PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

CLASIFICACIÓN DE TEXTO BASADO EN AGENTES INTELIGENTES

SEBASTIÁN ENRIQUE RODRÍGUEZ ORTIZ

INFORME FINAL DEL PROYECTO

PARA OPTAR AL TÍTULO PROFESIONAL DE

INGENIERO CIVIL EN INFORMÁTICA

DICIEMBRE, 2015

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

CLASIFICACIÓN DE TEXTO BASADO EN AGENTES INTELIGENTES

SEBASTIÁN ENRIQUE RODRÍGUEZ ORTIZ

Profesor Guía**: Rodrigo Alfaro Arancibia**

Profesor Correferente**: Claudio Cubillos Figueroa**

Carrera: **Ingeniería Civil Informática**

DICIEMBRE, 2015

"We don't read and write poetry because it's cute. We read and write poetry because we are members of the human race. And the human race is filled with passion. And medicine, law, business, engineering, these are noble pursuits and necessary to sustain life. But poetry, beauty, romance, love, these are what we stay alive for."

"Jhon Keating", Dead Poets Society (1989)

Agradecimientos

Quiero agradecer a mi profesor guía, Rodrigo Alfaro, tanto por el apoyo para la realización de esta investigación, como por la oportunidad para poder desarrollarme como un mejor profesional.

Agradecer también a mis padres y hermanos, por el constante apoyo, afecto y comprensión a lo largo no solo de mi formación academica, sino también de mi vida,sin ellos no hubiese podido llegar a esta instancia.

Finalmente agradecer a mis amigos, tanto los que estuvieron como los que están, motivándome cada día a ser una mejor persona.

Resumen

Twitter es una plataforma de microbloggeo donde usuarios comparten sus opiniones con una restricción en la cantidad de caracteres. Dadas las características sociales de Twitter, es una fuente potencial para poder realizar análisis de sentimiento. Dada esta misma razón, opiniones sobre ciertos sujetos tales como personas o marcas pueden cambiar en cortos periodos de tiempo. Un enfoque tradicional en la implementación de un clasificador de sentimientos tiene un rendimiento pobre, debido a que depende de cómo fue entrenado dicho clasificador. Se propone un método novedoso para enfrentar este problema, con la implementación de un sistema multi-agentes para clasificar y un mecanismo de análisis de corpus para el re-entrenamiento del clasificador. Este mecanismo consiste en un agente crítico el cual compara el corpus entrenado por el agente clasificador, con nuevos corpus de documentos de futuras instancias, usando principalmente dos métodos: análisis de t-student y diferencias de histogramas. Un clasificador basado en Naïve Bayes fue implementado junto a este mecanismo con múltiples configuraciones. Los resultados de la experimentación muestran que el mecanismo mejora el rendimiento, cuando es comparado con un clasificador Naive Bayes sin el uso de este mecanismo.

Palabras Clave: Clasificación de texto, Agentes inteligentes, Aprendizaje de máquinas, Sistemas multi-agentes.

Abstract

Twitter is a microblogging platform where users share opinions with a restricted amount of characters. Given the social characteristic of Twitter, it is a potential source for sentiment analysis. For this same reason, opinions of certain subjects such as people or brands can change in short periods of time. A traditional approach of a sentiment classifier implementation performs poorly since it depends on how it is trained. We propose a novel method for tackling this problem, with the implementation of a multi-agent system for classifying and corpus analysis mechanism for retraining the classifier. This mechanism consists of a critic agent which comparing the trained corpus of the classifier agent with new collections of documents from future time steps, using primarily two methods: t-student analysis and histogram differences. A Naïve-Bayes based classifier was implemented with this mechanism with multiple configurations. The results of experimental data show that the mechanism boosts its performance, when compared to a pure Naïve Bayes classifier.

Keywords: Text Classification, Intelligent Agents, Machine Learning, Multi-agent systems.

Índice

Lista de Figuras

Lista de Tablas

1. Introducción

La clasificación de textos corresponde al proceso de etiquetar textos en categorías temáticas predefinidas, dadas ciertas características de los mismos. Dados los avances de la tecnología a través del tiempo, la clasificación automatizada de textos ha sido una materia investigada que propone la automatización de las tareas involucradas en la clasificación de textos manual. Se ha utilizado en diversas aplicaciones, como son por ejemplo el filtrado de correos electrónicos y la organización de información.

Un punto importante para la automatización del proceso de clasificación de textos, es la transformación del texto desde el lenguaje natural a una representación que la máquina pueda entender. Esta representación implica tareas de pre procesamiento del texto, las cuales consisten en distintas técnicas para eliminar redundancia y posible ruido presente en los textos a clasificar.

Respecto a los distintos métodos para realizar esta automatización, se han estudiado y utilizado varios algoritmos dentro de este campo. En este proyecto los esfuerzos se concentrarán en la investigación y desarrollo de un clasificador de texto basado en agentes inteligentes. El concepto de agente ligado a la computación e inteligencia artificial, se refiere a un software o algoritmo que examina el ambiente donde opera mediante sensores o perceptores, para luego realizar acciones las cuales alteran este ambiente mediante efectores. Por su parte, un agente inteligente se refiere a la dotación de capacidad de razonamiento, gracias al área del aprendizaje de máquinas, para interpretar las percepciones, resolver problemas, crear inferencias y determinar las mejores acciones posibles a realizar por parte del algoritmo.

En el presente informe se desarrolla la definición de objetivos del proyecto, para luego formalizar la definición de la problemática a realizar, junto con el marco teórico del proyecto y las técnicas utilizadas para resolver la misma. Para finalizar, se presentará la experimentación con el prototipo de agentes inteligentes, comparando distintas configuraciones obtenidas para los distintos agentes, y presentar las conclusiones obtenidas con base en dichos resultados.

2. Definición de Objetivos

A continuación, se detallan cada uno de los objetivos, tanto generales como específicos.

2.1. Objetivo General

Realizar un análisis e implementación de una arquitectura utilizando agentes inteligentes para un clasificador de textos, buscando la mejora en el rendimiento del clasificador mediante su re-entrenamiento, a través de la evaluación de su conocimiento y un flujo de mensajes.

2.2. Objetivos Específicos

• Contextualizar el marco teórico del tema, analizar la situación actual de la clasificación de textos, agentes inteligentes, sistemas multi-agentes y definir las herramientas a utilizar para llevar a cabo la investigación.

• Proponer un modelo de sistema multi-agente enfocado en la clasificación de textos, definiendo los agentes clasificadores y críticos, sus funcionalidades y la interacción entre ellos.

• Realizar la implementación del modelo de sistema multi-agente, con las estructuras necesarias para permitir el funcionamiento, coordinación y comunicación entre agentes.

• Evaluar el rendimiento de dicho sistema con diferentes configuraciones y realizar comparaciones con otros algoritmos de clasificación de textos sin aplicar el análisis estadístico del corpus de entrenamiento, haciendo pruebas bajo un mismo conjunto de textos.

3. Definición del problema

3.1. Clasificación de texto

La clasificación de textos surge de la necesidad de separar documentos de un tema o clasificación específica de un conjunto de documentos de diferentes temas. Al lograr clasificar los documentos por temas, la búsqueda de información se puede realizar de manera sencilla.

Debido al elevado número de documentos que pueden pertenecer a una colección de documentos, particularmente en formato electrónico, realizar la clasificación en forma manual, provoca que la tarea sea complicada, costosa y que requiera mucho tiempo, por lo que surge la idea de hacerlo automáticamente.

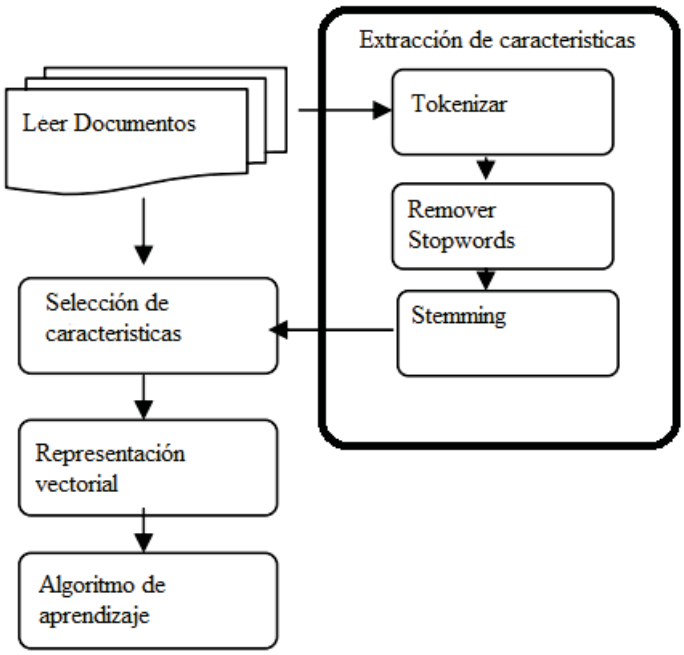

Figura 3.1: Proceso de clasificación de textos

La figura 3.1 muestra el proceso de la clasificación de textos automatizada descrito por [1], en el cual el clasificador lee los documentos y los somete a técnicas de preprocesamiento de documentos, también conocidas como reducción de la dimensionalidad, la cual está compuesta por técnicas de extracción y selección de características. La extracción de características esta compuesta por tres tareas, las cuales son tokenizar el texto, eliminación de las stopwords y el enraizamiento de las palabras (Stemming). Tanto estas tareas como la selección de características serán vistas en profundidad en el punto 3.1.4. Una vez que estos procesos son realizados, se procede a realizar la representación vectorial de los documentos para posteriormente direccionarlos al algoritmo de aprendizaje, que en este proyecto corresponderá a un agente inteligente.

3.1.1. Definición formal

Según [11], la clasificación de texto se define formalmente como la tarea de asignar un valor booleano para cada par$\left(d_j, c_i\right)$ perteneciente a DxC, donde D es el dominio de documentos y $C = c1, \ldots, cj$ es un conjunto de categorías predefinidas. Un valor de verdad (T) asignado a $\left(d_j, c_i\right)$ indica que el texto d_j clasifica dentro de la categoría c_i, mientras que un valor de falsedad (F) indica que el documento no clasifica dentro de la categoría. Más formalmente la tarea de clasificar texto consiste en encontrar una función $\Phi : DxC \rightarrow T, F$ llamada clasificadora (regla, hipótesis o modelo) que "coincida lo más posible" (alta efectividad) con la función desconocida $\mathring{\Phi} : DxC \rightarrow T, F$ que describe cómo los documentos deberían ser clasificados.

Para efectos de la construcción de un algoritmo de clasificación automatizada de texto se asume que las categorías son solo etiquetas simbólicas y no agregan conocimiento alguno que ayude a la construcción del clasificador. Por otra parte, se asume que no se cuenta con ningún conocimiento exógeno, y que la clasificación de los documentos solo se hará con conocimiento endógeno que puede ser extraído del documento en sí, es decir, no se cuenta con conocimiento tipo: fecha de publicación, tipo de documento, fuente de publicación, autor etc.

3.1.2. Clasificación de etiquetado simple versus etiquetado múltiple.

Existen distintas restricciones aplicadas a las tareas de clasificación de textos, las cuales dependen de la aplicación que se requiera utilizar. Esto conlleva que, por ejemplo, la clasificación para un documento d_j sea exactamente una sola categoría (d_j sea asignada a la categoría de arte) lo que es llamado etiquetado simple. Existe una variación de este etiquetado conocido como clasificación binaria, la cual consta de que en el caso de que un documento d_j sea clasificado en una categoría c_i, o a su complemento $\bar{c}_i$. Por su parte, el etiquetado múltiple consiste en que un documento d_j puede ser clasificado en cualquier número de categorías en el intervalo de $[0, |C|]$, siempre y cuando cada categoría c_i sean estocásticamente independientes.

Desde un punto de vista teórico, la clasificación binaria corresponde a un caso más general que su contraparte el etiquetado múltiple, sin embargo, es posible transformar los algoritmos utilizados en la clasificación binaria y aplicarlos en la clasificación de etiquetado múltiple. Para esto se necesita transformar el problema de la clasificación de etiquetado múltiple correspondiente a las categorías pertenecientes a C, y transformarlos a $|C|$problemas independientes para cada par $\{c_i, \bar{c}_i\}$. Cabe destacar que el caso de llevar un clasificador de etiquetado múltiple a un clasificador binario o de etiquetado simple es imposible, puesto que para un documento d_j , éste puede tener asignadas varias categorías c_i por lo que elegir la categoría más apropiada puede que no sea lo suficientemente obvio para el algoritmo.

3.1.3. Clasificación de texto centrada en los documentos versus categorías

Los clasificadores de textos tienen dos maneras de utilizarse: por una parte la clasificación centrada en los documentos consiste en analizar todas las categorías a las cuales un documento d_j podría ser asignado. Por otro lado, la clasificación centrada en categorías se refiere a que, dada una categoría c_i, se buscan todos los documentos que puedan corresponder con dicha categoría. Es necesario hacer esta distinción, puesto que cada uno de estos enfoques tiene distintas aplicaciones.

La clasificación centrada en los documentos es apropiada cuando los documentos a clasificar no están disponibles desde un comienzo, sino que están disponibles después de un cierto periodo de tiempo (por ejemplo en la clasificación de correos electrónicos). Por el contrario, la clasificación centrada en las categorías es apropiada cuando una nueva categoría es agregada luego de haber clasificado ya una cierta cantidad de documentos, o cuando estos documentos necesitan reclasificarse bajo las categorías existentes incluyendo las nuevas.

3.1.4. Técnicas de preprocesamiento de textos

El preprocesamiento de los textos a clasificar corresponde a una parte importante de la clasificación automática de textos, causando una mejora en el rendimiento del proceso de la clasificación. Esta mejora se debe a la eliminación de elementos redundantes sin perder significancia en los procedimientos necesarios, obteniendo de esta manera representaciones que ocupan menos recursos computacionales. Algunas técnicas de preprocesamiento de textos propuestas por [1] son:

3.1.4.1. Stop Words

Las Stop words se definen como términos que se consideran irrelevantes para la clasificación del documento, ya sea porque no presentan un contenido relevante que ayude al clasificador o por las posibles ocurrencias repetidas en el texto. Conjunciones, verbos auxiliares y artículos entre otras palabras son ejemplos de éstas.

3.1.4.2. Stemming

Esta técnica consiste en el enraizado de palabras comunes y agruparlas en un mismo grupo, eliminado así posibles redundancias por alcance de significados. Un ejemplo de esto sería la agrupación de entrena, entrenaba y entrenan en una categoría de entrenar.

3.1.4.3. Normalización de frecuencias

En el proceso de frecuenciar las palabras para textos largos, puede ocurrir que palabras relevantes para el clasificador se repitan un número considerable de veces, por lo que no es raro realizar una normalización de estas frecuencias para ahorrar espacio de recursos computacionales, junto con una mejor representación de estos.

3.1.4.4. Categorización de las características

Técnica de categorizar ciertas características de textos para realizar una transformación de características similares agrupándolas en una categoría de características madre. Para obtener estas características, existen distintas métricas propuestas por [8, 5], las cuales se detallan en la tabla A.1.

4. Marco Teórico

4.1. Agentes inteligentes

Los agentes corresponden a un área de investigación en la inteligencia artificial y pueden ser descritos como “Cualquier entidad que pueda percibir su entorno a través de sensores y realizar acciones en ese entorno mediante efectores” [9]. Ejemplos de estos agentes pueden ser: un robot cuyos sensores pueden ser cámaras o sensores infrarrojos, y sus efectores, distintos motores que interactúan con el ambiente; un agente humano por su parte tiene ojos, oídos y otros órganos que actúan de sensores, mientras que para los efectores se tienen las manos, piernas y otras extremidades; un agente de software tiene codificado cadenas de bits tanto para sus sensores como para sus acciones.

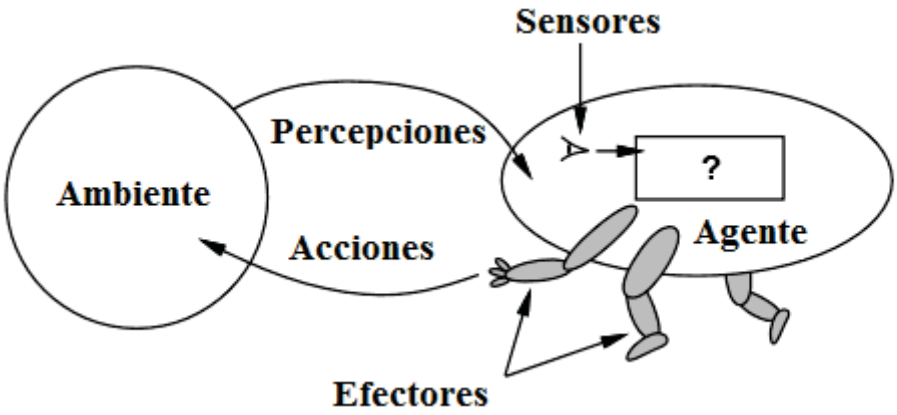

Figura 4.1: Generalización de un agente

Para que un agente sea inteligente debe estar dotado de autonomía, capacidad de raciocinio, y debe con cumplir ciertas metas . La capacidad de raciocinio de un agente implica que el agente debería siempre realizar la acción correcta para cierta percepción; mientras que la acción correcta podría definirse como la acción más exitosa dentro de todo el grupo de acciones. Para que el agente sea capaz de tomar esa acción correcta, es necesario entonces introducir medidas de rendimiento del agente.

Las medidas de rendimiento son el criterio de éxito para el comportamiento de un agente. Estas medidas se pueden definir dentro del contexto: cuando un agente es puesto en un ambiente, la secuencia de acciones realizada bajo las percepciones que recibe generan en el ambiente una serie de estados; si estos estados son deseables, se concluye que el agente ha realizado la tarea encomendada exitosamente. Por otra parte, hay que considerar que una medida de rendimiento fija no funciona de la misma manera para todos los agentes, por lo que la selección de una medida de rendimiento debería ser un tema estudiado cuidadosamente. Por ejemplo, la comparación de dos agentes encomendados a la tarea de mover cajas desde un punto A, a un punto B: el primer agente puede realizar esta tarea de manera mediocre hasta que traslade todas las cajas, mientras que el segundo agente puede mover estas cajas de manera rápida entre A y B pero tomando largos periodos de descanso después de mover cierto numero de cajas.

Teniendo en cuenta lo anterior, la capacidad de raciocinio de un agente se puede definir como: “Para cada posible secuencia de percepciones, el agente deberá seleccionar una acción que maximice su medida de rendimiento, dada la evidencia provista por la secuencia de percepciones y el conocimiento que el agente posea.”

4.1.1. Tipos de agentes inteligentes

Existen distintos tipos de agentes inteligentes, los cuales cumplen distintas funciones y por lo mismo, sus aplicaciones pueden variar dependiendo del problema que se esté enfrentando. Estos tipos son:

- Agentes de reflejos simples: Estos agentes actúan únicamente en base a su percepción del estado del ambiente, ignorando todos los estados históricos que han ocurrido a lo largo de su funcionamiento. Estos agentes se basan en reglas de condiciones y acciones para determinar la siguiente acción a realizar.

- Agentes de reflejos basados en modelos: Estos se diferencian de los agentes de reflejos simples mediante el almacenamiento de una estructura encargada de describir sectores del ambiente donde se desarrolla el funcionamiento del agente. Gracias a esto, el agente puede observar el ambiente de manera parcial.

- Agentes basados en metas: Estos agentes expanden las capacidades de los agentes basados en modelos, a través de el uso de metas las cuales describen situaciones deseadas. De esta manera, el agente selecciona las acciones que más se acerquen a la meta a realizar.

- Agentes utilitarios: Estos agentes utilizan el modelo de los agentes basados en metas, al que implementan una función de utilidad. Dicha función es una medida para obtener la optimización de acciones a realizar y con esto mejorar su rendimiento.

- Agentes de aprendizaje: Estos agentes tienen la ventaja de poder operar en ambientes desconocidos y adaptarse a través de esta interacción con el ambiente. A continuación se desarrollará en mayor profundidad la idea de los agentes de aprendizaje puesto que serán los utilizados en este proyecto.

4.1.2. Agentes de aprendizaje

Tal como se describió anteriormente, los agentes de aprendizaje están dotados con la capacidad de aprender y adaptarse al ambiente en el cual están actuando. La arquitectura de un agente de aprendizaje propuesta por [9] se divide en 4 elementos importantes como puede ilustrar la figura 4.2 :

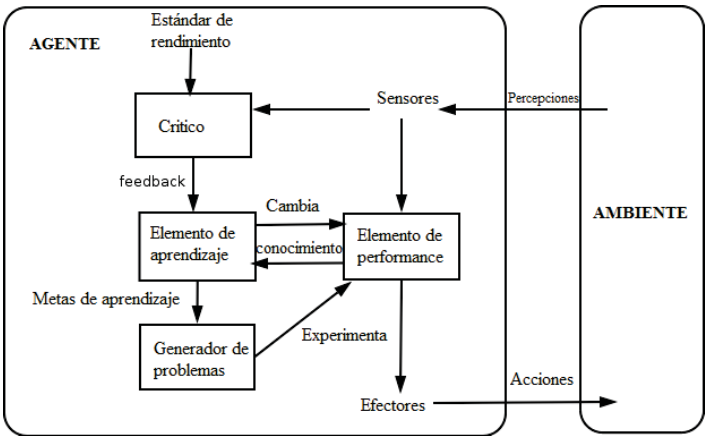

Figura 4.2: Agente de aprendizaje y sus componentes

El elemento de performance es lo que se considera un agente en si, el que que toma las percepciones y elige las acciones a realizar; y se distingue del elemento de aprendizaje ya que este último es el encargado de realizar mejoras. Por su parte, el elemento de aprendizaje depende en gran medida de la estructura del elemento de performance, ya que al diseñar un agente para que aprenda cierta habilidad, se deberá llevar a cabo un análisis para determinar cuál elemento de performance sería el más adecuado una vez que haya concluido su aprendizaje.

El elemento crítico, por otro lado, es el encargado de comunicarle al elemento de aprendizaje el desempeño del agente, todo esto mediante un estándar fijo. Este elemento es necesario debido a que las meras percepciones entregadas por los sensores del agente no son suficientes para indicar si está realizando su tarea. Conceptualmente, se debería considerar este elemento como independiente del agente en sí, puesto que el agente no debería modificar al crítico para que se adapte a su propio comportamiento.

El generador de problemas es el último componente de un agente de aprendizaje, y es el responsable de sugerir acciones que lo lleven a nuevas experiencias. Esta función es útil debido a que propone acciones exploratorias, dándole la ocasión al elemento de performance de ejecutar nuevas acciones y aprender de estas.

Las formas de aprendizaje para este tipo de agente han sido exhaustivamente estudiadas en el campo de la inteligencia artificial, y como tal se procederá a describir lo que se entiende como el aprendizaje de máquinas.

4.2. Aprendizaje de máquinas

El aprendizaje de máquinas es un área dentro de las ciencias de la computación ligada al campo de la inteligencia artificial, donde mediante la aplicación de elementos de optimizacion matemáticos y estadísticos se crean generalizaciones (modelos) de instancias de los algoritmos. Lo anterior genera la capacidad de operación del agente frente a ejemplos y tareas desconocidas. La aplicación de estas técnicas se realiza cuando la implementacion de un algoritmo con todos sus posibles casos e interacciones es imposible.

Para cierto problema a resolver, se encuentran tres tipos de enfoques que se pueden tomar dependiendo de la naturaleza del problema; estos enfoques son:

4.2.1. Aprendizaje supervisado

El aprendizaje supervisado se realiza mediante la recepción de un set de ejemplos etiquetados correspondientes a los datos de entrenamiento, que consisten en pares de objetos de entrada (típicamente vectores) y los datos de salida deseados. Contando con estos datos, el agente debe inferir una función que establezca la relación entre las percepciones y las acciones. Las salidas de la función inferida corresponden a las predicciones realizadas con el conocimiento de los datos de entrenamiento. El objetivo de este tipo de aprendizaje es encontrar el valor de la función para cualquier entrada válida luego de entrenarse con los datos de ejemplo.

Otro aspecto importante es que éste permite la recuperación de información con mayor facilidad, limitando las búsquedas a las clases o categorías que el usuario elige, basándose en el conocimiento intelectual que posee previamente del tema. Los usos comunes del aprendizaje supervisado están asociados a los problemas de regresión y clasificación; un ejemplo de esto sería su aplicación para el filtrado de spam.

El principal problema del aprendizaje supervisado es que al basar el aprendizaje de una función mediante ejemplos de las entradas y salidas del agente, es necesario que el ambiente sea completamente observable; puesto que si es parcialmente observable, es posible que los efectos inmediatos de las acciones realizadas no sean visibles.

4.2.2. Aprendizaje no supervisado

El aprendizaje no supervisado consiste en encontrar la estructura subyacente dentro de un set de datos sin etiquetar. La diferencia con el aprendizaje supervisado y el aprendizaje por refuerzo es que al no existir una etiqueta para estos datos, no se puede implementar un algoritmo que busque encontrar la solución mediante la maximización de una recompensa.

En el aprendizaje no supervisado sólo existen datos de entrada; su objetivo es encontrar regularidades o patrones en estos datos. Existen estructuras con ciertos patrones en los datos de entrada, los cuales aparecen con más frecuencia que otros; se desea encontrar lo que ocurre normalmente y lo que no, el modo en que están organizados los datos. Los documentos se agrupan de acuerdo a su contenido, se puede decir que se auto organizan. Esto se conoce como clasificación no supervisada o clustering. Se denomina de esta manera porque se lleva a cabo de forma totalmente automática, sin control o asistencia manual.

4.2.3. Aprendizaje por refuerzo

El aprendizaje por refuerzo es un área inspirada en la psicología conductista, el cual describe la manera en que agentes de software realizan acciones en un ambiente, con el objetivo de maximizar una función de recompensa acumulativa. El agente interactúa con el ambiente en pasos discretos de tiempo, en que éste recibe una observación con una recompensa asociada, para a continuación elegir una acción a ejecutar que otorgue una mayor recompensa. Una vez efectuada esta acción el ambiente cambia, por lo que el agente percibe nuevas observaciones y por

ende un nuevo set de acciones a realizar. Por lo general este proceso termina cuando se consigue una meta predefinida y con la mayor cantidad de recompensas por las acciones realizadas.

Un ejemplo de aprendizaje por refuerzo es el juego de ajedrez, donde un único movimiento no es relevante, sino la secuencia de movimientos correctos. Un movimiento es un parte de una buena política de juego. Los juegos son una valiosa fuente de investigación en inteligencia artificial, ya que son fáciles de describir pero difíciles de jugar al mismo tiempo. En el juego de ajedrez se tiene una cantidad reducida de reglas, sin embargo, es complejo por la gran variedad de movimientos posibles en cada instante del juego. El sistema tendrá información de su estado sólo al final, cuando se gane o se pierda el juego.

4.2.4. Algoritmos

A continuación se presentan los algoritmos relacionados con las máquinas de aprendizaje. Éstos han sido aplicados a distintos problemas junto con revisiones de sus aplicaciones en relación a la clasificación de textos.

4.2.4.1. Naive-bayes

El modelo de Naive-Bayes (Bayes ingenuo) es una de las formas más tradicionales y sencillas de las redes bayesianas, cuya investigación surge en la década de 1950; este modelo ha presentado éxito considerable en su aplicación en temas relacionados a los problemas de clasificación.[9]. Este modelo es llamado ingenuo debido a que asume que las palabras son condicionalmente independientes entre si dada una cierta clase. Esto resulta ser falso dentro del ámbito de la clasificación de textos, ya que la independencia condicional no se puede aplicar a la aparición de palabras en los documentos.

El clasificador utiliza el conjunto de datos de entrenamiento para estimar la probabilidad de pertenencia a una clase dada la frecuencia de palabras del documento de cada instancia del conjunto de entrenamiento. Esta probabilidad se calcula mediante el teorema de Bayes con una leve variación, dada su naturaleza ingenua y asumiendo la condicionalidad independiente entre palabras.

$$P\left(c_j \mid d\right) = P\left(c_j\right) \prod_{i=1}^{M} P\left(d_i \mid c_j\right) \tag{4.1}$$

A su vez, una estimación $\hat{P}\left(c_j\right)$para $P\left(c_j\right)$ se puede calcular mediante el cuociente de la cantidad de documentos de entrenamiento N_j asignados a la categoría c_j, y el total de documentos existentes$N$:

$$\hat{P}\left(c_j\right) = \frac{N_j}{N} \tag{4.2}$$

También una estimación $\hat{P}\left(d_i \mid c_j\right)$para $P\left(d_i \mid c_j\right)$ la cual toma en cuenta N_{ij} el cual corresponde al número de veces en el cual la palabra i ocurre en los documentos de la categoría c_j:

$$\hat{P}\left(d_i \mid c_j\right) = \frac{1 + N_{ij}}{M + \sum_{k=1}^{M} N_{kj}} \tag{4.3}$$

Respecto distintas aplicaciones de naive-bayes en la clasificación de textos, [10] no solo implementa Naive-Bayes para la clasificación de texto, sino que también propone mejoras para perfeccionar tanto su rendimiento como la precisión entregada. A su vez, [6] investigó sobre el uso de un clasificador de textos con la aplicación de filtrado de spam, haciendo la comparación con otros algoritmos de aprendizaje de máquinas y llegando a la conclusión que Naive-Bayes presentaba resultados prometedores.

4.3. Sistemas multi-agentes

Los sistemas multiagentes corresponden a una red distribuida de agentes los cuales trabajan conjuntamente para encontrar respuestas a problemas que van mas allá de las capacidades o conocimientos individuales de cada entidad. Esto implica que los agentes, además de estar dotados de inteligencia, también deben poseer un protocolo de comunicación para realizar todas las interacciones necesarias.

Entre los beneficios potenciales que otorgan los Sistemas Multi-agentes, se encuentran:

- Velocidad y eficiencia: Los agentes pueden funcionar asincrónicamente y en paralelo.
- Robustez y confiabilidad: La existencia de múltiples agentes presenta algunas características en la tolerancia a fallas.
- Escalabilidad y flexibilidad: El sistema puede ser escalado naturalmente agregando la cantidad necesaria de agentes.
- Costos: Este acercamiento descompone el problema en muchos subsistemas simples con costo unitario menor.
- Desarrollo y reutilización: Los agentes tienen la posibilidad de ser extendidos, reutilizados, probados y mantenidos fácilmente.

En la organización de los sistemas multi-agentes existen distintas formas en la que los agentes pueden interactuar entre sí, ya sea mediante una arquitectura vertical, horizontal o una combinación de ambas. En una arquitectura vertical, los agentes están organizados de tal manera que una primera capa de agentes recibe las percepciones del ambiente, para luego transmitirla hacia otros agentes, los cuales están encargados de tomar las decisiones, y así sucesivamente hasta llegar a los agentes encargados de realizar acciones. En una arquitectura horizontal, por el contrario, todos los agentes pueden percibir y actuar con el ambiente en forma paralela, afectandode manera asincrónica el ambiente en el cual están sujetos todos los agentes. La figura 4.3 ilustra las arquitecturas verticales y horizontales.

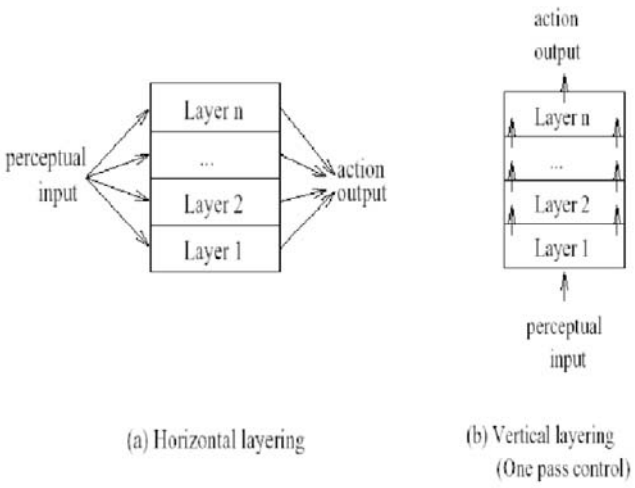

Figura 4.3: Arquitectura horizontal y vertical de sistemas multiagentes

Otra consideración no menor en los sistemas multi-agentes es la arquitectura de comunicación entre ellos; se trata de una arquitectura completamente conectada en la cual todos los agentes son capaces de comunicarse los unos a los otros. Por otro lado, una arquitectura egoísta implica que cada agente restringe su comunicación desde la nulidad hasta un máximo de un agente con el cual se comunica. La importancia de esto radica principalmente en el problema a modelar y la función que tendrá cada uno de los agentes dentro de este modelo.

Respecto a la infraestructura de comunicación entre agentes, pricipalmente hay que definir tres conceptos para una comunicación exitosa, estos son:

- Ontología: Modelos de abstracción de conceptos y las relaciones entre estos para realizar inferencias. La ontología está compuesta por clases, propiedades y relaciones. Las clases corresponden a los elementos generales que forman el lenguaje; las propiedades son las cualidades de estos elementos, y las relaciones son las interrelaciones entre las clases.

- Lenguajes de comunicación: Permiten a los agentes intercambiar y entender mensajes, éste se compone de tres elementos: Sintaxis, o cómo se estructuran los símbolos de la comunicación; semántica, o qué denotan los símbolos, y pragmática, que se refiere a cómo se interpretan estos símbolos.

- Protocolos de interacción: Patrón de comunicación entre agentes mediante una secuencia permitida de mensajes y sus restricciones. Esto provee leyes de comunicación y coordinación entre los agentes.

Respecto a sus aplicaciones, [5] desarrolló un framework de agentes colaborativos para clasificar textos. Si bien éste no aplica técnicas de aprendizaje de máquinas, introduce un método de comunicación colaborativa mediante la implementacion de una función para que un agente pida ayuda a los demás si es que falla al clasificar un texto. A su vez, si el agente logra clasificar un texto con éxito, le informa al resto de los agentes para compartir resultados.

5. Técnicas utilizadas

5.1. Definición de arquitectura

En lo que respecta a la definición de la arquitectura de los agentes, el enfoque propuesto está basado en la figura 4.2 de la sección 4.1.2 Agentes de aprendizaje. En éste, se presenta una arquitectura basada en tres elementos, divididos en tres agentes: El agente distribuidor, el agente clasificador y el agente crítico, tal como ilustra la figura 5.1:

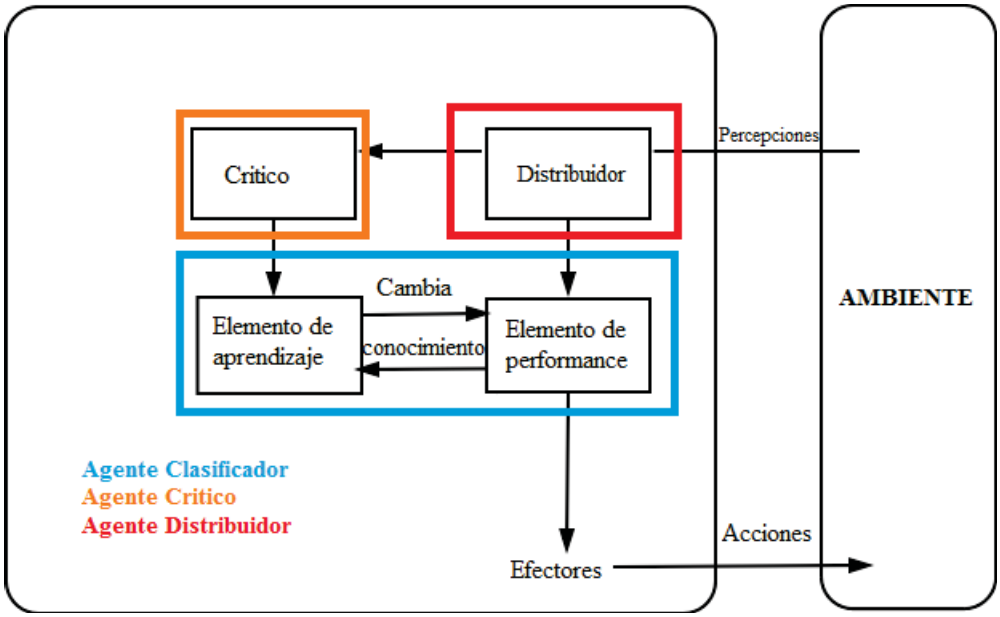

Figura 5.1: Elementos y división de los agentes para la arquitectura

5.2. Distribuidor

El agente distribuidor cumple la función de leer *n* tuits de la pool de documentos (el ambiente), y distribuirlos para su uso tanto para el agente clasificador como para el agente crítico. Para la implementación de las pruebas, la lectura de los *n* tuits de la pool de documentos es lo que aporta la temporalidad a las pruebas, simulando ser lotes de mensajes que este agente va captando para realizar las iteraciones. En su defecto, en una aplicación real este agente va captando tuits y guardándolos en lotes de tamaño *n* para su posterior distribución.

En el caso de las pruebas, el agente distribuidor es quien tiene conocimiento de las verdaderas clases de los tuits; con el que puede realizar tanto las operaciones de re-entrenamiento del agente clasificador de textos, como las evaluaciones del rendimiento en cada instancia del clasificador.

5.3. Clasificador de textos

Para el desarrollo de este proyecto, se implementó un clasificador de textos utilizando el algoritmo de Naive-Bayes con una modificación, la cual se conoce como Naive-Bayes multino-

mial. Ésta está representada en la ecuación 5.1:

$$P\left(c_j \mid d\right) = \ln P\left(c_j\right) + \sum_{i=1}^{M} \ln P\left(d_i \mid c_j\right) \tag{5.1}$$

La razón del uso de esta variación de Naive-Bayes es para evitar un underflow en el cálculo de las probabilidades para las palabras con baja ocurrencia debido a que si existe un vocabulario amplio, estas palabras tienden a tener un cálculo de probabilidad cercano a cero.

La función de re-entrenamiento del clasificador, por otro lado, consiste en separar los tuits de esa iteración, una vez que llega la señal desde el agente crítico. Esta división se realiza en un número definido según la prueba (50 %, 70 % o un 90 %) de la cantidad del lote de tuits en forma aleatoria, para finalmente añadirlos al conocimiento del mismo. Para obtener las verdaderas clases, el clasificador hace la consulta respectiva al agente distribuidor.

Cabe destacar que es posible limitar el conocimiento máximo de textos en el clasificador. En las pruebas detalladas en el punto 6 de este informe, se probaron distintas configuraciones de límites para comprobar estos efectos en el rendimiento del clasificador. Para establecer este límite en el conocimiento, se va comprobando si la cantidad de tuits que existe en el conocimiento, sumado a la cantidad de textos a re-entrenar, no supera el límite. En el caso de que no lo supere, el algoritmo re-entrena los nuevos tuits. En caso contrario, se calcula un *num* el cual indica por cuanto se está excediendo la cantidad de tuits en el conocimiento, para luego eliminar desde un rango de 1 a *num* tuits del conocimiento; a continuación se procede a añadir los nuevos tuits, se genera un nuevo clasificador bayes y posteriormente se entrena con este conocimiento. En el siguiente algoritmo se resume lo explicado anteriormente:

Algorithm 1 Eliminación de conocimiento previo y re-entrenamiento del clasificador

```
Conocimiento = [texto1,texto2,...,textoN]
re_train = [textoN+1,textoN+2,...,textoN+M]
if length(Conocimiento) + length(re_train) > Limite

    num = length(Conocimiento)+ length(re_train) - Limite
    eliminar(Conocimiento, 1:num)
    añadir(Conocimiento,re_train)
    reset_bayes()
    entrenar(Conocimiento)

else

    entrenar(re_train)
    añadir(Conocimiento,re_train)

end if
```

5.4. Crítico

La funcionalidad del crítico radica en el análisis del conocimiento del clasificador de textos. Utilizando distintas medidas estadísticas sobre las palabras de éste, obtiene información relevante. Estas medidas son las siguientes: Pruebas de t-student, pruebas de diferencias de histogramas y la creación de ranking de términos.

La finalidad de estos análisis sobre el conocimiento del clasificador, es analizar si existe una mínima diferencia con otro conjunto de textos. Formalmente, dado un corpus de documentos D y dos subconjuntos A_t y A_{t+1} ambos pertenecientes a D, debe probarse la hipótesis $(A_t - A_{t+1}) \sim N(0,1)$. Un ejemplo grafico de estos conjuntos se ilustra en la tabla 5.1:

		t1	t2	t3	t4	t5	t6	t7
At	d1	0	2	1	5	3	4	1
	d2	2	4	2	0	1	0	3
	d3	0	0	0	5	0	1	2
	d4	0	1	0	2	0	3	0
At+1	d5	1	0	0	4	6	0	2
	d6	4	5	6	0	0	1	2

Tabla 5.1: Conjuntos A_t y A_{t+1} con la representación de la frecuencia de los términos para cada documento.

Otro punto importante para el desarrollo de este proyecto, son dos suposiciones: La primera consiste en que el conjunto A_t es $i.i.d$.Lo anterior implica que los textos pertenecientes a A_t son$i.i.d.$, y esto conlleva a que las palabras de los textos están $i.i.d$. La segunda suposición consiste en que A_t y A_{t+1} corresponden a un flujo de datos, y es por esto que se realiza el análisis de estos dos conjuntos de tal manera que, si el conjunto A_t está normalmente distribuido, no ha de existir una diferencia significativa para un flujo de datos a futuro. En el caso de que exista una diferencia significativa dado cierto umbral, el agente crítico envía una señal al agente clasificador, señalando el re-entrenamiento del conjunto A_{t+1}e incorporándolo al conjunto A_t, de tal manera que, a medida que van llegando más conjuntos (A_{t+2}, A_{t+3},...,A_{t+n}) la diferencia en el conjunto $A_t$en contraste con el flujo futuro, vaya disminuyendo cada vez que éste requiera ser re-entrenado. En las figuras 5.2 y 5.3 se detalla el proceso descrito anteriormente, presentando el proceso tanto en una etapa de setup (la primera iteración), y luego para cada iteración en la cual llegue un nuevo conjunto de textos.

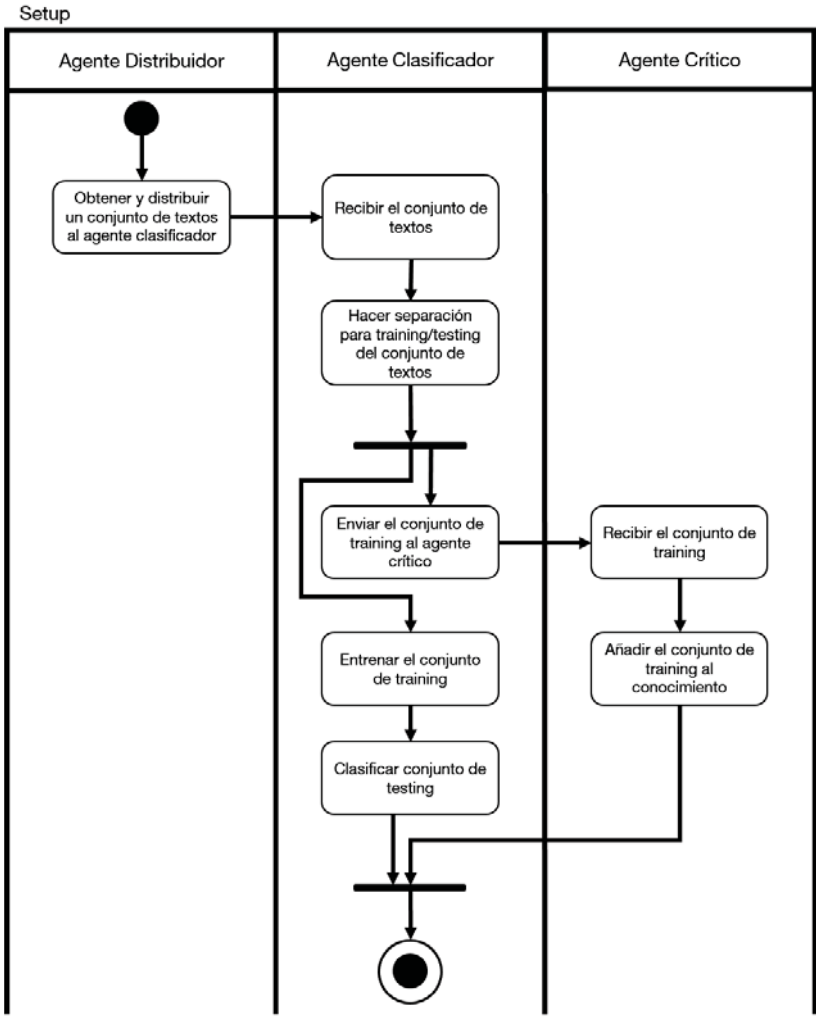

Figura 5.2: Diagrama de actividades entre los distintos agentes en para el proceso de setup del sistema.

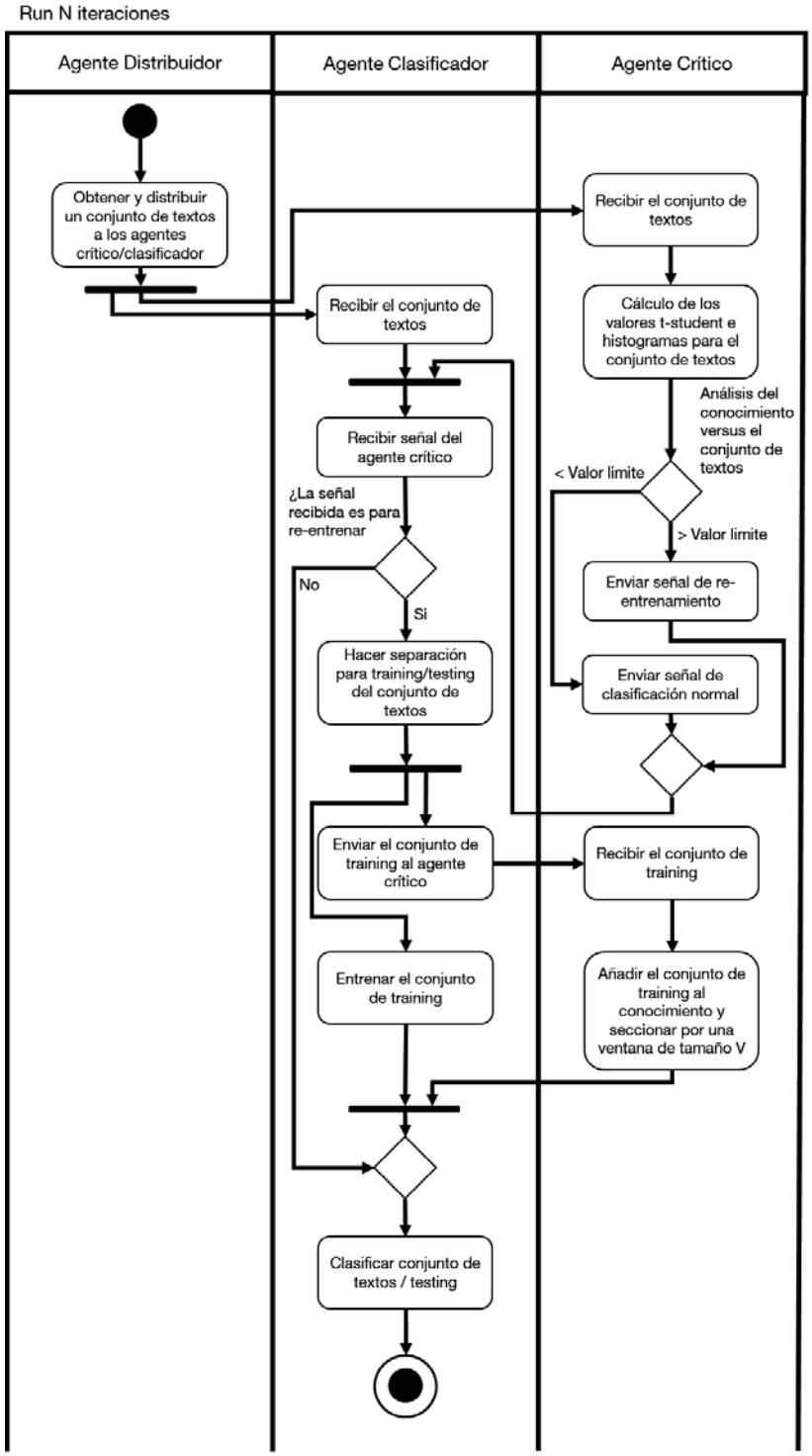

Figura 5.3: Diagrama de actividades entre los distintos agentes en el proceso para cada iteración.

A continuación se detallan las pruebas estadísticas realizadas por el agente crítico, con el fin de poder comprobar si se necesita realizar el proceso de re-entrenamiento.

5.4.1. Pruebas de t-student

Las pruebas de t-student son parte de las pruebas de contraste de hipótesis, donde, si la hipótesis nula no es rechazada, el resultado de la prueba tiene una distribución t-student. Dentro de los usos de estas pruebas se encuentra su utilización para determinar si dos conjuntos de datos son significativamente diferentes uno de otro. Para esto se emplean las medias, las varianzas de dichos conjuntos y dos hipótesis estadísticas: la hipótesis nula, que supone que para ambos conjuntos las medias son iguales, y por el contrario, la hipótesis alternativa, que indica que son desiguales. Lo anterior se representa en la ecuación 5.2:

$$\begin{aligned} H_0 &: \mu_a = \mu_b \\ H_1 &: \mu_a \neq \mu_b \end{aligned} \tag{5.2}$$

Para poder rechazar la hipótesis nula es necesario calcular el resultado de la prueba t y contrastarlo con un valor crítico. Si el resultado de la prueba es mayor que el valor crítico, se rechaza la hipótesis nula y se dice que las medias de ambos conjuntos de datos son distintas. En caso de que el resultado sea menor que el valor crítico, no se rechaza la hipótesis nula, sin embargo no se puede asegurar que las medias de ambos conjuntos sean iguales.

El resultado de la prueba t se calcula con dos fórmulas: Pruebas con varianzas iguales y tamaño de muestras distintas (ecuación 5.3) y pruebas con varianzas distintas y tamaño de muestras distintas (ecuación 5.4):

$$t = \frac{\overline{X}_1 - \overline{X}_2}{S_{x_1x_2}\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}, S_{x_1x_2} = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}} \tag{5.3}$$

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}} \tag{5.4}$$

Puesto que no se puede asumir que para estos conjuntos de datos las varianzas son iguales, se realiza una prueba F para igualdad de dos varianzas. El resultado de esta prueba se calcula con la ecuación 5.5:

$$F = \frac{S_1^2}{S_2^2} \tag{5.5}$$

Este resultado indica que la proporción, mientras más desviada sea de 1, más fuerte es la evidencia de que estas son distintas. Tal como para la prueba t, en ésta se contrastan dos hipótesis: la hipótesis nula que postula que las varianzas son iguales para ambos conjuntos, y la hipótesis alternativa, que supone que ambas varianzas son distintas. Este valor F se contrasta con sus valores críticos, y si estos son excedidos, se puede afirmar que ambas varianzas son distintas.

5.4.2. Diferencia de histogramas

Un histograma corresponde a una representación de las frecuencias de cada valor de un conjunto de datos. En la aplicación de los textos, esta representación se puede graficar, como se presenta en el ejemplo de la figura 5.4, donde se grafican los 50 términos con más ocurrencias dentro de un corpus dado.

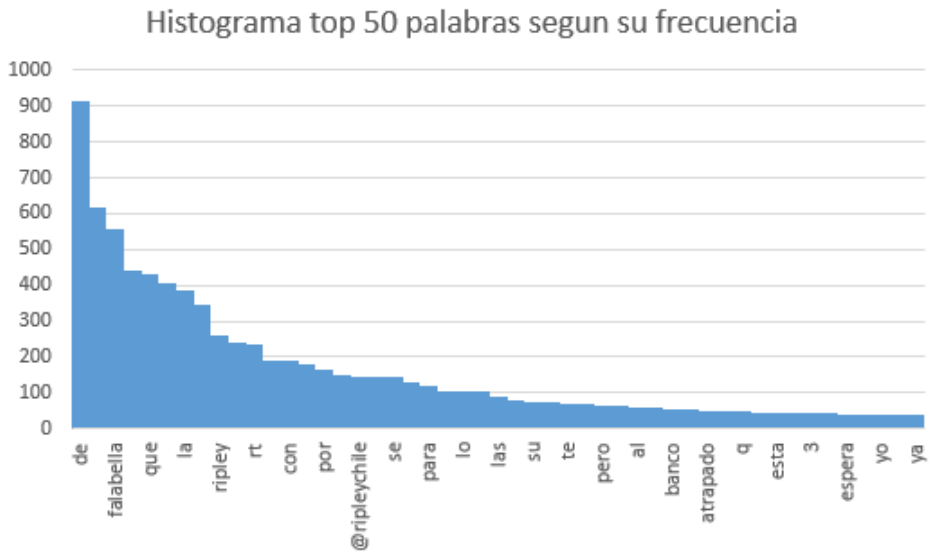

Figura 5.4: Histograma de tuits de las empresas Falabella y Ripley

El análisis de diferencias de histogramas. Sean dos conjuntos de datos A y B con sus correspondientes histogramas $H(A)$ y $H(B)$ y funciones de densidad de probabilidad P y Q respectivamente, se procede a calcular las distancias entre cada elemento de las funciones de densidad. Para esto se utiliza una distancia sorensen [3] la cual se calcula por la ecuación 5.6:

$$d_{sor} = \frac{\sum_{i=1}^{d} |P_i - Q_i|}{\sum_{i=1}^{d} (P_i + Q_i)} \tag{5.6}$$

Cuando el valor de esta distancia es cercana a 0, implica que los conjuntos de datos A y B son similares. Por otra parte, si el valor tiende a alejarse de 0, los conjuntos de datos son diferentes.

5.4.3. Ranking de términos

En un corpus, la cantidad de palabras que existe dentro de esta escala aumenta rápidamente tanto por la cantidad de los textos, como por la naturaleza de los mismos. En el caso que se representen los textos en un corpus como en la tabla 4.1, se generaría una matriz bidimensional DxT , donde D corresponde a los documentos y T a los términos que existen dentro de los documentos. Para reducir el tamaño de esta matriz es posible realizar operaciones de selección

de características las cuales filtrarán palabras no deseadas, puesto que aportan información poco relevante para el modelo.

Una vez reducida la dimensión, se pueden agrupar estos términos por el número de ocurrencias totales en el corpus de manera descendente, generando un histograma para esa nueva matriz. Un ejemplo de esto se puede ver en la figura 4.6, donde una vez obtenida esa matriz nueva, se realizan las pruebas descritas anteriormente. El ranking de términos para este proyecto llevó a cabo de dos maneras: Realizando el ranking para *n* términos de mayor a menor según sus frecuencias, y aplicando la estadística $tf - idf$, la cual calcula la frecuencia de un término y lo multiplica por la frecuencia inversa del documento donde aparece esta palabra. Aplicado estos pesos a cada palabra de la matriz DxT, se eliminan las palabras con bajos valores de $tf - idf$, para luego realizar un ranking de n términos según su frecuencia para esta nueva matriz.

5.5. Implementación

La implementación del proyecto se llevó a cabo en el lenguaje de programación Julia, el cual cumple con las características de ser un lenguaje de alto nivel, orientado a la programación dinámica de alto rendimiento [2] y relativamente sencillo para la realización de prototipos. Si bien Julia ofrece un amplio set de herramientas para el cálculo matemático o estadístico utilizado principalmente por el agente crítico, a la fecha de la creación de este informe, no existe un framework para la creación e implementación sencilla de agentes de software y sistemas multi agentes. Por esta razón se realizó una implementación sencilla utilizando los protocolos de red que provee Julia: cada agente es un servidor, y la comunicación se realiza mediante el envíO0o de mensajes a través del protocolo tcp/ip.

6. Experimentación

6.1. Datos a utilizar

Los datos utilizados en el proyecto, corresponden a un set de mensajes emitidos en la red social Twitter sobre las empresas chilenas Ripley y Falabella. Estos mensajes son denominados tuits y tienen la característica que son publicaciones con un límite de extensión de 140 caracteres conteniendo palabras, números, signos y enlaces; conformando documentos de un tamaño relativamente pequeño. Por otra parte, el set de mensajes está dividido en tres posibles categorías: positivos, negativos y neutrales; que corresponden a la opinión expresada por el usuario ante dicha empresa.

El pre-procesamiento aplicado a estos textos se realizó con el fin de remover los números, signos y enlaces en los mensajes, para luego separarlos en distintos archivos de textos relacionados a sus categorías correspondientes. Una vez en este estado son procesados por el distribuidor, el crítico y el clasificador. Respecto a las cantidades utilizadas, estas corresponden a un total de 3000 tuits, repartidos equitativamente para cada categoría, dejando la separación de entrenamiento y de testing para cada instancia del clasificador descrita anteriormente.

6.2. Técnicas y pruebas

En primer lugar, se realizaron dos pruebas de control las cuales consistieron en diseñar un clasificador que se entrena sólo en la primera instancia por un lado, y un clasificador que se entrena en cada instancia en que llegan los textos por otro. Estas pruebas de control servirán para comparar los resultados obtenidos en los siguientes experimentos realizados, e incluyéndolas, en total se llevaron a cabo diez instancias. Éstas se promediaron para obtener las medidas de evaluación.

6.2.1. Crítico con t-student

Las pruebas del crítico implementando t-student como medida de comparación entre textos conllevaron distintas configuraciones; esto para observar sus efectos en el rendimiento del clasificador dada su configuración. Las distintas iteraciónes realizadas involucraron la implementacion de los agentes con las siguientes configuraciones posibles:

- Límite de conocimiento: [500,1000,1500,2000]
- Separación de textos: [100,150,300]
- Ranking de palabras: [0,100,500,1000]
- Valores de separación para el re-entrenamiento: [50%,70%,90%]
- Valor de umbral crítico: [0.15,0.2,0.25,0.3]
- Uso de Tf-Idf: [Verdadero,Falso]

La cantidad de posibles resultados que se pueden obtener al combinar todas las configuraciones descritas anteriormente, corresponde al orden de las 576 pruebas; las cuales se repitieron 10 veces cada una con el fin de obtener el promedio para cada configuración. Una vez obtenidos los promedios, se procedió a eliminar todos los resultados los cuales no se re-entrenaban, y los que se re-entrenaban en cada iteración de tiempo del algoritmo; esto debido a que este comportamiento es similar al de las pruebas de control 1 y 2 respectivamente. Este proceso de filtrado dejó un total de 147 configuraciones que cumplen el criterio descrito anteriormente. Es decir, en un 74,48% de instancias de configuración se cumplía con alguno de los dos criterios de filtrado (re-entrenar todas las instancias y no re-entrenar ninguna).

6.2.2. Crítico con diferencia de histogramas

Para las pruebas del crítico implementando la diferencia de histogramas como medida para comparación de textos se utilizaron configuraciones similares a las pruebas del crítico con t-student, y se llevaron a cabo otras con parámetros distintos para ver los efectos de dichas configuraciones. Las pruebas fueron:

- Límite de conocimiento: [1000,2000]
- Separación de textos: [150,300]
- Ranking de palabras: [0]
- Valores de separación para el re-entrenamiento: [50%,70%,90%]
- Valor de umbral crítico: [0.05,0.1,0.15]
- Uso de Tf-Idf: [Falso]

La cantidad de posibles resultados que se pueden obtener al combinar estas configuraciones corresponde a un total de 36 pruebas realizadas; éstas, al igual que las pruebas del critico implementando t-student, se repitieron 10 veces cada uno obteniendo el promedio para cada configuración. A continuación se realizó el proceso de filtrado para eliminar las instancias re-entrenadas de la misma manera que en la prueba de control 1 y 2, lo que resultó en una reduccion de 36 a 26 instancias de configuración correspondiendo a una eliminación de un 27,78% de instancias de configuración según los criterios de filtrado.

6.3. Medidas de evaluación

En cuanto a la evaluación del rendimiento de los algoritmos de clasificación de textos se han propuesto distintos métodos consultando la literatura relacionada con el tema [1], los cuales coinciden en que se debe analizar la precisión, recall (recuerdo) y exactitud del clasificador. Para poder realizar estos análisis, se deben determinar los posibles resultados de la clasificación de un documento; y éstos pueden ser representados en una matriz de confusión.

		Valor actual	
		Positivos	Negativos
Valor predecido	Positivos	TP (Verdaderos Positivos)	FP (Falsos Positivos)
	Negativos	FN (Falsos negativos)	TN (Verdaderos Negativos)

Tabla 6.1: Matriz de confusión

En la tabla 4.1 se presenta la matriz de confusión acorde a los posibles resultados de la clasificación de un documento, los cuales corresponden a: verdaderos positivos (TP) si un documento fue clasificado correctamente en una categoría, falsos positivos (FP) si un documento fue clasificado incorrectamente en una clase que no pertenece; falsos negativos (FN) si un documento no es clasificado en una categoría en la que debería estar, y verdaderos negativo (TN) si un documento no fue clasificado en una categoría que no corresponde.

6.3.1. Precisión

La precisión (π_i) es la probabilidad condicional de que un documento d es clasificado en la categoría c_i, y representa la habilidad del clasificador para posicionar un documento en la categoría que le corresponde, en comparación con todos los documentos clasificados correcta e incorrectamente en esa categoría. Esta se define:

$$\pi_i = \frac{TP_i}{TP_i + FP_i} \tag{6.1}$$

6.3.2. Recuerdo

El recuerdo o recall (ρ_i) es la probabilidad de que el clasificador tome la decisión que un documento d_x sea clasificado en categoría c_i y que esta decisión sea acertada. Esta se define:

$$\rho_i = \frac{TP_i}{TP_i + FN_i} \tag{6.2}$$

6.3.3. Exactitud

La exactitud (A_i) corresponde a todos los aciertos del clasificador por sobre los de toda la muestra, esta es una medida definida como:

$$A_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{6.3}$$

6.3.4. Puntaje F_β

El puntaje F_β es un método de medida de exactitud que considera la precisión (π) y el recuerdo (ρ), el valor de F_β puede variar en el intervalo de $[0,1]$, con valores altos correspondientes a una buena exactitud del clasificador. Se define como:

$$F_\beta = (1+\beta^2)\frac{\pi\rho}{(\beta^2)\pi\rho} \tag{6.4}$$

El valor de β es útil para dar mayor énfasis a alguno de los dos valores considerados, generalmente β adopta el valor de $\beta = 1$, indicando una ponderación idéntica para ambos factores. Sin embargo, dependiendo de lo que se esté buscando evaluar, el valor de β puede inclinarse para darle mayor prioridad a la precisión o al recall.

6.3.5. Variaciones micro y macro de las medidas de evaluación

Existen dos tipos de variaciones para las medidas de evaluación descritas anteriormente: micro y macro variantes de las mismas. Éstas se utilizan cuando existen varios set de datos en las pruebas realizadas; en este caso, corresponden a los tres set de categorías utilizadas.

La evaluacion micro corresponde a la suma de los verdaderos positivos, falsos positivos y falsos negativos para luego aplicarlos en las distintas medidas de evaluación, y se utiliza cuando se requiere conocer el rendimiento cuando los sets de datos son de tamaño distinto. La evaluación micro se define para la precisión y recall como:

$$m\pi = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)} \tag{6.5}$$

$$m\rho = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)} \tag{6.6}$$

La evaluación macro corresponde al promedio de las precisiones y recall individuales de los distintos sets de datos. Se utiliza cuando se quiere conocer el rendimiento por sobre todos los set de datos, y se define:

$$M\pi = \frac{\sum_{i=1}^{n} \pi_i}{n} \tag{6.7}$$

$$M\rho = \frac{\sum_{i=1}^{n} \pi_\rho}{n} \tag{6.8}$$

6.4. Resultados

A continuación se presentarán los mejores resultados de las pruebas obtenidas dadas las distintas configuraciónes detalladas anteriormente en contraste con las pruebas de control propuestas, y una breve discusión de los resultados obtenidos. En primer lugar se grafican los resultados de las pruebas de control, luego se procede a graficar las 10 mejores pruebas de la totalidad de pruebas realizadas. Para las distintas configuraciones de dichas pruebas, se enumeran y se enlistan en la tabla del anexo A.2.

Como resumen se presenta la tabla 6.2, con el detalle de la cantidad de re-entrenamiento promedio y la proporcion de re-entrenamiento promedio versus cantidad de iteraciones, en conjunto con el mejor puntaje F_1 para todas las iteraciones y el puntaje F_1 promedio y la desviación estandar de dicho promedio. La prueba número 6 obtuvo el mejor puntaje F_1 para todas las iteraciones, mientras que la prueba número 8 obtuvo el mejor puntaje F_1 promedio. La prueba con menor variacion del puntaje F_1 corresponde a la prueba número 10, mientras que las pruebas que tuvieron la menor proporcion de re-entrenamiento fueron las pruebas 8 y 9.

Prueba N°	Cantidad de re-entrenamiento promedio	Ratio entrenamiento	Mejor Puntaje F1	Puntaje F1 Promedio	Puntaje F1 Desviación Estandar
Control 1	0	0,00%	0,5467	0,5471	0,0053
Control 2	9	100,00%	0,6563	0,6317	0,0489
1	27	93,10%	0,7417	0,6487	0,0835
2	25	86,21%	0,7331	0,6425	0,0719
3	24	82,76%	0,7331	0,6534	0,0601
4	25	86,21%	0,7320	0,6732	0,0838
5	27	93,10%	0,7306	0,6535	0,0640
6	17	89,47%	**0,7571**	0,6759	0,0882
7	18	94,74%	0,7442	0,6739	0,0654
8	15	**78,95%**	0,7336	**0,6789**	0,0592
9	15	**78,95%**	0,7297	0,6772	0,0610
10	16	84,21%	0,7182	0,6649	**0,0524**

Tabla 6.2: Resumen de resultados para todas las pruebas

6.4.1. Control

Para las pruebas de control, tal como se detalló en el punto 6.2, la primera prueba de control se realizó mediante el entrenamiento en la primera instancia y luego para cada instancia, sólo se realizó la clasificación de los textos que iban llegando. Para todas las iteraciones se obtuvo un rendimiento que está entre 0,5371 y 0,5561 de puntaje F_1 (iteración 4 y 3 respectivamente), y un valor promedio de 0,5471 para las 10 iteraciones. Respecto al valor de la cantidad de palabras anómalas en el texto, este no desciende debido a que, al no re-entrenarse el clasificador, no aumenta la cantidad de palabras en el conocimiento del clasificador para poder realizar la comparación.

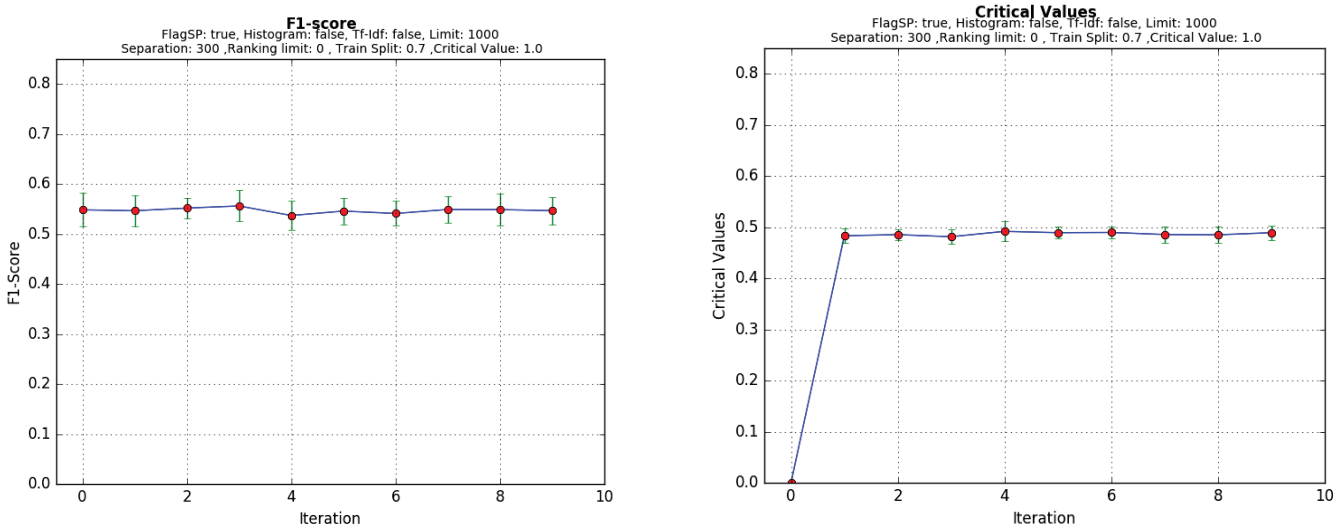

Figura 6.1: Puntaje F_1 y valores t-student para la prueba de control número 1

En la segunda prueba de control, el clasificador se re-entrenó para cada instancia en la cual llegaban los textos, dando como resultado una mejora por sobre la primera prueba de control. El inconveniente de re-entrenar todos los casos posibles es el aumento considerable del conocimiento del clasificador, lo cual si bien mejora el rendimiento, incrementa la complejidad de la computación del algoritmo y por ende, resulta en un aumento de tiempo en la clasificación. Respecto al valor t-student, a medida que se re-entrena el clasificador, la cantidad de palabras anómalas disminuye, lo cual es un comportamiento esperado. El mejor resultado fue obtenido en la iteración 4 con un 0,636 de puntaje F_1, mientras que el peor resultado fue en la iteración 1 con un 0,5385, y un promedio de 0,6317.

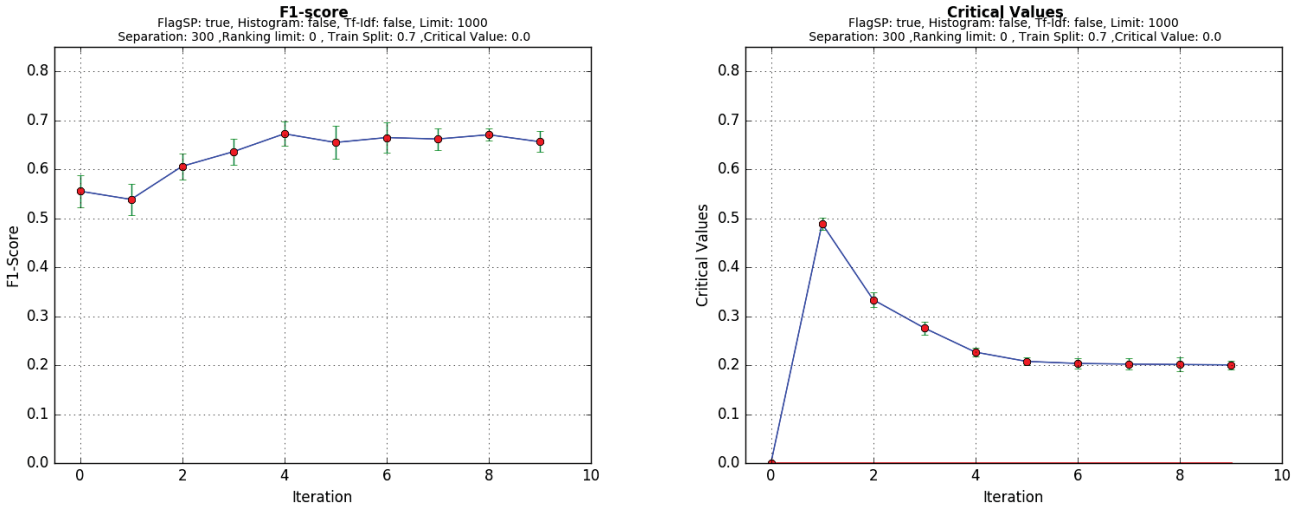

Figura 6.2: Puntaje F_1 y valores t-student para la prueba de control numero 2

6.4.2. Mejores resultados

Para la prueba 1 (figura 6.3), la separación entre iteraciones corresponde a 100 textos por iteración, lo cual permite que existan 30 iteraciones en total. De estas, se obtuvo un promedio de 27 instancias de re-entrenamiento para las 10 pruebas realizadas para esta configuración. En términos de rendimiento, esta configuración obtuvo el mejor puntaje F_1 en la ultima iteración, con un resultado de 0,7417 y un promedio total de 0,6486, destacando además una alta variabilidad en los promedios para cada iteración.

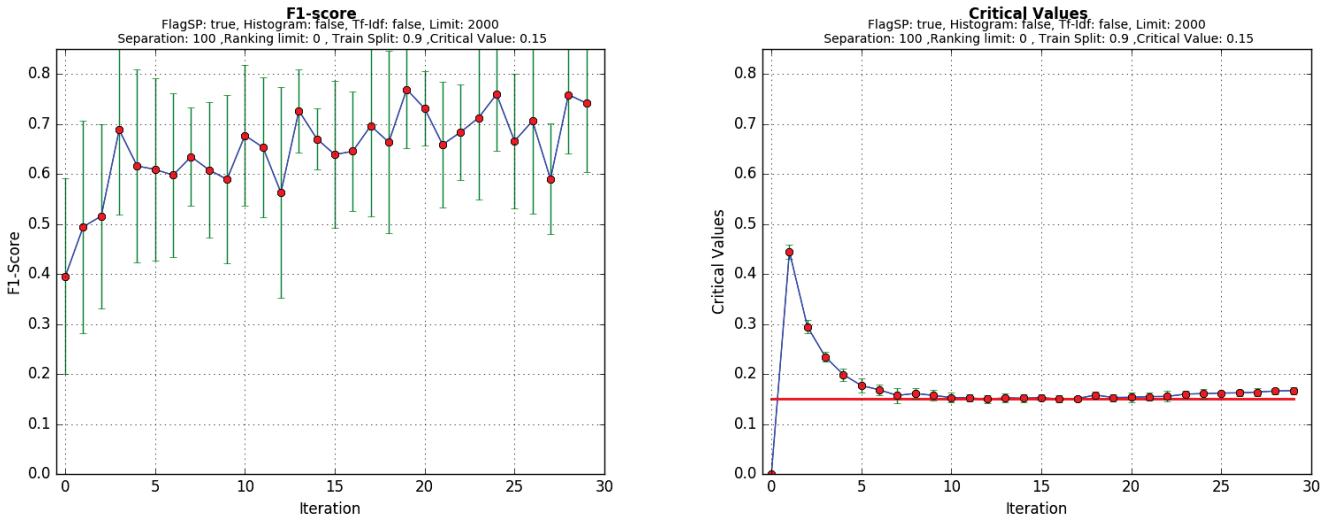

Figura 6.3: Gráfico F_1 y valores t-student para la prueba número 1

En la prueba 2 (figura 6.4) se presenta el mismo comportamiento que en la prueba 1 en términos de 30 iteraciones para esta configuracion, obteniéndose un promedio de 25 instancias de re-entrenamiento. Además de lo anterior, se obtuvo el mejor rendimiento en la última instancia, con un puntaje F_1 de 0,7331 y un promedio de 0,6425. Una diferencia importante que se pudo observar es la reducción en la variabilidad en los promedios para cada iteración, contrastando con la prueba 1.

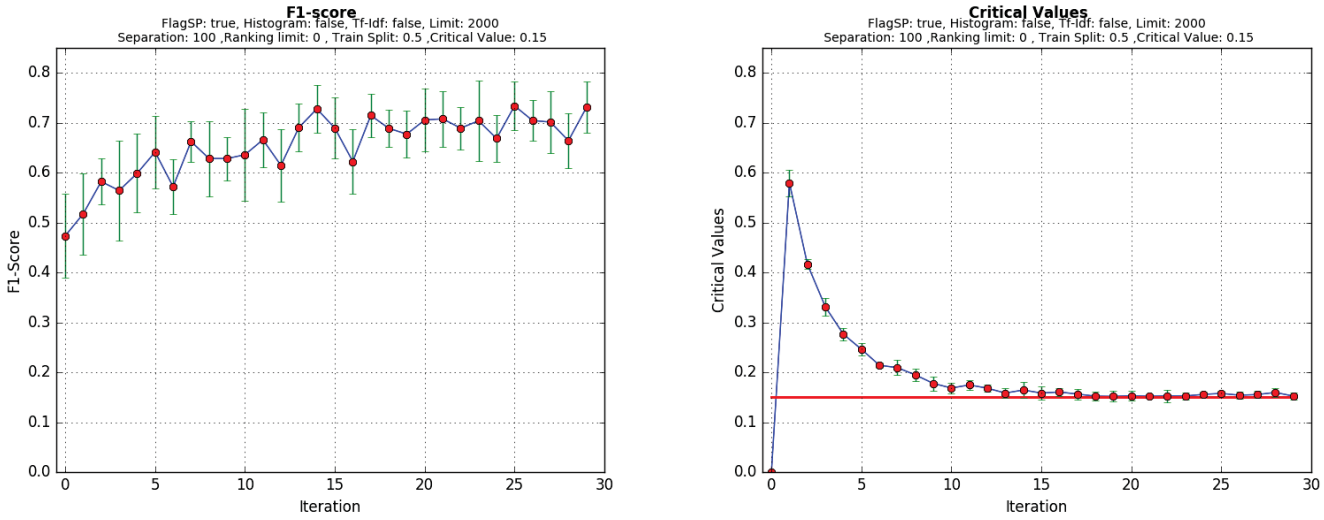

Figura 6.4: Gráfico F_1 y valores t-student para la prueba número 2

Para la prueba 3 (figura 6.5) se encontró un comportamiento similar a la prueba 2, con un promedio de 24 instancias de re-entrenamiento y baja variabilidad en cada iteración. En términos de rendimiento, se obtuvo un puntaje F_1 de 0,7331 y un promedio de 0,6534, siendo esta última medida mejor que en las dos pruebas anteriores.

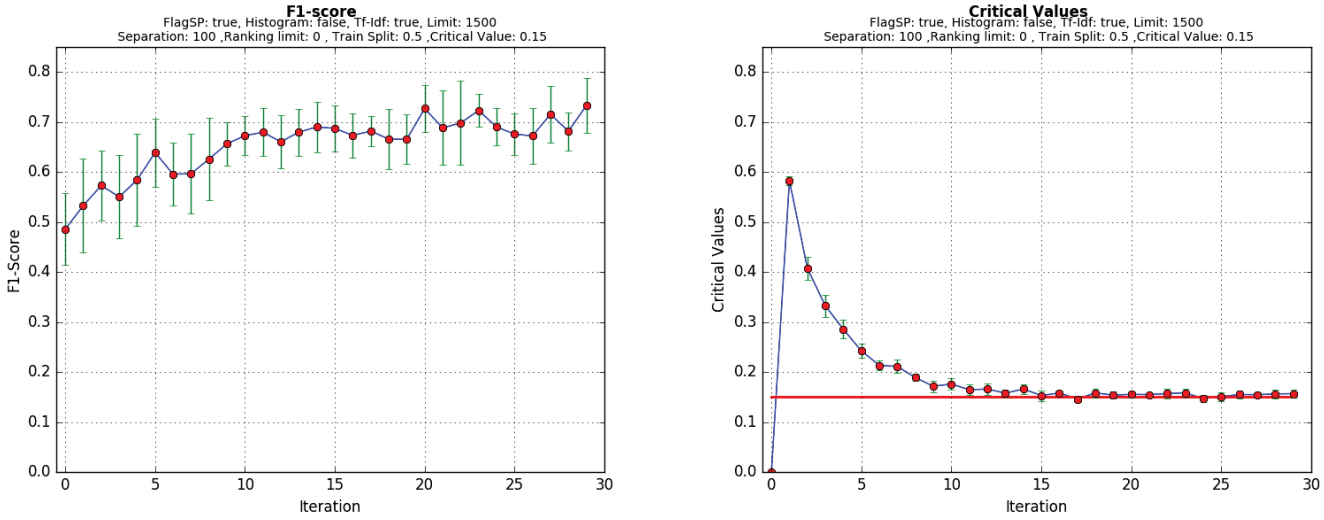

Figura 6.5: Gráfico F_1 y valores t-student para la prueba número 3

En la prueba 4 (figura 6.6) se presentó alta varibildad en cada iteración, observándose un comportamiento similar a la prueba 1. El rendimiento obtenido fue de un 0,732 para la última iteración y un promedio de 0,6732 de puntaje F_1. Respecto a este último valor, se advierte que empieza a haber una mejora en el rendimiento promedio con respecto a la prueba 1, la cual fue la que obtuvo mejor rendimiento en la iteración final.

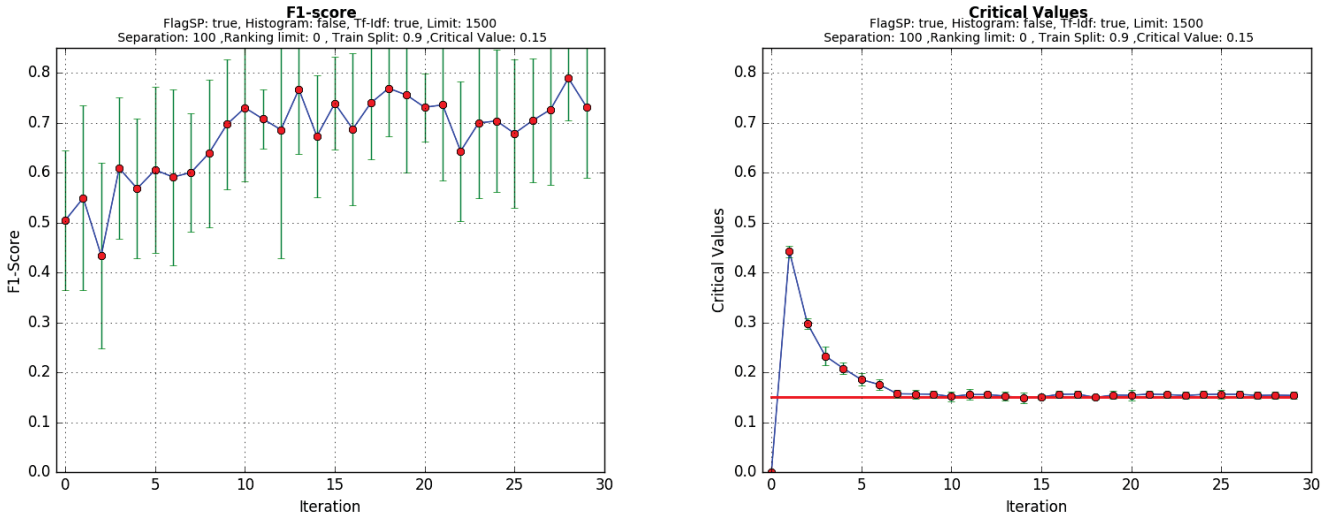

Figura 6.6: Gráfico F_1 y valores t-student para la prueba número 4

Para la prueba 5 (figura 6.7) se encuentra un comportamiento de baja variabilidad entre cada iteración de las pruebas y un promedio de 25 instancias de re-entrenamiento, con un rendimiento de 0,7306 en la ultima iteración y un resultado promedio de 0,6535 de puntaje F_1. En cuento a comparación con pruebas anteriores, esta tiene un comportamiento similar a las pruebas 2 y 3.

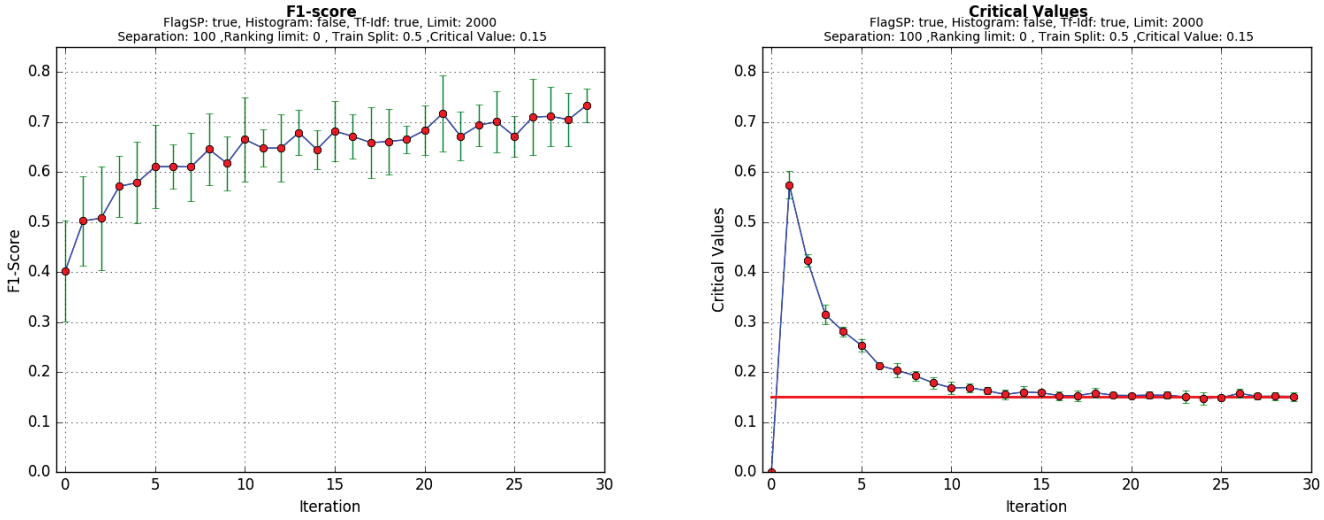

Figura 6.7: Gráfico F_1 y valores t-student para la prueba número 5

La prueba 6 (figura 6.8) se diferencia de las pruebas descritas anteriormente en cuanto a que la cantidad de textos por iteración es de 150, lo que permite un total de 20 iteraciones para la realización de estas pruebas. Se puede observar que existe una alta variabilidad entre iteraciones, y obteniendo un promedio de 17 instancias de re-entrenamiento y un rendimiento de 0,7571 en la última iteración, se observa un rendimiento promedio de 0,6758 de puntaje F_1.

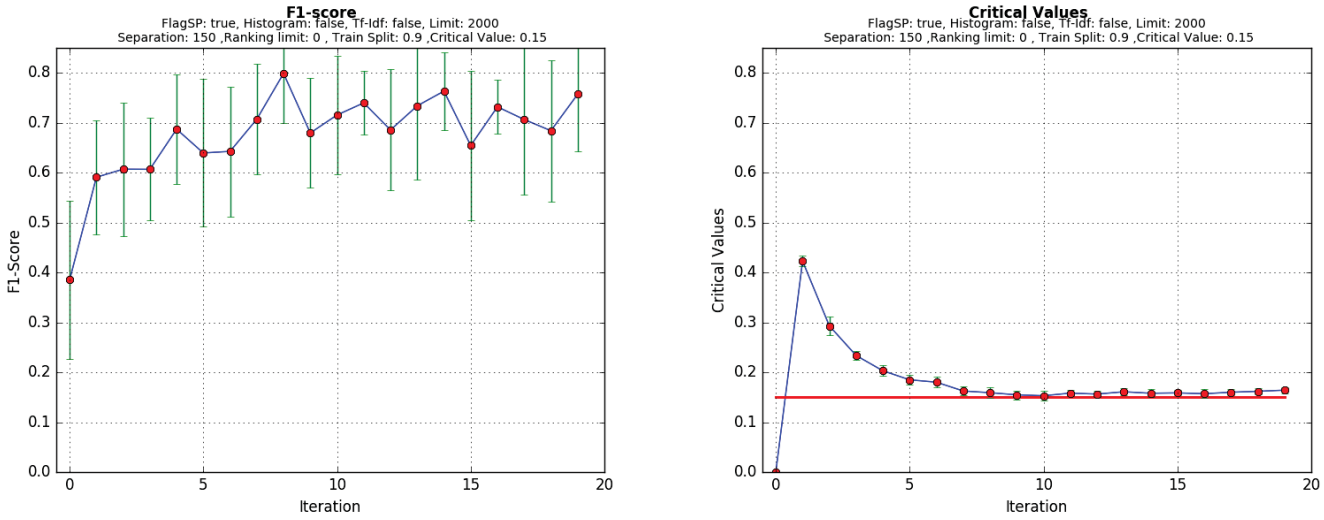

Figura 6.8: Gráfico F_1 y valores t-student para la prueba número 6

Para la prueba 7 (figura 6.9) existe una variabilidad relativamente baja en comparación con la prueba 1, obteniendo un promedio de 18 instancias de re-entrenamiento y un rendimiento del puntaje F_1 0,7442 en la última instancia y un promedio de 0,6739.

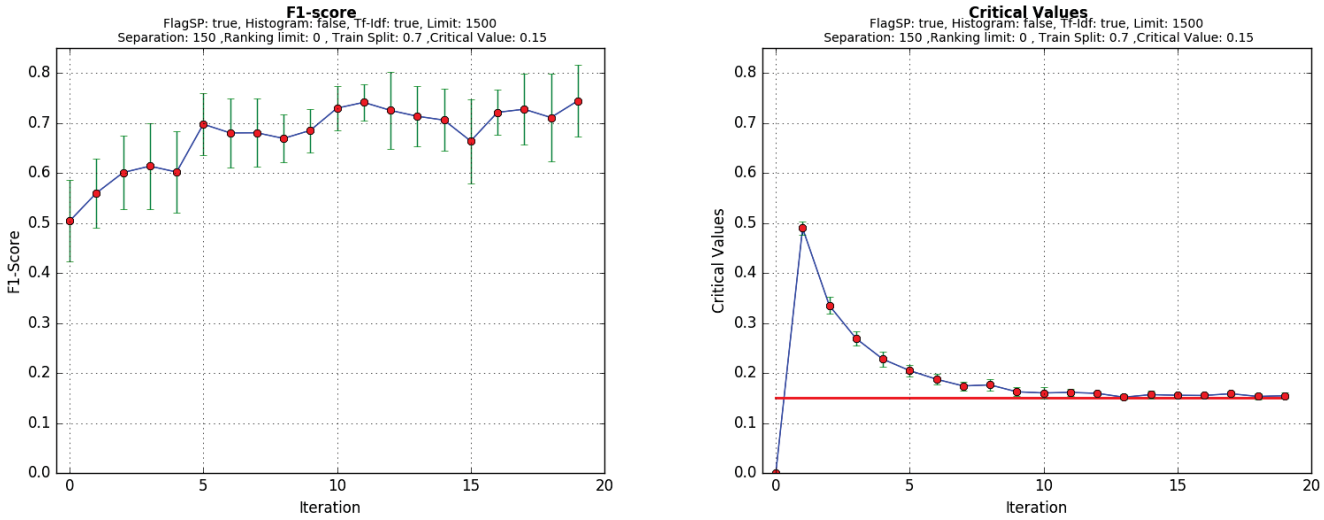

Figura 6.9: Gráfico F_1 y valores t-student para la prueba número 7

Para la prueba 8 (figura 6.10), al igual que en la prueba 7, existe una baja variabilidad en cada iteración para la prueba realizada. Obteniendo 15 instancias de re-entrenamiento promedio, es la más baja en conjunto con la prueba 9 dentro de todas las pruebas presentadas; esto debido a su proporción de entrenamientos promedios y su cantidad de iteraciones, que corresponde a la más baja con un 78.95%. En términos de rendimiento, se obtuvo un 0,7336 y un promedio de 0,6789 de puntaje F_1, presentándose el mejor resultado de puntaje F_1 promedio para todas las pruebas realizadas.

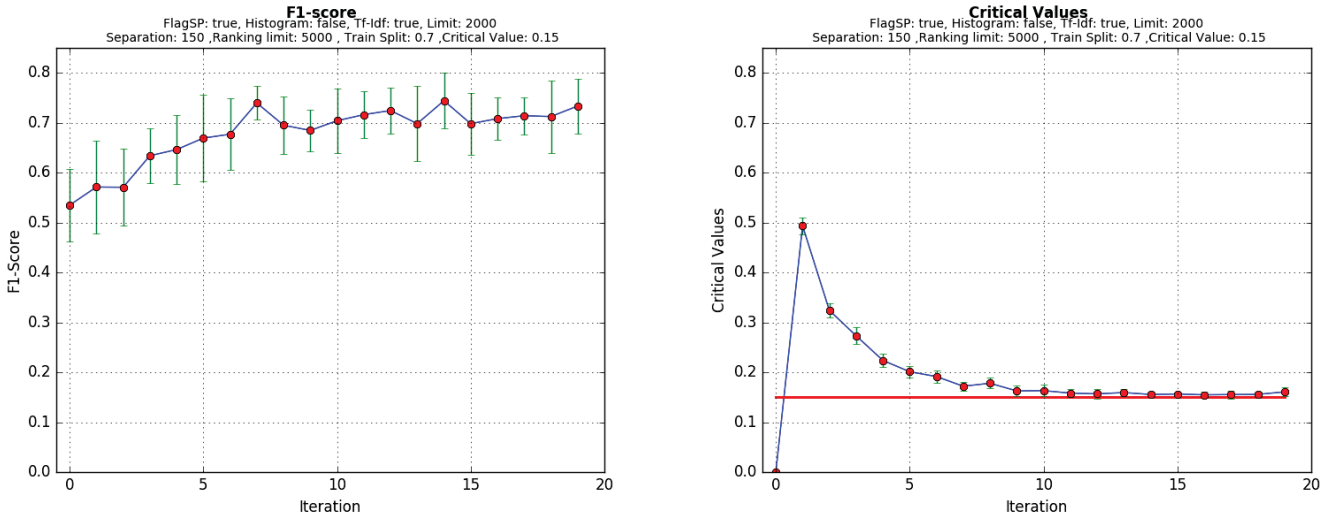

Figura 6.10: Gráfico F_1 y valores t-student para la prueba número 8

Para la prueba 9 (figura 6.11), al igual que la prueba 7 y 8 existe baja variabilidad en cada iteración, y al igual que en la prueba 8, 15 instancias de re-entrenamiento promedio. En cuanto a su rendimiento, se obtuvo un puntaje F_1 de 0,7297 y un promedio de 0,6772.

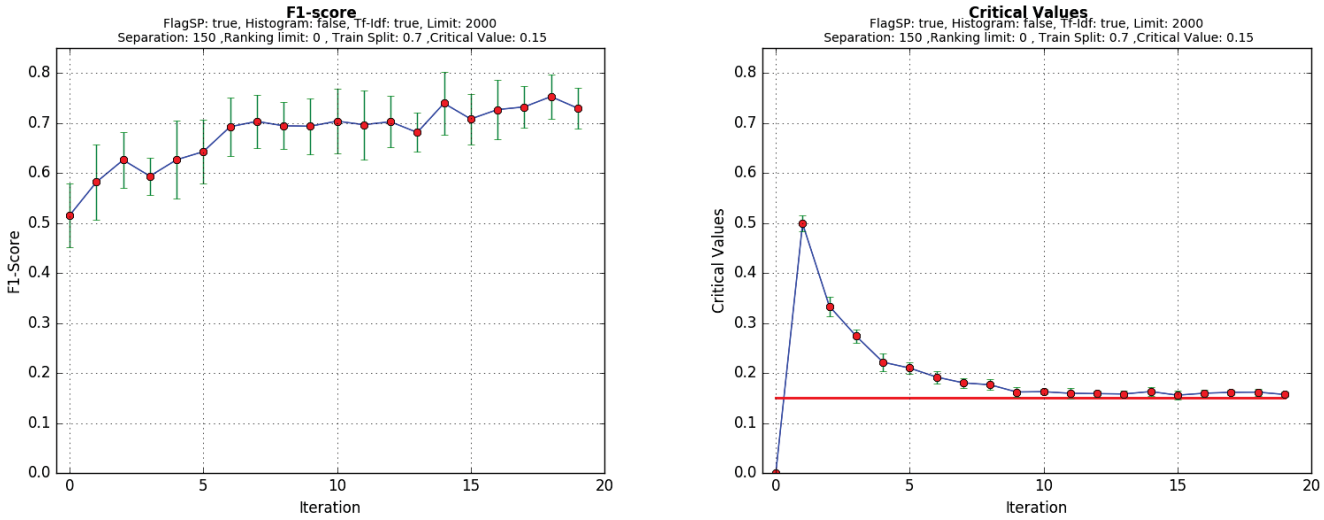

Figura 6.11: Gráfico F_1 y valores t-student para la prueba número 9

Finalmente, para la prueba 10 (figura 6.12), existe baja variabilidad en cada iteración en el análisis del puntaje F_1, pero presenta una alta variabilidad en el promedio de los valores críticos utilizados en el re-entrenamiento; esto puede ser debido a que la comparación entre el conomiento del clasificador y el nuevo conjunto de textos a clasificar se realizó con las 500 palabras las cuales presentan el mayor puntaje de Tf-Idf, dejando pocas palabras para poder llevar a cabo la comparación. En términos de rendimiento se obtuvo un puntaje F_1 de 0,7181 en la última iteración y un promedio de 0,6649.

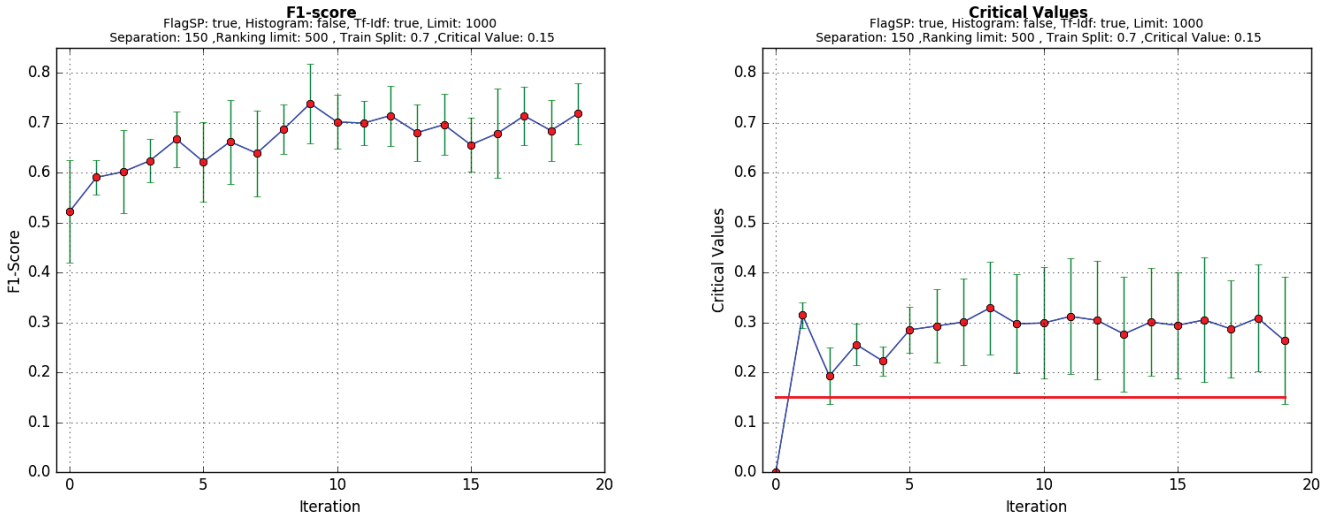

Figura 6.12: Gráfico F_1 y valores t-student para la prueba número 10

6.4.3. Test estadístico

En conjunto con las pruebas descritas anteriormente, se realizaron pruebas t-student de dos colas entre todas las pruebas descritas anteriormente con el objetivo deverificar que las mejoras fueran estadísticamente significativas. En la tabla 6.3 se ilustran los resultados, destacando en amarillo todos los resultados que son estadísticamente significativos. En el caso de la prueba de control 1 y control 2, todos los resultados corresponden a mejoras cuando se hace la comparación con todas las demás pruebas. El otro resultado obtenido con esta tabla evidencia que no existe una diferencia significativa al hacer la comparación entre las pruebas. Esto podria indicar que en términos de una implementación de este metodo para re-entrenar los clasificadores, se podría elegir el método más costo efectivo en términos de rendimiento de puntaje F_1 versus tiempo de ejecución.

	Control 2	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Prueba 6	Prueba 7	Prueba 8	Prueba 9	Prueba 10
Control 1	9,63	-2,35	-6,03	-4,67	-1,86	-4,50	-2,76	-3,80	-4,62	-5,68	-2,86
Control 2		-5,50	-12,95	-10,48	-4,36	-10,42	-6,02	-8,56	-10,48	-12,23	-8,16
Prueba 1			0,37	0,19	-0,02	0,30	-0,21	0,02	0,23	0,25	0,72
Prueba 2				-0,33	-0,32	-0,10	-0,68	-0,53	-0,25	-0,27	0,72
Prueba 3					0,19	0,20	-0,48	-0,24	0,07	0,10	0,88
Prueba 4						0,27	-0,16	0,05	0,21	0,22	0,61
Prueba 5							-0,60	-0,41	-0,13	-0,13	0,71
Prueba 6								0,29	0,52	0,55	1,01
Prueba 7									0,31	0,34	0,98
Prueba 8										0,01	0,82
Prueba 9											0,88

Tabla 6.3: Valores t de la prueba de t student de dos colas comparando las pruebas, con un intervalo de confianza del 95% y 18 grados de libertad.

7. Conclusión

Este estudio busca ser un aporte a la base de conocimiento para la problemática de la clasificación de texto automatizada mediante el uso e implementación de agentes inteligentes.

En la investigación propuesta, se indagó en distintas publicaciones de estudiosos del tema, los cuales aplicaron técnicas estadísticas sobre documentos con resultados favorables, proporcionando una base sólida para la investigación e implementación del proyecto.

Se implementaron dos mecanismos base para el agente crítico. Por una parte, el método del análisis mediante las pruebas de t-student, basado principalmente en el análisis estadístico del corpus de texto presente en el agente clasificador versus un conjunto de textos a clasificar, con el fin de observar si existen anomalías suficientes como para impulsar un re-entrenamiento con respecto a este nuevo conjunto de textos. El otro método consiste en analizar la diferencia de histogramas entre conocimiento del agente clasificador y el corpus de textos a clasificar.

En lo que respecta las pruebas, se pudo demostrar que la implementación del agente clasificador generó beneficios en el rendimiento del agente clasificador. Además, se demostró que un clasificador encargado de re-entrenar en todas las iteraciones no es el enfoque más óptimo para intentar resolver la problemática, tal como se vio con el test de siginificancia estadística.

Cabe destacar que el mejor clasificador en cuanto a rendimiento en la última iteración para todas las pruebas, lo obtuvo la prueba 6. Por otro lado, al analizar la configuración que obtuvo menor cantidad de re-entrenamiento para todas las pruebas seleccionadas y el promedio del puntaje F1, se encontró que el mejor clasificador lo obtuvo la prueba 8. Se observó una tendencia donde todas las pruebas que presentaban una alta variación en cada iteración, compartían la variable de la separación de los documentos para entrenar en un 90%; mientras que los valores más bajos se obtuvieron con la separación de documentos a entrenar en un 50%, indicando una mayor estabilidad en las pruebas con dicha configuración.

Como trabajo futuro, se puede indagar en la optimización del mismo agente crítico, ya sea implementando otro tipo de pruebas estadísticas para la comparación, o mejorando las mismas pruebas ya implementadas y las posibles acciones a realizar. Con respecto al agente clasificador, es posible realizar mejoras ya sea implementado distintos núcleos de clasificador (Maquinas de soporte vectorial, árboles de decisión, etc.), o enfocando el re-entrenamiento, mediante la creación de un conjunto de clasificadores que clasifiquen cooperativamente el conjunto de textos.

Referencias

[1] Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology*, 1(1), 2010.

[2] Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. *CoRR*, abs/1209.5145, 2012.

[3] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions, 2007.

[4] Dwight Deugo, Michael Weiss, and Elizabeth A. Kendall. Reusable patterns for agent coordination. In *Coordination of Internet Agents: Models, Technologies, and Applications*, pages 347–368. 2001.

[5] Yueyu Fu, Weimao Ke, and Javed Mostafa. Automated text classification using a multi-agent framework. In Mary Marlino, Tamara Sumner, and Frank M. Shipman III, editors, *JCDL*, pages 157–158. ACM, 2005.

[6] Thiago S. Guzella and Walmir M. Caminhas. Review: A review of machine learning approaches to spam filtering. *Expert Syst. Appl.*, 36:10206–10222, September 2009.

[7] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[8] Guangzhi Qu, Hui Zhang, and Craig T. Hartrick. Multi-label classification with bayes' theorem. In Yongsheng Ding, Yonghong Peng, Riyi Shi, Kuangrong Hao, and Lipo Wang, editors, *BMEI*, pages 2281–2285. IEEE, 2011.

[9] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[10] Karl-Michael Schneider. Techniques for improving the performance of naive bayes for text classification. In *In Proceedings of CICLing 2005*, pages 682–693, 2005.

[11] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.

[12] Deqing Wang, Hui Zhang, Rui Liu, Weifeng Lv, and Datao Wang. t-test feature selection approach based on term frequency for text categorization. *Pattern Recognition Letters*, 45:1 – 10, 2014.

Anexos

A. Tablas generales

Nombre	Definición
$Frecuencia\,de\,documento\,(DF)$	$DF(t_k) = P(t_k)$
$Frecuencia\,de\,terminos\,(TF)$	$tf\,(f_i,d_j) = \frac{freq_{ij}}{max_k freq_{kj}}$
$Frecuencia\,inversa\,de\,documentos\,(IDF)$	$\lvert idf\rvert = log\frac{\lvert D\rvert}{\lvert\#(f1)\rvert}$
$TF-IDF$	$tf-idf\,(t,d,D) = tf\,(t,d)\,idf\,(t,D)$

Tabla A.1: Tecnicas de selección de caracteristicas

Prueba N°	Función Crítico	Tf-Idf	Límite de conocimiento	Separación de textos	Ranking límites	Separación para el entrenamiento	Límite de valor crítico
1	t-student	false	2000	100	0	90%	15%
2	t-student	false	2000	100	0	50%	15%
3	t-student	true	1500	100	0	50%	15%
4	t-student	true	1500	100	0	90%	15%
5	t-student	true	2000	100	0	50%	15%
6	t-student	false	2000	150	0	90%	15%
7	t-student	true	1500	150	0	70%	15%
8	t-student	true	2000	150	5000	70%	15%
9	t-student	true	2000	150	0	70%	15%
10	t-student	true	1000	150	500	70%	15%

Tabla A.2: Enumeración y configuraciones de las pruebas

B. Tablas de resultados

Prueba Control 1										
Iteración	0	1	2	3	4	5	6	7	8	9
Exactitud	0,513333	0,520333	0,514333	0,524667	0,513	0,522333	0,511667	0,520333	0,512333	0,517
Precisión	0,591115	0,582734	0,589579	0,59269	0,568767	0,581989	0,57561	0,589397	0,591307	0,583583
Recuerdo	0,514318	0,516131	0,520845	0,526098	0,510599	0,516042	0,512498	0,515861	0,514882	0,516569
Puntaje F1	0,548373	0,546511	0,551854	0,556093	0,537054	0,545836	0,541073	0,549042	0,548829	0,54665
Valores T Crit	0	0,483251	0,485365	0,481321	0,49181	0,489196	0,489671	0,485603	0,485064	0,489164

Tabla B.1: Resultados de la prueba control número 1

Prueba Control 2										
Iteración	0	1	2	3	4	5	6	7	8	9
Exactitud	0,514444	0,499	0,582667	0,627333	0,664667	0,646	0,656333	0,655	0,663	0,651333
Precisión	0,598973	0,57884	0,632983	0,649102	0,680788	0,663972	0,674965	0,669018	0,678702	0,664512
Recuerdo	0,519102	0,504405	0,582135	0,623467	0,664626	0,646046	0,655208	0,655053	0,662793	0,648394
Puntaje F1	0,555401	0,538548	0,606177	0,635965	0,672583	0,654837	0,664859	0,661914	0,670605	0,656313
Valores T Crit	0	0,488966	0,333781	0,275995	0,226685	0,207806	0,203541	0,202206	0,20141	0,200505

Tabla B.2: Resultados de la prueba control número 2

Prueba 1										
Iteración	0	1	2	3	4	5	6	7	8	9
Exactitud	0,47	0,51	0,56	0,71	0,61	0,63	0,64	0,626	0,617	0,593
Precisión	0,383333	0,535357	0,527103	0,722341	0,622778	0,601111	0,585833	0,651972	0,631806	0,583739
Recuerdo	0,445	0,486905	0,514444	0,663333	0,615556	0,619444	0,612222	0,622517	0,59245	0,599794
Puntaje F1	0,394986	0,494549	0,515695	0,688337	0,616004	0,609198	0,597733	0,634669	0,608235	0,589306
Valores T Crit	0	0,444965	0,29471	0,234089	0,198662	0,176942	0,168223	0,157114	0,161218	0,157105
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
	0,671	0,679	0,591	0,698	0,667	0,669	0,634	0,703	0,682	0,791
	0,674058	0,653932	0,554102	0,734323	0,677818	0,634102	0,634871	0,698158	0,683689	0,776043
	0,682162	0,658931	0,578256	0,720746	0,665275	0,646966	0,660081	0,696329	0,651173	0,765709
	0,677324	0,653221	0,562867	0,726564	0,670199	0,638862	0,645539	0,696477	0,663551	0,769252
	0,153031	0,152376	0,149696	0,152293	0,151623	0,152794	0,150006	0,150933	0,158009	0,152973
	20	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
	0,744	0,697	0,667	0,74	0,77	0,73	0,74	0,64	0,75	0,73
	0,740589	0,670729	0,694847	0,732222	0,758782	0,671111	0,711667	0,582778	0,767143	0,743333
	0,723924	0,649296	0,677124	0,69754	0,76326	0,664048	0,707143	0,601349	0,753333	0,742222
	0,730888	0,658851	0,683774	0,712509	0,75962	0,665869	0,706091	0,589946	0,758686	0,741707
	0,153491	0,154799	0,155183	0,159781	0,160808	0,161815	0,162328	0,163731	0,166076	0,166605

Tabla B.3: Resultados de la prueba número 1

Prueba 2										
Iteración	1	2	3	4	5	6	7	8	9	10
Exactitud	0,394	0,446	0,452	0,53	0,55	0,58	0,58	0,592	0,638	0,608
Precisión	0,450385	0,575748	0,553585	0,616068	0,622713	0,650049	0,653508	0,635944	0,659659	0,636006
Recuerdo	0,383935	0,45718	0,472881	0,534722	0,544764	0,57731	0,575219	0,588317	0,633068	0,600287
Puntaje F1	0,402297	0,50217	0,507768	0,571155	0,578818	0,610525	0,610878	0,610322	0,645522	0,61718
Valores T Crit	0	0,573999	0,423432	0,314588	0,280972	0,253342	0,212987	0,203241	0,192367	0,178051
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
	0,656	0,642	0,636	0,669	0,635	0,676	0,664	0,651	0,651	0,658
	0,679322	0,657309	0,668227	0,690163	0,648904	0,687337	0,68268	0,671035	0,674634	0,674605
	0,652111	0,638744	0,630347	0,667871	0,641618	0,676801	0,660144	0,647238	0,648756	0,655725
	0,665089	0,647745	0,647925	0,678777	0,645122	0,68181	0,671142	0,658739	0,66093	0,664976
	0,168343	0,168863	0,16245	0,155007	0,160069	0,158592	0,152737	0,152636	0,158457	0,153475
	20	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
	0,679	0,711	0,669	0,685	0,693	0,669	0,698	0,703	0,7	0,73
	0,691573	0,724584	0,67531	0,698208	0,70392	0,676087	0,719359	0,714763	0,705338	0,739105
	0,67562	0,709985	0,667476	0,689066	0,697213	0,666242	0,701732	0,708137	0,70462	0,72722
	0,683412	0,717114	0,671323	0,693546	0,700468	0,671071	0,709708	0,711358	0,704898	0,733083
	0,152995	0,1538	0,153496	0,150315	0,14708	0,148668	0,157343	0,151433	0,151677	0,151191

Tabla B.4: Resultados de la prueba número 2

Prueba 3										
Iteración	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Exactitud	0,398	0,502	0,53	0,526	0,558	0,61	0,574	0,59	0,604	0,642
Precisión	0,605811	0,613021	0,644261	0,586518	0,608836	0,670367	0,609606	0,616303	0,6448	0,672079
Recuerdo	0,410225	0,476771	0,520369	0,522857	0,563672	0,612768	0,582544	0,579145	0,609278	0,642181
Puntaje F1	0,485684	0,532714	0,57289	0,550455	0,584185	0,639026	0,595494	0,596736	0,625912	0,656567
Valores T Crit	0	0,582952	0,407303	0,332099	0,285475	0,242218	0,213261	0,211091	0,188793	0,171454
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
	0,664	0,67	0,649	0,671	0,686	0,683	0,667	0,669	0,658	0,658
	0,687787	0,689428	0,67711	0,691687	0,692605	0,691448	0,687298	0,691582	0,675142	0,672615
	0,65855	0,67065	0,644853	0,668386	0,688344	0,683163	0,66028	0,672715	0,657046	0,658279
	0,672782	0,679465	0,660461	0,679774	0,690195	0,687221	0,673261	0,681864	0,665846	0,665332
	0,175961	0,164572	0,165957	0,157586	0,166009	0,15294	0,158122	0,145956	0,158245	0,153715
	20	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
	0,721	0,694	0,687	0,718	0,683	0,672	0,661	0,713	0,669	0,73
	0,732215	0,695499	0,702769	0,724907	0,696569	0,678882	0,67886	0,723984	0,680676	0,732127
	0,722011	0,681205	0,693564	0,721758	0,683933	0,673019	0,666069	0,707548	0,682891	0,73401
	0,727032	0,688189	0,698046	0,723084	0,690133	0,67586	0,672171	0,715493	0,681622	0,733056
	0,155389	0,154798	0,156833	0,158269	0,146758	0,150598	0,155082	0,154579	0,156229	0,156698

Tabla B.5: Resultados de la prueba número 3

Prueba 4										
Iteración	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Exactitud	0,54	0,55	0,46	0,6	0,56	0,65	0,64	0,615	0,636	0,705
Precisión	0,507302	0,570608	0,439074	0,61881	0,557857	0,604048	0,578889	0,593226	0,661487	0,706348
Recuerdo	0,52	0,534444	0,449841	0,619286	0,585556	0,623175	0,613889	0,614289	0,625294	0,692235
Puntaje F1	0,504486	0,549433	0,433734	0,609043	0,568157	0,605553	0,59126	0,600409	0,639146	0,696987
Valores T Crit	0	0,442413	0,297671	0,232519	0,207742	0,185324	0,175356	0,156838	0,156478	0,156101
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
	0,724	0,694	0,703	0,778	0,677	0,761	0,686	0,723	0,762	0,764
	0,735247	0,725448	0,707348	0,76954	0,669036	0,745974	0,685567	0,737746	0,768019	0,772778
	0,725457	0,692291	0,668518	0,765176	0,678058	0,732247	0,692386	0,743533	0,771045	0,740225
	0,730173	0,707546	0,686095	0,767066	0,672951	0,738907	0,687482	0,739555	0,769158	0,755347
	0,151418	0,155372	0,155359	0,152387	0,148806	0,150474	0,155924	0,156306	0,149676	0,154448
	20	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
	0,713	0,742	0,669	0,703	0,689	0,665	0,708	0,716	0,785	0,7
	0,744966	0,752592	0,634745	0,693748	0,717337	0,689036	0,713484	0,72777	0,797648	0,724493
	0,719496	0,725603	0,65523	0,707305	0,691092	0,673036	0,697798	0,726253	0,781914	0,743645
	0,730974	0,735845	0,643099	0,69952	0,703681	0,678713	0,7044	0,72641	0,789284	0,732026
	0,15401	0,15627	0,155302	0,15354	0,15558	0,156046	0,155777	0,15364	0,154294	0,153571

Tabla B.6: Resultados de la prueba número 4

Prueba 5										
Iteración	1	2	3	4	5	6	7	8	9	10
Exactitud	0,402	0,484	0,54	0,528	0,562	0,62	0,552	0,654	0,608	0,614
Precisión	0,563548	0,59517	0,639192	0,601905	0,638387	0,670164	0,599737	0,689391	0,641189	0,644373
Recuerdo	0,421312	0,462581	0,537111	0,532889	0,565145	0,616738	0,547894	0,638864	0,61633	0,613162
Puntaje F1	0,473472	0,517663	0,582139	0,56423	0,59877	0,641353	0,571926	0,662404	0,628362	0,628235
Valores T Crit	0	0,579843	0,416582	0,330982	0,276332	0,246166	0,214123	0,209311	0,194704	0,177675
	10	11	12	13	14	15	16	17	18	19
	0,627	0,642	0,6	0,684	0,725	0,677	0,618	0,709	0,683	0,672
	0,648402	0,683846	0,623082	0,702958	0,73291	0,699392	0,624835	0,72243	0,694159	0,686641
	0,624438	0,649981	0,606766	0,678929	0,723077	0,679785	0,620188	0,707101	0,6841	0,668433
	0,636049	0,666004	0,614642	0,690516	0,727819	0,689362	0,622442	0,714637	0,688972	0,677251
	0,16844	0,174717	0,168075	0,158416	0,164512	0,158359	0,160533	0,156383	0,152168	0,151723
	20	21	22	23	24	25	26	27	28	29
	0,698	0,698	0,689	0,696	0,662	0,724	0,704	0,696	0,659	0,729
	0,709809	0,712829	0,697093	0,712319	0,676976	0,742799	0,709011	0,707339	0,666609	0,732148
	0,701416	0,703233	0,681919	0,695972	0,66046	0,725289	0,700307	0,695622	0,661969	0,729207
	0,705549	0,70793	0,689312	0,703943	0,668512	0,733864	0,704612	0,701312	0,664181	0,730636
	0,152941	0,152019	0,152634	0,152565	0,155707	0,157671	0,154086	0,155349	0,159615	0,152339

Tabla B.7: Resultados de la prueba número 5

Prueba 6										
Iteración	1	2	3	4	5	6	7	8	9	10
Exactitud	0,453333	0,586667	0,6	0,58	0,7	0,646667	0,646667	0,694	0,799333	0,673333
Precisión	0,385396	0,622713	0,626236	0,630397	0,68369	0,646653	0,643611	0,703665	0,817926	0,680158
Recuerdo	0,411349	0,569841	0,593849	0,591746	0,691468	0,64123	0,643452	0,711193	0,781932	0,679293
Puntaje F1	0,385348	0,590865	0,607009	0,606757	0,687189	0,639424	0,642707	0,707204	0,798839	0,679588
Valores T Crit	0	0,422773	0,293049	0,23361	0,203036	0,1849	0,180174	0,162553	0,159165	0,154809
	10	11	12	13	14	15	16	17	18	19
	0,713333	0,736	0,692	0,713333	0,746	0,687333	0,735333	0,72	0,676667	0,733333
	0,716984	0,739263	0,676813	0,721984	0,779861	0,659242	0,732825	0,71209	0,685371	0,760794
	0,715811	0,744747	0,702173	0,748413	0,750472	0,654541	0,731827	0,701661	0,685025	0,75377
	0,715565	0,739826	0,685584	0,733926	0,763398	0,654328	0,731946	0,706552	0,684059	0,757088
	0,15303	0,158223	0,156321	0,161007	0,157973	0,158753	0,157041	0,160021	0,161918	0,16419

Tabla B.8: Resultados de la prueba número 6

Prueba 7										
Iteración	1	2	3	4	5	6	7	8	9	10
Exactitud	0,502222	0,528889	0,573333	0,595556	0,597778	0,682222	0,686667	0,684445	0,662222	0,676667
Precisión	0,543908	0,589385	0,626955	0,622396	0,611556	0,705536	0,682008	0,684693	0,678484	0,691062
Recuerdo	0,477426	0,535969	0,579305	0,605686	0,592998	0,690175	0,678479	0,676477	0,660456	0,679955
Puntaje F1	0,504025	0,559986	0,600969	0,613773	0,60191	0,697647	0,68021	0,680466	0,669172	0,685227
Valores T Crit	0	0,489667	0,334904	0,268908	0,227647	0,204812	0,187541	0,174109	0,176241	0,162464
	10	11	12	13	14	15	16	17	18	19
	0,725333	0,735556	0,724444	0,710222	0,703111	0,661333	0,719778	0,727556	0,709111	0,740667
	0,737041	0,759088	0,727809	0,720628	0,704117	0,667356	0,721895	0,725558	0,71047	0,747516
	0,722699	0,724956	0,723046	0,706977	0,707567	0,660726	0,720882	0,729942	0,71159	0,741084
	0,729725	0,741343	0,725227	0,713516	0,705811	0,663926	0,721314	0,727581	0,710978	0,744218
	0,160109	0,161185	0,159325	0,151661	0,156729	0,155472	0,155096	0,158555	0,153197	0,154504

Tabla B.9: Resultados de la prueba número 7

Prueba 8										
Iteración	1	2	3	4	5	6	7	8	9	10
Exactitud	0,524444	0,575556	0,577778	0,626667	0,64	0,671111	0,666667	0,728889	0,697778	0,686
Precisión	0,585031	0,587143	0,574156	0,64407	0,662847	0,678515	0,686674	0,748134	0,697076	0,687933
Recuerdo	0,496399	0,556715	0,567476	0,624523	0,631845	0,660746	0,667948	0,731919	0,693341	0,681487
Puntaje F1	0,534917	0,571237	0,570756	0,633919	0,646314	0,669262	0,6771	0,739716	0,69508	0,684644
Valores T Crit	0	0,49356	0,324118	0,273175	0,223793	0,201146	0,19149	0,172089	0,178024	0,162661
	10	11	12	13	14	15	16	17	18	19
	0,694889	0,711556	0,726	0,695111	0,742444	0,697778	0,710222	0,719111	0,712667	0,726889
	0,710068	0,718178	0,727279	0,703799	0,747608	0,698961	0,711981	0,716318	0,718914	0,734976
	0,698943	0,71447	0,722058	0,693751	0,741092	0,697445	0,704998	0,712213	0,706547	0,732224
	0,704098	0,716238	0,724546	0,698305	0,74423	0,698173	0,708295	0,71416	0,712431	0,733579
	0,16326	0,158307	0,156783	0,159389	0,155557	0,156463	0,154861	0,155366	0,155641	0,160803

Tabla B.10: Resultados de la prueba número 8

Prueba 9										
Iteración	1	2	3	4	5	6	7	8	9	10
Exactitud	0,471111	0,542222	0,611111	0,568889	0,617778	0,633333	0,68	0,697778	0,693333	0,689333
Precisión	0,590537	0,618308	0,648816	0,613628	0,633798	0,656319	0,708252	0,716889	0,69429	0,69974
Recuerdo	0,465561	0,550367	0,605999	0,575656	0,619627	0,630357	0,678004	0,6904	0,694844	0,68778
Puntaje F1	0,515371	0,581841	0,626052	0,593798	0,626507	0,642763	0,692531	0,703242	0,694429	0,693536
Valores T Crit	0	0,49884	0,333538	0,273921	0,221749	0,209853	0,191926	0,180342	0,176488	0,162002
	10	11	12	13	14	15	16	17	18	19
	0,708889	0,692889	0,691111	0,692889	0,732667	0,702444	0,724444	0,731111	0,750667	0,736889
	0,701014	0,703764	0,706821	0,6844	0,738227	0,711655	0,73007	0,734625	0,755495	0,727547
	0,70659	0,689872	0,698655	0,677936	0,74079	0,704741	0,723704	0,72983	0,750528	0,731905
	0,703744	0,696643	0,702523	0,681118	0,739374	0,708108	0,726812	0,732165	0,752882	0,729669
	0,162644	0,15936	0,158752	0,157991	0,162916	0,155652	0,159511	0,16135	0,161744	0,157062

Tabla B.11: Resultados de la prueba número 9

Prueba 10										
Iteración	1	2	3	4	5	6	7	8	9	10
Exactitud	0,506667	0,566667	0,578444	0,597778	0,653556	0,615556	0,648889	0,633333	0,675556	0,737778
Precisión	0,573861	0,616315	0,63363	0,641947	0,678977	0,631255	0,670829	0,647434	0,693033	0,743933
Recuerdo	0,489314	0,567556	0,574709	0,607858	0,655626	0,613297	0,654055	0,63098	0,68182	0,733734
Puntaje F1	0,522321	0,590574	0,601771	0,623902	0,666915	0,622018	0,662135	0,638918	0,687059	0,73868
Valores T Crit	0	0,3148	0,1928	0,2558	0,2224	0,285	0,2928	0,3006	0,329	0,2972
	10	11	12	13	14	15	16	17	18	19
	0,695556	0,695556	0,716889	0,668889	0,680889	0,650222	0,672889	0,704445	0,684444	0,706667
	0,704469	0,705492	0,724881	0,689621	0,709441	0,666999	0,683922	0,718402	0,692058	0,726231
	0,699119	0,693544	0,704661	0,671983	0,684041	0,645577	0,6741	0,709779	0,677367	0,710434
	0,701691	0,699418	0,714479	0,680318	0,696319	0,655917	0,678716	0,713952	0,684401	0,718169
	0,2988	0,312	0,3042	0,2766	0,3006	0,294	0,305	0,2868	0,3088	0,2634

Tabla B.12: Resultados de la prueba número 10