldsa_6* a 6 tópicos, que arrojó una medida F1 de 0.9010.

Tabla 5.16. Resultados unigramas sin *stop words*.

| | Modelo | Medida F1 |
|---|---|---|
| 1 | unigramas ssw + ldsa_6 | 0.9010 |
| 2 | unigramas ssw + lda_500 | 0.8995 |
| 3 | unigramas ssw + lsa_30 | 0.8995 |
| 4 | unigramas ssw + lda_600 | 0.8985 |
| 5 | unigramas ssw + ldsa_20 | 0.8980 |
| 6 | unigramas ssw + ldsa_40 | 0.8980 |
| 7 | unigramas ssw + ldsa_50 | 0.8980 |
| 8 | unigramas ssw + ldsa_30 | 0.8975 |
| 9 | unigramas ssw + lda_550 | 0.8970 |
| 10 | unigramas ssw + lsa_20 | 0.8965 |
| 11 | unigramas ssw | 0.8915 |
| 12 | unigramas ssw lda_550 | 0.6815 |
| 13 | unigramas ssw lsa_20 | 0.6675 |
| 14 | unigramas ssw lsa_30 | 0.6605 |
| 15 | unigramas ssw lda_600 | 0.6585 |
| 16 | unigramas ssw lsa_40 | 0.6360 |
| 17 | unigramas ssw lsa_500 | 0.6325 |
| 18 | unigramas ssw lsa_400 | 0.6265 |
| 19 | unigramas ssw lsa_450 | 0.6265 |
| 20 | unigramas ssw lsa_550 | 0.6255 |

En la tabla 5.17 se muestra otro modelo de espacio vectorial basado en *unigramas ssw* combinados con *bigramas ssw* en este caso la medida F1 del modelo base de 0.9190, y al igual que en los casos anteriores, esta mejorada por diferentes modelos combinados con sus modelos LDA, LSA o LDSA. El mejor de los casos es el modelo *unigramas ssw + 2gramas ssw + lda_600* a 600 tópicos, que arrojó una medida F1 de 0.9010.

Tabla 5.17. Resultados unigramas sin *stop words* combinados con bigramas sin *stop words*.

| | Modelo | Medida F1 |
|---|---|---|
| 1 | unigramas ssw + 2gramas ssw + lda_600 | 0.9270 |
| 2 | unigramas ssw + 2gramas ssw + lsa_20 | 0.9240 |
| 3 | unigramas ssw + 2gramas ssw + ldsa_30 | 0.9230 |
| 4 | unigramas ssw + 2gramas ssw + lsa_30 | 0.9230 |

| | Modelo | Medida F1 |
|---|---|---|
| 5 | unigramas ssw + 2gramas ssw + lsa_6 | 0.9215 |
| 6 | unigramas ssw + 2gramas ssw + lsa_4 | 0.9210 |
| 7 | unigramas ssw + 2gramas ssw + lsa_5 | 0.9210 |
| 8 | unigramas ssw + 2gramas ssw + lda_5 | 0.9205 |
| 9 | unigramas ssw + 2gramas ssw + ldsa_40 | 0.9205 |
| 11 | unigramas ssw + 2gramas ssw + lda_450 | 0.9205 |
| 12 | unigramas ssw + 2gramas ssw | 0.9190 |
| 13 | unigramas ssw + 2gramas ssw lsa_20 | 0.8245 |
| 14 | unigramas ssw + 2gramas ssw lsa_350 | 0.7495 |
| 15 | unigramas ssw + 2gramas ssw lsa_500 | 0.7485 |
| 16 | unigramas ssw + 2gramas ssw lsa_400 | 0.7475 |
| 17 | unigramas ssw + 2gramas ssw lsa_450 | 0.7470 |
| 18 | unigramas ssw + 2gramas ssw lsa_600 | 0.7460 |
| 19 | unigramas ssw + 2gramas ssw lsa_30 | 0.7370 |
| 20 | unigramas ssw + 2gramas ssw lsa_300 | 0.7370 |
| 21 | unigramas ssw + 2gramas ssw lsa_550 | 0.7360 |

En la tabla 5.18 se muestra un VSM basado en la combinación de *unigramas* con *bigramas ssw*, el cual obtiene una medida F1 de 0.9135, de igual forma este valor es superado por diferentes modelos combinados con sus modelos LDA, LSA o LDSA. El mejor de los casos es el modelo *unigramas + 2gramas ssw + ldsa_50* a 50 tópicos, que arrojó una medida F1 de 0.9215.

Tabla 5.18. Resultados unigramas combinados con bigramas sin *stop words*.

| | Modelo | Medida F1 |
|---|---|---|
| 1 | unigramas + 2gramas ssw + ldsa_50 | 0.9215 |
| 2 | unigramas + 2gramas ssw + ldsa_60 | 0.9215 |
| 3 | unigramas + 2gramas ssw + lda_350 | 0.9205 |
| 4 | unigramas + 2gramas ssw + lsa_30 | 0.9185 |
| 5 | unigramas + 2gramas ssw + ldsa_80 | 0.9180 |
| 6 | unigramas + 2gramas ssw + ldsa_40 | 0.9175 |
| 7 | unigramas + 2gramas ssw + ldsa_70 | 0.9165 |
| 8 | unigramas + 2gramas ssw + lda_400 | 0.9145 |
| 9 | unigramas + 2gramas ssw + lda_6 | 0.9145 |

| | Modelo | Medida F1 |
|---|---|---|
| 11 | unigramas + 2gramas ssw + ldsa_30 | 0.9145 |
| 12 | unigramas + 2gramas ssw | 0.9135 |
| 13 | unigramas + 2gramas ssw lsa_20 | 0.7870 |
| 14 | unigramas + 2gramas ssw lsa_550 | 0.7620 |
| 15 | unigramas + 2gramas ssw lsa_600 | 0.7530 |
| 16 | unigramas + 2gramas ssw lsa_350 | 0.7505 |
| 17 | unigramas + 2gramas ssw lsa_500 | 0.7480 |
| 18 | unigramas + 2gramas ssw lsa_450 | 0.7425 |
| 19 | unigramas + 2gramas ssw lsa_400 | 0.7355 |
| 20 | unigramas + 2gramas ssw lsa_300 | 0.7320 |
| 21 | unigramas + 2gramas ssw lsa_200 | 0.7115 |

En la tabla 5.19 el VSM mostrado es basado en *unigramas ssw* combinados con *bigramas* en este caso la medida F1 del modelo base de 0.9110, y se puede observar una mejora utilizando diferentes combinaciones LDA, LSA o LDSA. El mejor de los casos es el modelo combinado con el modelo ldsa a 60 tópicos, que arrojó una medida F1 de 0.9185.

Tabla 5.19. Resultados unigramas sin *stop words* combinados con bigramas.

| | Modelo | Medida F1 |
|---|---|---|
| 1 | unigramas ssw + 2gramas + ldsa_60 | 0.9185 |
| 2 | unigramas ssw + 2gramas + ldsa_70 | 0.9175 |
| 3 | unigramas ssw + 2gramas + ldsa_50 | 0.9175 |
| 4 | unigramas ssw + 2gramas + lsa_30 | 0.9145 |
| 5 | unigramas ssw + 2gramas + ldsa_40 | 0.9130 |
| 6 | unigramas ssw + 2gramas + lsa_60 | 0.9130 |
| 7 | unigramas ssw + 2gramas + ldsa_80 | 0.9120 |
| 9 | unigramas ssw + 2gramas + ldsa_30 | 0.9115 |
| 11 | unigramas ssw + 2gramas + lsa_5 | 0.9115 |
| 12 | unigramas ssw + 2gramas | 0.9110 |
| 13 | unigramas ssw + 2gramas lsa_600 | 0.7695 |
| 14 | unigramas ssw + 2gramas lsa_350 | 0.7670 |
| 15 | unigramas ssw + 2gramas lsa_550 | 0.7670 |
| 16 | unigramas ssw + 2gramas lsa_300 | 0.7580 |
| 17 | unigramas ssw + 2gramas lsa_500 | 0.7550 |
| 18 | unigramas ssw + 2gramas lsa_450 | 0.7460 |

| | Modelo | Medida F1 |
|---|---|---|
| 19 | unigramas ssw + 2gramas lsa_400 | 0.7445 |
| 20 | unigramas ssw + 2gramas lsa_250 | 0.6990 |
| 21 | unigramas ssw + 2gramas lsa_20 | 0.6765 |

En la tabla 5.20 el VSM mostrado es basado en *unigramas* combinados con *bigramas* en este caso la medida F1 del modelo base de 0.9035, y al igual que en todos los casos anteriores puede observar una mejora utilizando diferentes combinaciones. El mejor de los casos es el modelo combinado con el modelo lsa a 90 tópicos, que arrojó una medida F1 de 0.9150.

Tabla 5.20. Resultados unigramas combinados con bigramas.

| | Model | Medida F1 |
|---|---|---|
| 1 | unigramas + 2gramas + lsa_90 | 0.9150 |
| 2 | unigramas + 2gramas + ldsa_50 | 0.9150 |
| 3 | unigramas + 2gramas + ldsa_70 | 0.9135 |
| 4 | unigramas + 2gramas + ldsa_40 | 0.9125 |
| 5 | unigramas + 2gramas + ldsa_90 | 0.9125 |
| 6 | unigramas + 2gramas + lsa_60 | 0.9125 |
| 7 | unigramas + 2gramas + ldsa_60 | 0.9120 |
| 8 | unigramas + 2gramas + lsa_100 | 0.9120 |
| 9 | unigramas + 2gramas + lsa_70 | 0.9110 |
| 11 | unigramas + 2gramas + lsa_30 | 0.9105 |
| 12 | unigramas + 2gramas | 0.9035 |
| 13 | unigramas + 2gramas lsa_550 | 0.7490 |
| 14 | unigramas + 2gramas lsa_400 | 0.7455 |
| 15 | unigramas + 2gramas lsa_500 | 0.7455 |
| 16 | unigramas + 2gramas lsa_450 | 0.7385 |
| 17 | unigramas + 2gramas lsa_350 | 0.7325 |
| 18 | unigramas + 2gramas lsa_600 | 0.7325 |
| 19 | unigramas + 2gramas lsa_300 | 0.7195 |
| 20 | unigramas + 2gramas lsa_250 | 0.6720 |
| 21 | unigramas + 2gramas lsa_150 | 0.6695 |

## 5.8. Discusión

Como se ha observado en este capítulo, en los experimentos realizados los casos base propuestos, que utilizan caracteritsicas básicas, no han sido aquellos que han obtenido los mejores resultados de clasificación, si no que como se ha presentado en el apartado de trabajos relacionados, los modelos de espacio vectorial combinados fueron los mejor clasificados.

No obstante, como se observa al aplicar el método propuesto, es posible mejorar los resultados de clasificación para el VSM que se combine con alguno de sus modelos LDA, LSA o LDSA.

Como consecuencia de lo anterior, se puede inferir que al combinar el mejor de los casos con alguno de sus modelos LDA, LSA o LDSA se obtendrá un VSM que al clasificarlo supere la clasificación de su antecesor.

Por otro lado, no se ha podido inferir con estos experimentos un número óptimo de tópicos, ya que como se puede ver, un número pequeño de los mismos puede mejorar la clasificación, pero aún un número alto también, no obstante, con un trabajo de análisis más exhaustivo es posible encontrar una relación.

También se ha podido comprobar que los algoritmos LDA y LSA ofrecen pobres resultados de clasificación en el ámbito de la multi clasificación por lo que es necesario adaptarlos a este contexto.

Finalmente, dentro de los distintos experimentos que se observan en el Anexo A se puede identificar uno que fue desarrollado utilizando una máquina de soporte vectorial como clasificador base con RA*k*el como multi clasificador, utilizando un *PolyKernel* como kernel del clasificador, es importante mencionar que se desarrollaron otros experimentos con kernels como son *NormalizedPolyKernel*, *PUK* y *RBFKernel* pero en todos se observa un comportamiento similar, en el cual se verifica que al utilizar una SVM como clasificador base se obtienen pobres resultados, dónde los modelos LDA y LSA ofrecen los mejores resultados de clasificación, aunque, al complementar los modelos de características base

con sus complementos LDA y/o LSA, si se obtiene una mejora en los resultados de clasificación, sin embargo, en ambos casos los resultados son muy bajos en comparación con aquellos experimentos en los cuáles se utiliza Naïve Bayes, como se ha mencionado anteriormente este mismo comportamiento se observa utilizando otros tipos de kernel en el clasificador base.

Aunque un análisis más exhaustivo en los parámetros propios de la SVM podrían mejorar dichos resultados, por lo que el análisis de estos experimentos dan origen a un trabajo futuro.

# 6. Conclusiones, trabajo futuro y aportaciones finales

## 6.1. Conclusiones

Como se observa en el desarrollo de este trabajo, el modelo espacio vectorial basado en conjunto de características, complementado con su modelo LDA o LSA, en el cual se basa la propuesta de esta tesis presenta mejores resultados de clasificación que el VSM en el cual se basa dentro de un entorno multi etiquetado, esto se verifica con los diversos experimentos realizados dentro de los cuales se proponen diversos VSM's basados en diferentes características, cuya medida de clasificación F1 es mejorada al construir el modelo propuesto, lo cual demuestra la hipótesis propuesta.

Se han identificado en el estado del arte diversos conjuntos de características, principalmente utilizados para representar un corpus de documentos, los cuáles se basan principalmente en la utilización de palabras o bien N-gramas, aplicando el algoritmo *tf –idf* para la construcción de los respectivos modelos espacio vectorial, por lo que en este trabajo se desarrollaron diversas aplicaciones que permitieron construir estos modelos, considerando principalmente el modelo basado en palabras como el caso base con el cual comparar los resultados de clasificación del modelo propuesto.

De acuerdo a los trabajos consultados los algoritmos LDA y LSA se utilizan principalmente para realizar una reducción dimensional de un VSM, creando un nuevo VSM en el cual se identifican grupos de tópicos en los cuáles se agrupan los diferentes vectores de los documentos que conforman al corpus, que al ser clasificado en un entorno uniclase ofrecen buenos resultados, aunque en un entorno multi clasificado no se obtiene el mismo comportamiento, sin embargo, en este último entorno ofrecen buena una alternativa como complemento semántico que permite mejorar la clasificación, lo cual se observa en los diferentes experimentos realizados, al mejorar los VSM's complementados los resultados

de clasificación con respecto a los VSM‘s sin complemento. Se consideran semánticos ya que los algoritmos LDA y LSA identifican relaciones entre las palabras para que estas puedan formar un concepto.

Dentro de las aplicaciones desarrolladas con el objetivo de complementar los distintos VSM’s es importante relacionar los vectores que las componen identificando los documentos a los que pertenecen para así evitar errores, principalmente al complementar vectores en forma errónea.

En este trabajo se ha utilizado el corpus Reuters-21578, ya que a pesar de ser un corpus no balanceado, fue posible elegir un sub grupo de tópicos balanceado, por otro lado los archivos de este corpus no contienen etiquetas y provee un conjunto de *stop words*.

La medida F1 fue elegida a ser utilizada en el desarrollo de este trabajo, ya que es una medida que incluye a las medidas precisión y recall, las cuáles, incluyendo a la medida F1 son de las más utilizadas dentro de los distintos trabajos consultados.

Finalmente, la dimensión del complemento propuesto depende del número de tópicos con base en el cual es creado, aunque, de acuerdo a los experimentos realizados no es posible precisar un número óptimo, ya que este varía entre un número pequeño o alto en diferentes conjuntos de características.

## 6.2. Aportaciones científicas obtenidas

Las aportaciones de este trabajo se enumeran a continuación.

- Un modelo de características basado en la combinación de un conjunto de características con el resultado de aplicarles algún algoritmo de reducción espacial, como puede ser LDA y LSA, que ofrece buenos resultados al realizar el proceso de clasificación.
- Una propuesta para el uso de algoritmos de agrupamiento o reducción tipo LDA y LSA aplicados en la multi clasificación como complementos semánticos.

- Una metodología basada en una heurística que permite identificar el mejor modelo de características clasificadas.

## 6.3. Trabajo Futuro

El trabajo futuro que se desprende de este trabajo se presenta a continuación, como se observa en los experimentos no se propone un número idóneo de tópicos para los algoritmos LDA y LSA, ya que en este trabajo se propone un rango de tópicos para generar los modelos, observando que alguno de ellos permite mejorar la clasificación, pero no se cuenta con una metodología propicia que lo proponga.

Por otro lado también se pueden utilizar los modelos basados en LDA y LSA, como es L-LDA, esperando de igual manera mejorar los resultados de clasificación al ser modelos orientados a la multi clasificación, ya que los modelos LDA y LSA son reductores dimensionales, principalmente.

También se hace importante conocer y fundamentar el porque los modelos LDA y LSA logran el efecto de mejorar la clasificación, para así poder construir modelos basados en esos enfoques.

Otro trabajo propuesto, es construir una metodología que permita conocer conceptualmente los tópicos que arrojan LDA y LSA considerando el conjunto de palabras que los conforman.

Un trabajo de investigación futuro, es definir la relación de un VSM combinado con algún modelo LDA, LSA o LDSA para así implementar un esquema basado en umbrales, de acuerdo a los resultados arrojados por estos últimos modelos, y poder así buscar una mejora mayor.

Finalmente, se propone el análisis más a detalle de los parámetros de *support vector machine*, SVM, para su uso como algoritmo de clasificación base en algoritmos de multi clasificación que permita el mejoramiento de los resultados al momento de su utilización en

diferentes corpus de documentos, así también, se propone la utilización de otros algoritmos de clasificación como clasificadores base y el análisis de sus resultados.

# 7. Referencias


[1] Joachims T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Proceedings 10th European Conference on Machine Learning, Springer Verlag (1998).

[2] Sidorov G., Velasquez F., Stamatatos E., Gelbukh A. and Chanona-Hernández L.: "Syntactic Dependency-based N-grams as Classification Features". LNAI 7630, 2012, pp. 1–11. DOI: 10.1007/978-3-642-37798-3_1.

[3] Sidorov G.: "Non-continuous Syntactic N-grams". Polibits, vol. 48, pp. 67–75, 2013.

[4] Yang Y. and Pedersen J.O.: "A Comparative Study on Feature Selection in Text Categorization". In Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97. CA, USA (1997).

[5] Sidorov G.: "Contrucción no-lineal de n-gramas en la lingüística computacional". Sociedad Mexicana de Inteligencia Aritificial, México, 2013.

[6] Salton G. and McGill M.: "Introduction to Modern Information Retrieval", McGraw Hill, 1983.

[7] Salton G.: "Automatic text processing", Addison-Wesley Longman Publishing Co. Inc., Boston, MA., USA., 1988.

[8] Sidorov G., Gelbukh A., Gómez-Adorno H. and Pinto D.: "Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model", Computación y Sistemas 18(3), 2014.

[9] Manning C. and Schütze H.: "Foundations of Statistical Natural Language Processing". MIT Press, Cambridge, MA (1999).

[10] Blei D.M., Ng A.Y. and Jordan M.I.: "Latent Dirichlet Allocation", Journal of Machine Learning Research 3, 2003.

[11] Ramage D., Hall D., Nallapati R. and Manning C.D.: "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora", Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 2009.

[12] Rosario B.: “Latent Semantic Indexing: An overview”, INFOSYS 240 Spring 2000. DOI: 10.1.1.94.2661.

[13] Dumais S.T., Furnas G.W., Landauer T.K., Deerwester S. and Harshman R.: “Using Latent Semantic Analysis To Improve Access To Textual Information”, SIGCHI Conference on Human Factors in Computing Systems, pp. 281-285, ACM, 1998, DOI: 10.1.1.51.5563.

[14] Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K. and Harshman R.: “Indexing by latent semantic analysis”, Journal of the American Society for Iinformation Science, vol. 41(6), pp. 391 – 407, 1990, DOI:10.1.1.108.8490.

[15] Forsythe G.E., Malcom M.A. and Moler C.B.: “Computer Methods for Mathematical Computations” (Chapter 9: Least squares and the singular value decomposition). Englewood Cliffs, NJ. Prentice Hall, 1977.

[16] Cilibrasi R.L. and Vitányi M.B.P.: “The Google Similarity Distance”, IEEE Transactions on knowledge and data engineering, Vol. 19, No. 3, (2007), DOI: 10.1109/TKDE.2007.48.

[17] Sebastiani F.: “Text Categorization”, In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK, 2005.

[18] Zhang M.-L. and Zhou Z.-L.: “A Review on Multi-Label Learning Algorithms”, IEEE Transactions on Knowledge and Data Engineering, vol. 26(8), pps. 1819 – 1837, 2014, DOI:10.1109/TKDE.2013.39.

[19] Yang H. and Callan J.: “Near-Duplicate Detection by Instance-level Constrained Clustering”. In Proc. of the 29th Conference on research and development in IR, 2006.

[20] Henzinger M.: “Finding Near-Duplicate Web Pages: a Large-Scale Evaluation of Algorithms”. In Proc. of the 29th Conference on research and development in IR, 2006.

[21] Stein B. and Meyer zu Eiben S.: “Near Similarity Search and Plagiarism Analysis”. In From Data and Information Analysis to Knowledge Engineering. Springer, 2006.

[22] Tsoumakas G., Katakis I. and Vlahavas I.: “Mining Multi-label Data”, In Data Mining and Knowledge Discovery Handbook, Springer, Rokach L., Maimon O. editors, 2010, DOI= 10.1007/978-0-387-09823-4_34.

[23] Boutell M.R., Luo J., Shen X. and Brown C.M.: “Learning multi-label scene classification”, Pattern Recognition, vol. 37(9), 2004, DOI=10.1.1.113.3921.

[24] Diplaris S., Tsoumakas G., Pericles A.M. and Vlahavas I.: "Protein Classification with Multiple Algorithms", 10th Panhelllenic Conference on Informatics, P. Bozanis and E.N. Houstis (Eds.), Spring-Verlag, LNCS 3746, pps. 448 – 456, 2005, DOI=10.1.1.100.2868.

[25] Goncalves T. and Quaresma P.: "A Preliminary Approach to the Multilabel Classification Problem of Portuguese Juridical Documents", Proceedings of the 11th Portuguese Conference on Artificial Intelligence, 2003.

[26] Lauser B. and Hotho A.: "Automatic multi-label subject indexing in multilingual environment", Proceedings of the 7th European Conference in Research and Advanced Technology for Digital Libraries, 2003.

[27] Li T. and Ogihara M.: "Detecting emotion in music", Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA, 2003.

[28] Clare A. and King R.D.: "Knowledge Discovery in Multi-Label Phenotype Data", Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery, Freiburg, Germany, 2001.

[29] Schapire R.E. and Singer Y.: "Boostexter: a boosting-based system for text categorization", Machine Learning, vol. 39(2/3), pps. 135-168, 2000.

[30] Zhang M.-L. and Zhou Z.-L.: "A k-Nearest Neighbor Based Algorithm for Multi-label Classification", Proceedings of the 1st IEEE International Conference on Granular Computing, 2005.

[31] Freund Y. and Schapire R.E.: "A decision-theoretic generalization of on-line learning and an application to boosting", Journal of Computer and System Sciences, vol. 55(1), pps. 119 – 139, 1997.

[32] McCallum A.: "Multi-label text classification with a mixture model trained by EM", Proceedings of the AAAI'99 Workshop on Text Learning, 1999.

[33] Elisseeff A. and Weston J.: "A kernel method for multi-labelled classification". Advances in Neural Information Processing Systems 14, 2002.

[34] Godbole S. and Sarawagi S.: "Discriminative Methods for Multi-labeled Classification", Proceedings of the 8th Pacific-Asia Conferenceon Knowledge Discovery and Data Mining, 2004.

[35] Dumais S., Platt J., Heckerman D. and Sahami M.: "Inductive learning algorithms and representations for text categorization", Proceedings of the seventh international conference on Information and knowledge management, pps. 148-155, ACM New York, USA, 1998, DOI:10.1145/288627.288651

[36] Bramer M.A.: "Principles of Data Mining", Undergraduate Topics in Computer Science, Second Edition, Springer 2013, DOI:10.1007/978-1-4471-4884-5_4

[37] Heckerman D.: "A Tutorial on Learning With Bayesian Networks", Learning in Graphical Models, Microsoft Research, 1996, DOI:10.1.1.15.4522

[38] Vapnik V.N.: "The Nature of Statistical Learning Theory", Springer, Nueva York, 1995.

[39] Godbole S. and Sarawagi S.: "Discriminative methods for multi-labeled classification", Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2004, DOI: 10.1007/978-3-540-24775-3_5

[40] Turner M. D., Chakrabarti C., Jones T. B., Xu J. F., Fox P. T., Luger G. F., Laird A.R. and Turner J. A.: "Automated annotation of functional imaging experiments via multi-label classification". Frontiers in Neuroscience 7(240), 2013. DOI=10.3389/fnins.2013.00240

[41] Dharmadhikari S.C., Ingle M. and Kulkarni P.: "A comparative analysis of supervised multi-label text classification methods". IJERA 1(4), pps. 1952-1961, 2011.

[42] Madjarov G., Kocev D., Gjorgjevikj D. and Džeroski S.: "An extensive experimental comparison of methods for multi-label learning". Pattern Recognition 45, 2012, DOI=3084–3104 10.1016/j.patcog.2012.03.004

[43] Tsoumakas G. and Vlahavas I.: "Random *k*-Labelsets: An Ensemble Method for Multilabel Classification". In Proceedings of the 18th European conference on Machine Learning (ECML '07), Springer-Verlag, Berlin, Heidelberg, 406-417. 2007. DOI=10.1007/978-3-540-74958-5_38

[44] Tsoumakas G., Katakis I. and Vlahavas L.: "Random k-Labelsets for Multilabel Classification", IEEE Transactions on Knowledge and Data Engineering 23(7), pp.1079 - 1089, July 2011. DOI=10.1109/TKDE.2010.164

[45] Meka: A Multi-label Extension to Weka. http://meka.sourceforge.net/

[46] Della Pietra, S., Della Pietra, V. and Lafferty J.: "Inducing features of random fields". IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4), 380 – 397, 1997.

[47] Brychcín T. and Král P.: "Novel Unsupervised Features for Czech Multi-label Document Classification", 13th Mexican International Conference on Artificial Intelligence, MICAI, 2014, DOI=10.1007/978-3-319-13647-9_8.

[48] Lund K. and Burgess C.: "Producing high-dimensional semantic spaces from lexical co-occurrence", Behavior Research Methods Instruments and Computers 28(2), 203-208, 1996.

[49] Rohde D.L.T., Gonnerman L.M. and Plaut D.C.: "An improved method for deriving word meaning from lexical co-ocurrence", Cognitive Psychology 7, pps. 573–605, 2004.

[50] McCallum A.K.: Mallet: A machine learning for language toolkit (2002), http://mallet.cs.umass.edu

[51] Kazawa H., Izumitani T., Taira H. and Maeda E.: "Maximal margin labeling for multi-topic text categorization", Advances in Neural Information Processing Systems 17, pps. 649–656, MIT Press, 2005, DOI=10.1.1.85.9942

[52] Ueda N. and Saito K.: "Parametric mixture models for multi-labeled text", In Advances in Neural Information Processing Systems 15, MIT Press, pps. 721–728, 2003, DOI=10.1.1.67.5909

[53] Kanj S., Abdallah F. and Denoeux T.: "Evidential Multi-label Classification Using the Random k-Label Sets Approach". Advances in Intelligent and Soft Computing, pps. 21-28, Springer, 2012. DOI=10.1007/978-3-642-29461-7_2

[54] Fürnkranz J., Hüllermeier E., Mencia E.L. and Brinker K.: "Multilabel classification via calibrated label ranking", Journal Machine Learning 73(2), pps. 133 - 153, November 2008, Kluwer Academic Publishers Hingham, MA, USA, DOI=10.1007/s10994-008-5064-8

[55] Zhang M-L. and Zhou Z-H.: "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization", Knowledge and Data Engineering, IEEE Transactions on 18(10), pps. 1338-1351, Octubre 2006, DOI=10.1109/TKDE.2006.162

[56] The Stanford Parser: A statistical parser. http://nlp.stanford.edu/software/lex-parser.shtml

[57] Keikha M., Khonsari A. and Oroumchian F.: "Rich document representation and classification: An analysis", Knowledge-Based Systems 22(1), pps. 67-71, Enero 2009, DOI=10.1016/j.knosys.2008.06.002.

[58] Zhang W., Yoshida T. and Tang X.: "Text classification based on multi-word with support vector machine", Knowledge-Based Systems 21(8), pps. 879-886, Diciembre 2008, DOI=10.1016/j.knosys.2008.03.044.

[59] Shang C., Li M., Feng S., Jiang Q. and Fan J.: "Feature selection via maximizing global information gain for text classification", Knowledge-Based Systems 54, pps. 298-309, Diciembre 2013, DOI=10.1016/j.knosys.2013.09.019.

[60] WEKA: Data Mining Software. http://www.cs.waikato.ac.nz/ml/weka/

# Anexo A

En el presente Anexo se presentan diversos experimentos realizados utilizando otros algoritmos de clasificación, así como diferentes modelos de espacio vectorial basados en diferentes características y combinaciones.

Los clasificadores utilizados se basan en diferentes algoritmos que incluyen al algoritmo RA*k*EL utilizando como clasificadores base Naïve Bayes y ZeroR, teniendo como objetivo mostrar el comportamiento de los resultados de clasificación, en los cuáles RA*k*EL mejora la clasificación.

# Experimentos

## Experimento A.1

En este experimento se utilizo un multi clasificador basado en relevancia binaria y como clasificador base se utilizo un naïve bayes, como se observa este clasificador se acerca a los mejores resultados obtenidos con el clasificador RA*k*EL.

De igual forma el conjunto de características mejor clasificadas se refieren a la combinación del conjunto de unigramas ssw combinadas con bigramas ssw y a su vez todo este conjunto con el modelo LDA 6 y 1000, sin embargo, en este caso no se obtiene una mejora del conjunto de datos de unigramas combinados con bigramas ssw, por lo que este conjunto es el mejor clasificado de su tipo.

El concentrado de resultados se puede observar en la tabla A.1.1, en la tabla A.1 se muestran los parámetros utilizados en el clasificador.

Tabla A.1. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | Relevancia Binaria |
| Clasificador base | Naïve Bayes |
| Método de valudación | k-Fold cross with k = 10 |

Tabla A.1.1. Concentrado de resultados del experimento A.1.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 2gramas ssw + lda_6 | 0.898 | 0.915 | 0.9064 |
| unigramas ssw + 2gramas ssw + lda_1000 | 0.897 | 0.915 | 0.9059 |
| unigramas ssw + 2gramas ssw | 0.896 | 0.912 | 0.9039 |
| unigramas + 2gramas ssw | 0.896 | 0.908 | 0.9020 |
| unigramas ssw + 2gramas ssw + lda_500 | 0.892 | 0.911 | 0.9014 |
| unigramas ssw + 2gramas | 0.886 | 0.906 | 0.8959 |
| unigramas ssw + lda_500 | 0.885 | 0.899 | 0.8919 |
| unigramas ssw + lda_1000 | 0.887 | 0.896 | 0.8915 |
| unigramas ssw + 2gramas ssw | 0.874 | 0.906 | 0.8897 |
| unigramas ssw + lda_1500 | 0.886 | 0.893 | 0.8895 |
| unigramas ssw + lda_6 | 0.886 | 0.892 | 0.8890 |
| unigramas ssw + 2gramas ssw + lda_1000 | 0.873 | 0.905 | 0.8887 |
| unigramas ssw + 2gramas ssw + lda_100 | 0.86 | 0.919 | 0.8885 |
| unigramas ssw + 2gramas ssw + lda_6 | 0.874 | 0.903 | 0.8883 |
| unigramas + 2gramas | 0.88 | 0.896 | 0.8879 |
| unigramas + 2gramas ssw | 0.876 | 0.897 | 0.8864 |
| unigramas + lda_6 | 0.886 | 0.886 | 0.8860 |
| unigramas + lda_4 | 0.885 | 0.887 | 0.8860 |
| unigramas + lda_500 | 0.879 | 0.893 | 0.8859 |
| unigramas ssw + 2gramas sintácticos | 0.872 | 0.9 | 0.8858 |
| unigramas + lda_1500 | 0.881 | 0.89 | 0.8855 |
| unigramas | 0.884 | 0.886 | 0.8850 |
| unigramas ssw | 0.883 | 0.887 | 0.8850 |
| unigramas + lsa_50 | 0.878 | 0.892 | 0.8849 |
| unigramas ssw + 2gramas ssw + lda_500 | 0.868 | 0.902 | 0.8847 |
| unigramas ssw + lda_10 | 0.882 | 0.887 | 0.8845 |
| unigramas ssw + 3gramas sintactos | 0.881 | 0.887 | 0.8840 |
| unigramas ssw + 3gramas ssw | 0.881 | 0.887 | 0.8840 |
| unigramas + lda_1000 | 0.881 | 0.887 | 0.8840 |
| unigramas ssw + lsa_50_0f | 0.87 | 0.898 | 0.8838 |
| unigramas ssw + 4gramas | 0.881 | 0.886 | 0.8835 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 4gramas | 0.881 | 0.886 | 0.8835 |
| unigramas ssw + lda_4 | 0.881 | 0.886 | 0.8835 |
| unigramas ssw + lsa_6 | 0.88 | 0.887 | 0.8835 |
| unigramas ssw + 3gramas | 0.878 | 0.889 | 0.8835 |
| unigramas + lsa_6 | 0.881 | 0.885 | 0.8830 |
| unigramas + 3gramas sintácticos | 0.88 | 0.886 | 0.8830 |
| unigramas + 3gramas ssw | 0.88 | 0.886 | 0.8830 |
| unigramas ssw + 5gramas | 0.879 | 0.886 | 0.8825 |
| unigramas ssw + lsa_10 | 0.878 | 0.887 | 0.8825 |
| unigramas ssw + lda_300 | 0.872 | 0.893 | 0.8824 |
| unigramas + 2gramas sintactico | 0.87 | 0.895 | 0.8823 |
| unigramas + lda_10 | 0.882 | 0.882 | 0.8820 |
| unigramas ssw + lsa_4 | 0.878 | 0.886 | 0.8820 |
| unigramas + 4gramas sintacticos | 0.88 | 0.883 | 0.8815 |
| unigramas + 5gramas | 0.88 | 0.883 | 0.8815 |
| unigramas + lsa_10 | 0.88 | 0.883 | 0.8815 |
| 2gramas ssw | 0.887 | 0.876 | 0.8815 |
| unigramas ssw + 2gramas | 0.864 | 0.899 | 0.8812 |
| unigramas + lsa_4 | 0.88 | 0.882 | 0.8810 |
| unigramas ssw + 5gramas sintácticos | 0.878 | 0.884 | 0.8810 |
| unigramas ssw + 5gramas | 0.878 | 0.884 | 0.8810 |
| unigramas + 4gramas | 0.878 | 0.883 | 0.8805 |
| unigramas + 5gramas sintácticos | 0.878 | 0.882 | 0.8800 |
| unigramas + 4gramas ssw | 0.878 | 0.882 | 0.8800 |
| unigramas ssw + 4gramas sintacticos | 0.877 | 0.883 | 0.8800 |
| unigramas ssw + lda_250 | 0.865 | 0.894 | 0.8793 |
| unigramas + lda_300 | 0.872 | 0.886 | 0.8789 |
| unigramas + lda_250 | 0.872 | 0.886 | 0.8789 |
| unigramas + 2gramas | 0.865 | 0.893 | 0.8788 |
| unigramas ssw + lda_200 | 0.866 | 0.889 | 0.8773 |
| unigramas + 5gramas ssw | 0.875 | 0.879 | 0.8770 |
| unigramas + 3gramas | 0.871 | 0.883 | 0.8770 |
| unigramas + lsa_100 | 0.866 | 0.888 | 0.8769 |
| unigramas ssw + 2gramas ssw + lda_100 | 0.838 | 0.917 | 0.8757 |
| unigramas ssw + lsa_100 | 0.856 | 0.893 | 0.8741 |
| unigramas + lsa_50 | 0.856 | 0.89 | 0.8727 |
| unigramas + 2gramas sintácticos | 0.859 | 0.884 | 0.8713 |
| unigramas ssw + lda_500 | 0.856 | 0.886 | 0.8707 |
| unigramas ssw + lsa_150 | 0.85 | 0.892 | 0.8705 |
| unigramas ssw + lsa_50 | 0.844 | 0.898 | 0.8702 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + lsa_150 | 0.857 | 0.883 | 0.8698 |
| unigramas ssw + lda_1000 | 0.859 | 0.88 | 0.8694 |
| unigramas ssw + 2gramas sintacticos | 0.855 | 0.884 | 0.8693 |
| unigramas + lsa_100 | 0.851 | 0.888 | 0.8691 |
| unigramas + lsa_200 | 0.858 | 0.88 | 0.8689 |
| unigramas + lda_200 | 0.855 | 0.883 | 0.8688 |
| unigramas ssw + lda_6 | 0.858 | 0.879 | 0.8684 |
| unigramas | 0.86 | 0.876 | 0.8679 |
| unigramas + lsa_6 | 0.857 | 0.879 | 0.8679 |
| unigramas ssw + lda_1500 | 0.859 | 0.876 | 0.8674 |
| unigramas + lda_4 | 0.86 | 0.874 | 0.8669 |
| unigramas ssw + lda_10 | 0.858 | 0.876 | 0.8669 |
| unigramas ssw + lsa_100 | 0.842 | 0.893 | 0.8668 |
| unigramas + lda_1000 | 0.857 | 0.876 | 0.8664 |
| unigramas ssw | 0.856 | 0.877 | 0.8664 |
| unigramas ssw + lsa_6 | 0.853 | 0.88 | 0.8663 |
| unigramas + lda_500 | 0.847 | 0.886 | 0.8661 |
| unigramas ssw + 3gramas sintácticos | 0.862 | 0.87 | 0.8660 |
| unigramas + lda_1500 | 0.856 | 0.876 | 0.8659 |
| unigramas ssw + lsa_4 | 0.853 | 0.879 | 0.8658 |
| unigramas ssw + lsa_150 | 0.841 | 0.892 | 0.8657 |
| unigramas + 3gramas sintácticos | 0.861 | 0.87 | 0.8655 |
| unigramas + lsa_10 | 0.852 | 0.879 | 0.8653 |
| unigramas + lda_300 | 0.852 | 0.879 | 0.8653 |
| unigramas + lda_250 | 0.851 | 0.88 | 0.8653 |
| unigramas ssw + lsa_10 | 0.849 | 0.882 | 0.8652 |
| unigramas + lda_6 | 0.863 | 0.867 | 0.8650 |
| unigramas + 3gramas ssw | 0.86 | 0.87 | 0.8650 |
| unigramas ssw + lsa_200 | 0.85 | 0.88 | 0.8647 |
| unigramas + lda_10 | 0.86 | 0.869 | 0.8645 |
| unigramas + lsa_4 | 0.856 | 0.873 | 0.8644 |
| unigramas ssw + lda_4 | 0.855 | 0.874 | 0.8644 |

# Experimento A.2

Este experimento utiliza un multi clasificador denominado multiLabel CC, el cual es un clasificador similar a binaria relevancia, como se observa el clasificador base es Naïve Bayes.

Como se puede observar en este caso se obtienen valores similares que en el experimento A.1, aunque en este caso se agrega una combinación con el modelo LDA a 500 tópicos.

Por otro lado, también se observa el caso dónde no se obtiene una mejora del conjunto de datos de unigramas combinados con bigramas ssw, por lo que este conjunto es el mejor clasificado de su tipo.

El concentrado de resultados se puede observar en la tabla A.2.1, en la tabla A.2 se muestran los parámetros utilizados en el clasificador.

Tabla A.2. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | MultiLabel CC |
| Clasificador base | Naïve Bayes |
| Método de valudación | k-Fold cross with k = 10 |

Tabla A.2.1. Concentrado de resultados del experimento A.2.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 2gramas ssw + lda_1000 | 0.879 | 0.933 | 0.9052 |
| unigramas ssw + 2gramas ssw + lda_6 | 0.877 | 0.935 | 0.9051 |
| unigramas ssw + 2gramas ssw + lda_500 | 0.874 | 0.938 | 0.9049 |
| unigramas ssw + 2gramas ssw | 0.875 | 0.936 | 0.9045 |
| unigramas + 2gramas ssw | 0.876 | 0.928 | 0.9013 |
| unigramas ssw + 2gramas | 0.866 | 0.925 | 0.8945 |
| unigramas ssw + lda_6 | 0.866 | 0.922 | 0.8931 |
| unigramas ssw + 3gramas sintácticos | 0.865 | 0.92 | 0.8917 |
| unigramas + 2gramas | 0.864 | 0.919 | 0.8907 |
| unigramas ssw | 0.856 | 0.926 | 0.8896 |
| unigramas ssw + lsa_6 | 0.855 | 0.926 | 0.8891 |
| unigramas ssw + 2gramas sintácticos | 0.858 | 0.922 | 0.8888 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + 3gramas sintácticos | 0.869 | 0.909 | 0.8886 |
| unigramas ssw + lda_1000 | 0.854 | 0.926 | 0.8885 |
| unigramas + lda_1500 | 0.863 | 0.915 | 0.8882 |
| unigramas ssw + 3gramas | 0.86 | 0.918 | 0.8881 |
| 2gramas ssw | 0.879 | 0.897 | 0.8879 |
| unigramas ssw + lsa_4 | 0.851 | 0.928 | 0.8878 |
| unigramas ssw + lda_1500 | 0.853 | 0.925 | 0.8875 |
| unigramas + lda_6 | 0.867 | 0.909 | 0.8875 |
| unigramas ssw + 4gramas | 0.858 | 0.919 | 0.8875 |
| unigramas ssw + 3gramas ssw | 0.859 | 0.917 | 0.8871 |
| unigramas ssw + lda_500 | 0.851 | 0.925 | 0.8865 |
| unigramas ssw + lsa_10 | 0.85 | 0.926 | 0.8864 |
| unigramas + lda_1000 | 0.859 | 0.915 | 0.8861 |
| unigramas | 0.857 | 0.916 | 0.8855 |
| unigramas + 3gramas ssw | 0.862 | 0.91 | 0.8853 |
| unigramas + lda_300 | 0.859 | 0.913 | 0.8852 |
| unigramas ssw + 5gramas | 0.856 | 0.916 | 0.8850 |
| unigramas ssw + 4gramas sintacticos | 0.855 | 0.917 | 0.8849 |
| unigramas ssw + 4gramas ssw | 0.854 | 0.918 | 0.8848 |
| unigramas + 3gramas | 0.861 | 0.91 | 0.8848 |
| unigramas ssw + lda_4 | 0.851 | 0.921 | 0.8846 |
| unigramas ssw + 5gs | 0.853 | 0.918 | 0.8843 |
| unigramas + lda_4 | 0.86 | 0.91 | 0.8843 |
| unigramas ssw + lda_10 | 0.849 | 0.922 | 0.8840 |
| unigramas + lsa_6 | 0.855 | 0.915 | 0.8840 |
| unigramas + 5gramas | 0.862 | 0.907 | 0.8839 |
| unigramas ssw + 5gramas sintácticos | 0.852 | 0.918 | 0.8838 |
| unigramas + 4gramas sintácticos | 0.859 | 0.91 | 0.8838 |
| unigramas + lda_500 | 0.851 | 0.919 | 0.8837 |
| unigramas + lsa_4 | 0.854 | 0.915 | 0.8834 |
| unigramas + 5gramas sintácticos | 0.859 | 0.909 | 0.8833 |
| unigramas + 2gramas sintácticos | 0.856 | 0.912 | 0.8831 |
| unigramas + 4gramas ssw | 0.858 | 0.909 | 0.8828 |
| unigramas + 4gramas | 0.859 | 0.907 | 0.8823 |
| unigramas + lsa_10 | 0.853 | 0.912 | 0.8815 |
| unigramas + lda_10 | 0.862 | 0.9 | 0.8806 |
| unigramas ssw + 2gramas ssw + lda_100 | 0.835 | 0.93 | 0.8799 |
| unigramas ssw + lsa_100 | 0.844 | 0.919 | 0.8799 |
| unigramas ssw + lda_250 | 0.838 | 0.925 | 0.8794 |
| unigramas + lsa_50 | 0.844 | 0.916 | 0.8785 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + lda_250 | 0.848 | 0.91 | 0.8779 |
| unigramas + lsa_100 | 0.847 | 0.911 | 0.8778 |
| unigramas ssw + lsa_150 | 0.845 | 0.913 | 0.8777 |
| unigramas + 5gramas ssw | 0.853 | 0.903 | 0.8773 |
| unigramas + lsa_150 | 0.847 | 0.909 | 0.8769 |
| unigramas ssw + lda_200 | 0.836 | 0.922 | 0.8769 |
| unigramas ssw + lsa_50 | 0.834 | 0.924 | 0.8767 |
| unigramas ssw + lda_300 | 0.838 | 0.918 | 0.8762 |
| unigramas + lsa_200 | 0.846 | 0.9 | 0.8722 |
| unigramas ssw + lsa_200 | 0.843 | 0.899 | 0.8701 |
| unigramas ssw + lsa_250 | 0.838 | 0.895 | 0.8656 |
| unigramas + lsa_250 | 0.841 | 0.889 | 0.8643 |
| unigramas + lda_200 | 0.828 | 0.903 | 0.8639 |
| unigramas ssw + 2gramas ssw + lda_50 | 0.834 | 0.894 | 0.8630 |
| 2gramas | 0.838 | 0.884 | 0.8604 |
| unigramas ssw + lda_150 | 0.808 | 0.915 | 0.8582 |
| unigramas ssw + lda_100 | 0.816 | 0.904 | 0.8577 |
| unigramas ssw + lsa_300 | 0.832 | 0.883 | 0.8567 |
| unigramas + _lsa_300 | 0.834 | 0.879 | 0.8559 |
| unigramas + lda_150 | 0.815 | 0.9 | 0.8554 |
| unigramas + lda_50 | 0.825 | 0.878 | 0.8507 |
| unigramas ssw + lsa_1500 | 0.811 | 0.889 | 0.8482 |
| unigramas ssw + lsa_1000 | 0.811 | 0.889 | 0.8482 |
| unigramas ssw + lsa_500 | 0.81 | 0.885 | 0.8458 |
| unigramas + _lsa_1000 | 0.816 | 0.877 | 0.8454 |
| unigramas + _lda_100 | 0.809 | 0.883 | 0.8444 |
| unigramas + _lsa_1500 | 0.813 | 0.876 | 0.8433 |
| unigramas + _lsa_500 | 0.812 | 0.873 | 0.8414 |
| unigramas ssw + lda_50 | 0.799 | 0.879 | 0.8371 |
| 2gramas sintácticos | 0.811 | 0.841 | 0.8257 |
| 2gramas ssw 300_lsa | 0.746 | 0.893 | 0.8129 |
| 3gramas ssw  300_lsa | 0.681 | 0.953 | 0.7944 |
| 3gramas 300_lsa | 0.694 | 0.891 | 0.7803 |
| 4gramas 300_lsa | 0.643 | 0.949 | 0.7666 |
| 2gramas sintácticos  300_lsa | 0.708 | 0.822 | 0.7608 |
| 3gramas | 0.788 | 0.735 | 0.7606 |
| 2gramas ssw 200_lsa | 0.654 | 0.864 | 0.7445 |
| unigramas ssw + 2gramas ssw 300_lsa | 0.689 | 0.801 | 0.7408 |
| 2gramas 300_lsa | 0.684 | 0.797 | 0.7362 |
| unigramas ssw + 3gramas 300_lsa | 0.677 | 0.783 | 0.7262 |

# Experimento A.3

Este experimento utiliza un multi clasificador denominado MultiLabel LP, el cual también es un clasificador similar a binaria relevancia y a multilabel CC, como se observa el clasificador base es un Naïve Bayes.

Como se puede observar, al ser un clasificador similar en este caso se obtienen valores parecidos que en los experimentos A.1 y A.2, el mejor conjunto de características clasificadas sigue teniendo como base el conjunto de unigramas ssw y bigramas ssw, aunque en este caso los mejores resultados son ofrecidos por las combinaciones con los modelos LDA 100, 6, 500 y 1000, hasta este experimento el modelo dominante es el modelo LDA.

Por otro lado, también se observa el caso dónde no se obtiene una mejora del conjunto de datos de unigramas combinados con bigramas ssw, por lo que este conjunto es el mejor clasificado de su tipo.

El concentrado de resultados se puede observar en la tabla A.3.1, en la tabla A.3 se muestran los parámetros utilizados en el clasificador.

Tabla A.3. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | MultiLabel LC |
| Clasificador base | Naïve Bayes |
| Método de valudación | k-Fold cross with k = 10 |

Tabla A.3.1. Concentrado de resultados del experimento A.3.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 2gramas ssw + lda_100 | 0.923 | 0.923 | 0.9230 |
| unigramas ssw + 2gramas ssw + lda_6 | 0.922 | 0.922 | 0.9220 |
| unigramas ssw + 2gramas ssw + lda_500 | 0.919 | 0.919 | 0.9190 |
| unigramas ssw + 2gramas ssw + lda_1000 | 0.918 | 0.918 | 0.9180 |
| unigramas ssw + 2gramas ssw | 0.916 | 0.916 | 0.9160 |
| unigramas + 2gramas ssw | 0.912 | 0.912 | 0.9120 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 2gramas | 0.912 | 0.912 | 0.9120 |
| unigramas + 3gramas sintácticos | 0.902 | 0.902 | 0.9020 |
| unigramas ssw + 3gramas sintácticos | 0.902 | 0.902 | 0.9020 |
| unigramas + 2gramas | 0.902 | 0.902 | 0.9020 |
| unigramas ssw + lda_500 | 0.899 | 0.899 | 0.8990 |
| unigramas ssw + 2gramas sintácticos | 0.898 | 0.898 | 0.8980 |
| 2gramas ssw | 0.898 | 0.898 | 0.8980 |
| unigramas ssw + lda_1000 | 0.898 | 0.898 | 0.8980 |
| unigramas + lda_1500 | 0.898 | 0.898 | 0.8980 |
| unigramas + lda_1000 | 0.898 | 0.898 | 0.8980 |
| unigramas + 5gramas sintácticos | 0.896 | 0.896 | 0.8960 |
| unigramas ssw + 4gramas sintácticos | 0.896 | 0.896 | 0.8960 |
| unigramas ssw + lda_10 | 0.896 | 0.896 | 0.8960 |
| unigramas | 0.895 | 0.895 | 0.8950 |
| unigramas + 4gramas sintácticos | 0.895 | 0.895 | 0.8950 |
| unigramas ssw + 3gramas ssw | 0.895 | 0.895 | 0.8950 |
| unigramas ssw + lda_1500 | 0.895 | 0.895 | 0.8950 |
| unigramas ssw + lda_6 | 0.895 | 0.895 | 0.8950 |
| unigramas + lsa_4 | 0.895 | 0.895 | 0.8950 |
| unigramas + 2gramas sintácticos | 0.893 | 0.893 | 0.8930 |
| unigramas + 3gramas ssw | 0.893 | 0.893 | 0.8930 |
| unigramas + 4gramas ssw | 0.893 | 0.893 | 0.8930 |
| unigramas ssw + 3gramas | 0.893 | 0.893 | 0.8930 |
| unigramas + lsa_6 | 0.893 | 0.893 | 0.8930 |
| unigramas + lda_500 | 0.893 | 0.893 | 0.8930 |
| unigramas + 5gramas ssw | 0.892 | 0.892 | 0.8920 |
| unigramas ssw + lsa_10 | 0.892 | 0.892 | 0.8920 |
| unigramas + lda_6 | 0.892 | 0.892 | 0.8920 |
| unigramas ssw | 0.89 | 0.89 | 0.8900 |
| unigramas ssw + 4gramas | 0.89 | 0.89 | 0.8900 |
| unigramas ssw + 4gramas ssw | 0.89 | 0.89 | 0.8900 |
| unigramas ssw + lsa_6 | 0.89 | 0.89 | 0.8900 |
| unigramas ssw + lda_4 | 0.89 | 0.89 | 0.8900 |
| unigramas + lsa_10 | 0.89 | 0.89 | 0.8900 |
| unigramas + lda_250 | 0.89 | 0.89 | 0.8900 |
| unigramas ssw + 5gramas sintácticos | 0.889 | 0.889 | 0.8890 |
| unigramas ssw + 5gramas ssw | 0.889 | 0.889 | 0.8890 |
| unigramas ssw + lsa_4 | 0.889 | 0.889 | 0.8890 |
| unigramas + 3gramas | 0.887 | 0.887 | 0.8870 |
| unigramas + 5gramas | 0.887 | 0.887 | 0.8870 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 5gramas | 0.887 | 0.887 | 0.8870 |
| unigramas ssw + lda_300 | 0.887 | 0.887 | 0.8870 |
| unigramas + lsa_5 | 0.887 | 0.887 | 0.8870 |
| unigramas + 4gramas | 0.886 | 0.886 | 0.8860 |
| unigramas ssw + lda_200 | 0.886 | 0.886 | 0.8860 |
| unigramas + lda_10 | 0.886 | 0.886 | 0.8860 |
| unigramas + lda_4 | 0.886 | 0.886 | 0.8860 |
| unigramas ssw + lsa_50 | 0.883 | 0.883 | 0.8830 |
| unigramas + lda_30 | 0.883 | 0.883 | 0.8830 |
| 2gramas | 0.882 | 0.882 | 0.8820 |
| unigramas + lda_200 | 0.882 | 0.882 | 0.8820 |
| unigramas ssw + lsa_100 | 0.88 | 0.88 | 0.8800 |
| unigramas ssw + lda_250 | 0.88 | 0.88 | 0.8800 |
| unigramas + lsa_100 | 0.88 | 0.88 | 0.8800 |
| unigramas ssw + 2gramas ssw + lda_50 | 0.88 | 0.88 | 0.8800 |
| unigramas + lsa_150 | 0.876 | 0.876 | 0.8760 |
| unigramas + lsa_200 | 0.873 | 0.873 | 0.8730 |
| unigramas ssw + lsa_150 | 0.87 | 0.87 | 0.8700 |
| unigramas ssw + lsa_200 | 0.869 | 0.869 | 0.8690 |
| unigramas ssw + lda_100 | 0.867 | 0.867 | 0.8670 |
| unigramas ssw + lda_50 | 0.863 | 0.863 | 0.8630 |
| unigramas + lsa_25 | 0.862 | 0.862 | 0.8620 |
| unigramas + lsa_300 | 0.858 | 0.858 | 0.8580 |
| unigramas ssw + lsa_500 | 0.857 | 0.857 | 0.8570 |
| unigramas ssw + lsa_250 | 0.857 | 0.857 | 0.8570 |
| unigramas + lsa_1000 | 0.855 | 0.855 | 0.8550 |
| unigramas + lda_150 | 0.854 | 0.854 | 0.8540 |
| unigramas ssw + lsa_1500 | 0.853 | 0.853 | 0.8530 |
| unigramas ssw + lsa_1000 | 0.853 | 0.853 | 0.8530 |
| unigramas ssw + lda_150 | 0.853 | 0.853 | 0.8530 |
| unigramas + lsa_500 | 0.849 | 0.849 | 0.8490 |
| unigramas + lda_50 | 0.848 | 0.848 | 0.8480 |
| unigramas ssw + lsa_300 | 0.847 | 0.847 | 0.8470 |
| unigramas + lsa_1500 | 0.847 | 0.847 | 0.8470 |
| unigramas + lda_100 | 0.844 | 0.844 | 0.8440 |
| 2gramas ssw 300_lsa | 0.843 | 0.843 | 0.8430 |
| 3gramas ssw 300_lsa | 0.843 | 0.843 | 0.8430 |
| 2gramas sintácticos | 0.841 | 0.841 | 0.8410 |
| 4gramas 300_lsa | 0.815 | 0.815 | 0.8150 |
| 3gramas 300_lsa | 0.798 | 0.798 | 0.7980 |

# Experimento A.4

En este experimento se utiliza un clasificador más sencillo que en los experimentos anteriores, lo cual se puede observar en los resultados ya que estos fueron los mismos para todo el conjunto de datos y de un bajo valor, a pesar del utilizar como clasificador base un Naïve Bayes.

El concentrado de resultados se puede observar en la tabla A.4.1, en la tabla A.4 se muestran los parámetros utilizados en el clasificador.

Tabla A.4. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | MultiLabel MajorityLabelSet |
| Clasificador base | Naïve Bayes |
| Método de valudación | k-Fold cross with k = 10 |

Tabla A.4.1. Concentrado de resultados del experimento A.4.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas 300_lda | 0.422 | 0.422 | 0.4220 |
| unigramas 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas 50_lda | 0.422 | 0.422 | 0.4220 |
| unigramas 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 300_lda | 0.422 | 0.422 | 0.4220 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + 2gramas sintácticos 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 50_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 2gramas sintácticos 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 300_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 50_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 3gramas sintácticos 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 300_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 50_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 4gramas sintácticos 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 300_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas + 5gramas sintácticos 50_lda | 0.422 | 0.422 | 0.4220 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + 5gramas sintácticos 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 300_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 50_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 300_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 50_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 2gramas sintácticos 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 10_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 10_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 200_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 200_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 300_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 300_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 4_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 4_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 50_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 3gramas sintácticos 50_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 4gramas sintácticos 100_lda | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 4gramas sintácticos 100_lsa | 0.422 | 0.422 | 0.4220 |
| unigramas ssw + 4gramas sintácticos 10_lda | 0.422 | 0.422 | 0.4220 |

# Experimento A.5

En este experimento se utiliza un clasificador RA*k*EL, no obstante, la importancia de este experimento recae en observar el uso del clasificador base, en este caso se utiliza un ZeroR, por lo que a pesar de tener un buen multi clasificador, este debe ser acompañado de un buen clasificador base, ya que se puede observar en los resultados que estos fueron los mismos para todo el conjunto de datos, además de ser muy bajos.

El concentrado de resultados se puede observar en la tabla A.5.1, en la tabla A.5 se muestran los parámetros utilizados en el clasificador.

Tabla A.5. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | Rakel1 |
| Clasificador base | ZeroR |
| Método de valudación | k-Fold cross with k = 10 |

Tabla A.5.1. Concentrado de resultados del experimento A.5.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas 200_lda | 0.122 | 0.122 | 0.1220 |
| unigramas 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas 4_lda | 0.122 | 0.122 | 0.1220 |
| unigramas 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 200_lda | 0.122 | 0.122 | 0.1220 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + 2gramas sintácticos 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 4_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 2gramas sintácticos 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 200_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 4_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 3gramas sintácticos 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 200_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 4_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 4gramas sintácticos 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 200_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 4_lda | 0.122 | 0.122 | 0.1220 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + 5gramas sintácticos 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas + 5gramas sintácticos 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 200_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 4_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 200_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 4_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 2gramas sintácticos 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 100_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 100_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 10_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 10_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 200_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 200_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 300_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 300_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 4_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 4_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 50_lda | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 3gramas sintácticos 50_lsa | 0.122 | 0.122 | 0.1220 |
| unigramas ssw + 4gramas sintácticos 100_lda | 0.122 | 0.122 | 0.1220 |

# Experimento A.6

En este experimento se muestran un listado de los resultados obtenidos por el conjunto de características mejor clasificadas, utilizando la mejor combinación de multi clasificador RA*k*EL y clasificador base Naïve Bayes.

Como se puede observar los mejores resultados se encuentran utilizando modelos LSA, LDA y LDSA, esto antes del resultado de clasificación arrojado por el conjunto base de características conformado por unigramas ssw y bigramas.

El concentrado de resultados se puede observar en la tabla A.6.1, en la tabla A.6 se muestran los parámetros utilizados en el clasificador.

Tabla A.6. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | Rakel1 |
| Clasificador base | Naïve Bayes |
| Método de valudación | k-Fold cross with k = 10 |
| Corpus de prueba | unigramas sin *stop words* combinadas con bigramas con *stop words*. |

Tabla A.6.1. Concentrado de resultados del experimento A.6.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 2gramas + ldsa_60 | 0.919 | 0.918 | 0.9185 |
| unigramas ssw + 2gramas + ldsa_70 | 0.918 | 0.917 | 0.9175 |
| unigramas ssw + 2gramas + ldsa_50 | 0.919 | 0.916 | 0.9175 |
| unigramas ssw + 2gramas + lsa_30 | 0.916 | 0.913 | 0.9145 |
| unigramas ssw + 2gramas + ldsa_40 | 0.914 | 0.912 | 0.9130 |
| unigramas ssw + 2gramas + lsa_60 | 0.914 | 0.912 | 0.9130 |
| unigramas ssw + 2gramas + ldsa_80 | 0.913 | 0.911 | 0.9120 |
| unigramas ssw + 2gramas + lsa_90 | 0.913 | 0.911 | 0.9120 |
| unigramas ssw + 2gramas + ldsa_30 | 0.912 | 0.911 | 0.9115 |
| unigramas ssw + 2gramas + lsa_5 | 0.912 | 0.911 | 0.9115 |
| unigramas ssw + 2gramas + lsa_6 | 0.912 | 0.911 | 0.9115 |
| unigramas ssw + 2gramas + lsa_70 | 0.912 | 0.911 | 0.9115 |
| unigramas ssw + 2gramas | 0.911 | 0.911 | 0.9110 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + 2gramas + lda_4 | 0.911 | 0.911 | 0.9110 |
| unigramas ssw + 2gramas + lsa_4 | 0.911 | 0.911 | 0.9110 |
| unigramas ssw + 2gramas + ldsa_20 | 0.912 | 0.909 | 0.9105 |
| unigramas ssw + 2gramas + ldsa_5 | 0.912 | 0.909 | 0.9105 |
| unigramas ssw + 2gramas + ldsa_6 | 0.912 | 0.909 | 0.9105 |
| unigramas ssw + 2gramas + lsa_100 | 0.912 | 0.909 | 0.9105 |
| unigramas ssw + 2gramas + lsa_20 | 0.912 | 0.909 | 0.9105 |
| unigramas ssw + 2gramas + lsa_80 | 0.912 | 0.909 | 0.9105 |
| unigramas ssw + 2gramas + lsa_50 | 0.911 | 0.908 | 0.9095 |
| unigramas ssw + 2gramas + lda_5 | 0.909 | 0.909 | 0.9090 |
| unigramas ssw + 2gramas + ldsa_10 | 0.909 | 0.909 | 0.9090 |
| unigramas ssw + 2gramas + ldsa_4 | 0.909 | 0.909 | 0.9090 |
| unigramas ssw + 2gramas + lsa_40 | 0.909 | 0.908 | 0.9085 |
| unigramas ssw + 2gramas + lda_6 | 0.908 | 0.908 | 0.9080 |
| unigramas ssw + 2gramas + lsa_10 | 0.908 | 0.908 | 0.9080 |
| unigramas ssw + 2gramas + ldsa_90 | 0.91 | 0.906 | 0.9080 |
| unigramas ssw + 2gramas + ldsa_100 | 0.91 | 0.905 | 0.9075 |
| unigramas ssw + 2gramas + lda_300 | 0.909 | 0.905 | 0.9070 |
| unigramas ssw + 2gramas + lda_500 | 0.907 | 0.905 | 0.9060 |
| unigramas ssw + 2gramas + lsa_150 | 0.907 | 0.905 | 0.9060 |
| unigramas ssw + 2gramas + lda_450 | 0.907 | 0.903 | 0.9050 |
| unigramas ssw + 2gramas + lda_200 | 0.906 | 0.903 | 0.9045 |
| unigramas ssw + 2gramas + ldsa_150 | 0.906 | 0.903 | 0.9045 |
| unigramas ssw + 2gramas + lda_550 | 0.905 | 0.903 | 0.9040 |
| unigramas ssw + 2gramas + lsa_300 | 0.903 | 0.902 | 0.9025 |
| unigramas ssw + 2gramas + lda_20 | 0.903 | 0.9 | 0.9015 |
| unigramas ssw + 2gramas + lda_10 | 0.9 | 0.9 | 0.9000 |
| unigramas ssw + 2gramas + lda_350 | 0.9 | 0.899 | 0.8995 |
| unigramas ssw + 2gramas + lsa_600 | 0.901 | 0.898 | 0.8995 |
| unigramas ssw + 2gramas + lda_600 | 0.9 | 0.897 | 0.8985 |
| unigramas ssw + 2gramas + lsa_350 | 0.9 | 0.896 | 0.8980 |
| unigramas ssw + 2gramas + lsa_550 | 0.9 | 0.896 | 0.8980 |
| unigramas ssw + 2gramas + lda_400 | 0.897 | 0.894 | 0.8955 |
| unigramas ssw + 2gramas + lda_70 | 0.897 | 0.894 | 0.8955 |
| unigramas ssw + 2gramas + lda_100 | 0.898 | 0.893 | 0.8955 |
| unigramas ssw + 2gramas + lda_250 | 0.897 | 0.893 | 0.8950 |
| unigramas ssw + 2gramas + lsa_200 | 0.894 | 0.892 | 0.8930 |
| unigramas ssw + 2gramas + ldsa_300 | 0.893 | 0.888 | 0.8905 |
| unigramas ssw + 2gramas + lsa_250 | 0.89 | 0.886 | 0.8880 |
| unigramas ssw + 2gramas + ldsa_600 | 0.89 | 0.883 | 0.8865 |

# Experimento A.7

En este experimento se muestran un listado de los resultados obtenidos por el conjunto de características mejor clasificadas en segundo lugar, utilizando la mejor combinación de multi clasificador RA*k*EL y clasificador base Naïve Bayes.

Como se puede observar, al igual que en el experimento anterior, los mejores resultados se encuentran utilizando modelos LSA, LDA y LDSA, esto antes del resultado de clasificación arrojado por el conjunto base de características conformado por unigramas y bigramas.

El concentrado de resultados se puede observar en la tabla A.7.1, en la tabla A.7 se muestran los parámetros utilizados en el clasificador.

Tabla A.7. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | Rakel1 |
| Clasificador base | Naïve Bayes |
| Método de valudación | k-Fold cross with k = 10 |
| Corpus de prueba | unigramas con *stop words* combinadas con bigramas con *stop words*. |

Tabla A.7.1. Concentrado de resultados del experimento A.7.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + 2gramas + lsa_90 | 0.916 | 0.914 | 0.9150 |
| unigramas + 2gramas + ldsa_50 | 0.917 | 0.913 | 0.9150 |
| unigramas + 2gramas + ldsa_70 | 0.915 | 0.912 | 0.9135 |
| unigramas + 2gramas + ldsa_40 | 0.914 | 0.911 | 0.9125 |
| unigramas + 2gramas + ldsa_90 | 0.914 | 0.911 | 0.9125 |
| unigramas + 2gramas + lsa_60 | 0.914 | 0.911 | 0.9125 |
| unigramas + 2gramas + ldsa_60 | 0.913 | 0.911 | 0.9120 |
| unigramas + 2gramas + lsa_100 | 0.913 | 0.911 | 0.9120 |
| unigramas + 2gramas + lsa_70 | 0.913 | 0.909 | 0.9110 |
| unigramas + 2gramas + lsa_30 | 0.912 | 0.909 | 0.9105 |
| unigramas + 2gramas + lsa_40 | 0.912 | 0.909 | 0.9105 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas + 2gramas + lsa_80 | 0.91 | 0.908 | 0.9090 |
| unigramas + 2gramas + lsa_20 | 0.912 | 0.906 | 0.9090 |
| unigramas + 2gramas + lda_400 | 0.91 | 0.906 | 0.9080 |
| unigramas + 2gramas + ldsa_20 | 0.91 | 0.906 | 0.9080 |
| unigramas + 2gramas + ldsa_5 | 0.91 | 0.906 | 0.9080 |
| unigramas + 2gramas + ldsa_100 | 0.909 | 0.906 | 0.9075 |
| unigramas + 2gramas + ldsa_30 | 0.909 | 0.906 | 0.9075 |
| unigramas + 2gramas + lsa_5 | 0.909 | 0.906 | 0.9075 |
| unigramas + 2gramas + lsa_50 | 0.909 | 0.906 | 0.9075 |
| unigramas + 2gramas + lda_600 | 0.909 | 0.905 | 0.9070 |
| unigramas + 2gramas + ldsa_80 | 0.907 | 0.903 | 0.9050 |
| unigramas + 2gramas + ldsa_4 | 0.906 | 0.903 | 0.9045 |
| unigramas + 2gramas + lda_500 | 0.907 | 0.902 | 0.9045 |
| unigramas + 2gramas + lda_6 | 0.906 | 0.902 | 0.9040 |
| unigramas + 2gramas | 0.905 | 0.902 | 0.9035 |
| unigramas + 2gramas + lsa_4 | 0.903 | 0.902 | 0.9025 |
| unigramas + 2gramas + ldsa_6 | 0.904 | 0.901 | 0.9025 |
| unigramas + 2gramas + lda_4 | 0.903 | 0.901 | 0.9020 |
| unigramas + 2gramas + lda_5 | 0.903 | 0.901 | 0.9020 |
| unigramas + 2gramas + ldsa_10 | 0.903 | 0.901 | 0.9020 |
| unigramas + 2gramas + lsa_6 | 0.903 | 0.901 | 0.9020 |
| unigramas + 2gramas + lsa_10 | 0.902 | 0.901 | 0.9015 |
| unigramas + 2gramas + lda_550 | 0.903 | 0.899 | 0.9010 |
| unigramas + 2gramas + lsa_150 | 0.903 | 0.899 | 0.9010 |
| unigramas + 2gramas + lda_450 | 0.903 | 0.898 | 0.9005 |
| unigramas + 2gramas + ldsa_150 | 0.902 | 0.898 | 0.9000 |
| unigramas + 2gramas + lda_10 | 0.901 | 0.896 | 0.8985 |
| unigramas + 2gramas + lda_150 | 0.901 | 0.896 | 0.8985 |
| unigramas + 2gramas + lda_300 | 0.9 | 0.895 | 0.8975 |
| unigramas + 2gramas + lda_250 | 0.901 | 0.893 | 0.8970 |
| unigramas + 2gramas + lda_350 | 0.898 | 0.895 | 0.8965 |
| unigramas + 2gramas + lda_200 | 0.892 | 0.889 | 0.8905 |
| unigramas + 2gramas + lsa_300 | 0.888 | 0.886 | 0.8870 |
| unigramas + 2gramas + lda_80 | 0.889 | 0.884 | 0.8865 |
| unigramas + 2gramas + lsa_450 | 0.887 | 0.884 | 0.8855 |
| unigramas + 2gramas + ldsa_450 | 0.886 | 0.883 | 0.8845 |
| unigramas + 2gramas + lsa_350 | 0.885 | 0.883 | 0.8840 |
| unigramas + 2gramas + lsa_600 | 0.885 | 0.883 | 0.8840 |
| unigramas + 2gramas + ldsa_350 | 0.885 | 0.882 | 0.8835 |
| unigramas + 2gramas + ldsa_600 | 0.885 | 0.882 | 0.8835 |

# Experimento A.8

En este experimento se muestran un listado de los resultados obtenidos al clasificar diversos conjuntos de características, utilizando la combinación de multi clasificador RA*k*EL y un clasificador SVM utilizado como clasificador base, utilizando como kernel del clasificador *PolyKernel*.

Como se puede observar, en este caso, los mejores resultados no se encuentran utilizando modelos que se complementan con sus reducciones LSA, LDA y LDSA, sino que dichas reducciones ofrecen los mejores resultados, aunque si es importante mencionar que los modelos complementados con las reducciones LSA, LDA y LDSA si mejoran los resultados de clasificación obtenidos por los casos base, los cuáles se muestran en negritas, conformados por unigramas y bigramas con y sin *stop words*.

El concentrado de resultados se puede observar en la tabla A.8.1, en la tabla A.8 se muestran los parámetros utilizados en el clasificador.

Tabla A.8. Parámetros de Meka.

| Parámetro | Valor |
|---|---|
| Algoritmo de MultiEtiquetado | Rakel1 |
| Clasificador base | Support Vector Machine |
| Kernel de la SVM | PolyKernel |
| Método de valudación | k-Fold cross with k = 10 |
| Corpus de prueba | unigramas con y sin *stop words* combinadas con bigramas con y sin *stop words*. |

Tabla A.8.1. Concentrado de resultados del experimento A.8.

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw 10_lda | 0.552 | 0.55 | 0.5510 |
| unigramas ssw 4_lda | 0.55 | 0.55 | 0.5500 |
| unigramas ssw 6_lda | 0.536 | 0.536 | 0.5360 |
| unigramas ssw 250_lsa | 0.534 | 0.532 | 0.5330 |
| unigramas ssw 100_lsa | 0.532 | 0.532 | 0.5320 |
| unigramas ssw 10_lsa | 0.532 | 0.532 | 0.5320 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw 150_lsa | 0.532 | 0.532 | 0.5320 |
| unigramas ssw 4_lsa | 0.532 | 0.532 | 0.5320 |
| unigramas ssw 50_lsa | 0.532 | 0.532 | 0.5320 |
| unigramas ssw 6_lsa | 0.532 | 0.532 | 0.5320 |
| unigramas ssw 200_lsa | 0.529 | 0.529 | 0.5290 |
| unigramas ssw 300_lsa | 0.529 | 0.529 | 0.5290 |
| unigramas ssw 1000_lsa | 0.527 | 0.525 | 0.5260 |
| unigramas ssw 1500_lsa | 0.527 | 0.525 | 0.5260 |
| unigramas ssw 500_lsa | 0.527 | 0.525 | 0.5260 |
| unigramas ssw 1000_lda | 0.522 | 0.522 | 0.5220 |
| unigramas ssw 50_lda | 0.522 | 0.522 | 0.5220 |
| unigramas ssw 300_lda | 0.52 | 0.518 | 0.5190 |
| unigramas ssw + bigramas ssw + ldsa_350 | 0.518 | 0.518 | 0.518 |
| unigramas ssw + bigramas ssw + ldsa_400 | 0.518 | 0.518 | 0.518 |
| unigramas ssw + bigramas ssw + ldsa_450 | 0.518 | 0.518 | 0.518 |
| unigramas ssw + bigramas ssw + ldsa_500 | 0.518 | 0.518 | 0.518 |
| unigramas ssw + bigramas ssw + ldsa_550 | 0.518 | 0.518 | 0.518 |
| unigramas ssw + bigramas ssw + ldsa_600 | 0.518 | 0.518 | 0.518 |
| unigramas + bigramas ssw + lda_400 | 0.511 | 0.511 | 0.511 |
| unigramas ssw + bigramas ssw + lsa_350 | 0.507 | 0.507 | 0.507 |
| unigramas ssw + bigramas ssw + lsa_400 | 0.507 | 0.507 | 0.507 |
| unigramas ssw + bigramas ssw + lsa_450 | 0.507 | 0.507 | 0.507 |
| unigramas ssw + bigramas ssw + lsa_500 | 0.507 | 0.507 | 0.507 |
| unigramas ssw + bigramas ssw + lsa_550 | 0.507 | 0.507 | 0.507 |
| unigramas ssw + bigramas ssw + lsa_600 | 0.507 | 0.507 | 0.507 |
| unigramas + bigramas ssw + lda_350 | 0.505 | 0.504 | 0.504 |
| unigramas + bigramas + lda_600 | 0.504 | 0.504 | 0.504 |
| unigramas ssw + bigramas ssw + ldsa_5 | 0.504 | 0.504 | 0.504 |
| unigramas + bigramas ssw + lda_150 | 0.5 | 0.5 | 0.500 |
| unigramas + bigramas ssw + lda_500 | 0.5 | 0.5 | 0.500 |
| unigramas + bigramas ssw + ldsa_350 | 0.5 | 0.5 | 0.500 |
| unigramas + bigramas ssw + ldsa_400 | 0.5 | 0.5 | 0.500 |
| unigramas + bigramas ssw + ldsa_450 | 0.5 | 0.5 | 0.500 |
| unigramas + bigramas ssw + ldsa_500 | 0.5 | 0.5 | 0.500 |
| unigramas + bigramas ssw + ldsa_550 | 0.5 | 0.5 | 0.500 |
| unigramas + bigramas ssw + ldsa_600 | 0.5 | 0.5 | 0.500 |
| unigramas ssw + bigramas + lda_450 | 0.5 | 0.5 | 0.500 |
| unigramas ssw + bigramas ssw + ldsa_10 | 0.5 | 0.5 | 0.500 |
| unigramas ssw + bigramas ssw + ldsa_20 | 0.5 | 0.5 | 0.500 |
| unigramas ssw + bigramas ssw + ldsa_30 | 0.5 | 0.5 | 0.500 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + bigramas ssw + ldsa_4 | 0.5 | 0.5 | 0.500 |
| unigramas ssw + bigramas ssw + ldsa_40 | 0.5 | 0.5 | 0.500 |
| unigramas ssw + bigramas ssw + ldsa_50 | 0.498 | 0.496 | 0.497 |
| unigramas ssw + bigramas ssw + ldsa_6 | 0.498 | 0.496 | 0.497 |
| **unigramas + bigramas ssw** | **0.496** | **0.496** | **0.496** |
| unigramas ssw + bigramas ssw + lda_350 | 0.496 | 0.496 | 0.496 |
| unigramas ssw + bigramas ssw + ldsa_60 | 0.496 | 0.496 | 0.496 |
| unigramas ssw + bigramas ssw + ldsa_70 | 0.496 | 0.496 | 0.496 |
| unigramas ssw + bigramas ssw + lsa_20 | 0.496 | 0.496 | 0.496 |
| unigramas ssw + bigramas ssw + lsa_30 | 0.496 | 0.496 | 0.496 |
| unigramas ssw + bigramas ssw + lsa_40 | 0.496 | 0.496 | 0.496 |
| unigramas ssw + bigramas + ldsa_350 | 0.495 | 0.493 | 0.494 |
| unigramas ssw + bigramas + ldsa_450 | 0.495 | 0.493 | 0.494 |
| unigramas ssw + bigramas + ldsa_500 | 0.495 | 0.493 | 0.494 |
| unigramas ssw + bigramas + ldsa_600 | 0.495 | 0.493 | 0.494 |
| unigramas ssw + bigramas ssw + lsa_6 | 0.495 | 0.493 | 0.494 |
| unigramas ssw + bigramas + lda_10 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + ldsa_400 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + ldsa_550 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + lsa_350 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + lsa_400 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + lsa_450 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + lsa_500 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + lsa_550 | 0.493 | 0.493 | 0.493 |
| unigramas ssw + bigramas + lsa_600 | 0.493 | 0.493 | 0.493 |
| **unigramas ssw + bigramas ssw** | **0.493** | **0.493** | **0.493** |
| unigramas + bigramas + lda_550 | 0.489 | 0.489 | 0.489 |
| unigramas ssw + bigramas + lda_500 | 0.489 | 0.489 | 0.489 |
| unigramas ssw + bigramas + ldsa_300 | 0.486 | 0.486 | 0.486 |
| unigramas ssw + bigramas + lsa_300 | 0.486 | 0.486 | 0.486 |
| unigramas ssw + bigramas + lda_4 | 0.484 | 0.482 | 0.483 |
| unigramas + bigramas + lda_450 | 0.482 | 0.482 | 0.482 |
| unigramas + bigramas + lda_500 | 0.482 | 0.482 | 0.482 |
| unigramas ssw + bigramas + lda_200 | 0.482 | 0.482 | 0.482 |
| unigramas ssw + bigramas + lda_50 | 0.482 | 0.482 | 0.482 |
| unigramas + bigramas + lda_30 | 0.478 | 0.478 | 0.478 |
| unigramas + bigramas + lda_40 | 0.478 | 0.478 | 0.478 |
| unigramas + bigramas + lda_400 | 0.478 | 0.478 | 0.478 |
| unigramas + bigramas + lda_70 | 0.478 | 0.478 | 0.478 |
| unigramas ssw + bigramas + lda_5 | 0.478 | 0.478 | 0.478 |

| Modelo | Precision | Recall | Medida F1 |
|---|---|---|---|
| unigramas ssw + bigramas + lda_6 | 0.477 | 0.475 | 0.476 |
| unigramas + bigramas + lda_10 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lda_150 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lda_300 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lda_4 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lda_5 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lsa_350 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lsa_400 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lsa_450 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lsa_500 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lsa_550 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lsa_600 | 0.475 | 0.475 | 0.475 |
| unigramas ssw + bigramas + lda_250 | 0.475 | 0.475 | 0.475 |
| unigramas ssw + bigramas + lda_40 | 0.475 | 0.475 | 0.475 |
| unigramas ssw + bigramas + lda_400 | 0.475 | 0.475 | 0.475 |
| unigramas ssw + bigramas + lda_70 | 0.475 | 0.475 | 0.475 |
| unigramas + bigramas + lda_100 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + lda_200 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + lda_350 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + lda_50 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + lda_6 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + lda_80 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + ldsa_30 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + ldsa_60 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + lsa_30 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + lsa_60 | 0.471 | 0.471 | 0.471 |
| unigramas ssw + bigramas + lda_550 | 0.471 | 0.471 | 0.471 |
| unigramas ssw + bigramas + ldsa_4 | 0.471 | 0.471 | 0.471 |
| unigramas ssw + bigramas + ldsa_5 | 0.471 | 0.471 | 0.471 |
| unigramas + bigramas + ldsa_300 | 0.468 | 0.468 | 0.468 |
| unigramas + bigramas + lsa_300 | 0.468 | 0.468 | 0.468 |
| **unigramas ssw + bigramas** | **0.468** | **0.468** | **0.468** |
| **unigramas + bigramas** | **0.464** | **0.464** | **0.464** |