UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

PROYECTO DE FIN DE CARRERA

ESTUDIO DE HERRAMIENTAS PARA TOPIC DETECTION EN COMENTARIOS Y POST DE FACEBOOK

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN:

SISTEMAS DE TELECOMUNICACIÓN

Autor: David García Rodríguez
Tutor: Ángel Cuevas Rumín
Año: 2014/2015

Título: Estudio de herramientas para Topic Detection and Tracking
Autor: David García Rodríguez
Director: Ángel Cuevas Rumín

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ___ de___ de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la **CALIFICACIÓN** de

Fdo.: PRESIDENTE **Fdo.: VOCAL** **Fdo.: SECRETARIO**

Especial mención a mis padres,
Argimiro y Mª Remedios,
y a mi hermano, Víctor.

Agradecimientos

Hace muchos años atrás desde que comencé Ingeniería Técnica de Telecomunicación en la especialidad de Sistemas de Telecomunicaciones.
Y después realicé la adaptación al grado en Sistemas de Telecomunicación.
En aquel tiempo en la Universidad viví muchas alegrías y éxitos, pero también sufrimientos y decepciones.
El camino fue largo y duro y después de aprobar todas las materias y asignaturas encontré un trabajo en el cual sigo y muy contento pero hizo que se retrasara algo aún fundamental para acabar, el proyecto de fin de carrera. Hasta que llega el momento de poner el último punto a esta memoria y atravesar el túnel en el que muchas veces ves la luz del final pero justamente es el tramo mas duro.

En la universidad conocí buenos amigos y compañeros, incluso mi hermano también decidió realizar la misma ingeniería y junto a todos ellos he pasado muy buenos momentos. Juntos sufriendo días interminables de prácticas y clases. Con algunos de ellos aún quedó para rememorar viejas clases.

También a los profesores que supieron enseñarme tantas cosas interesantes a lo largo de esta etapa.

A mis padres, por su INFINITA paciencia, y apoyarme en todo momento, al igual que mi hermano que nos hemos apoyado mutuamente en esta etapa en la universidad.

A mi tutor, Ángel Cuevas, por proponerme este proyecto cuando en un principio le pregunte por otro distinto, también por ser paciente conmigo y ayudarme con mis dudas.

A los creadores y programadores de las herramientas de detección de palabras clave o temas recurrentes en textos.

Y a los futuros contribuyentes del Proyecto por seguir desarrollando esta investigación, además de darles mí apoyo y animo.

Muchas gracias a todos.

Resumen

Las redes sociales se han vuelto cada vez más importantes en los últimos años. Entre ellas destaca Facebook y Twitter, donde los usuarios comparten opiniones y otros contenidos de forma pública.

Dichas redes contiene grandes cantidades de información oculta a simple vista. En este trabajo se pretende trabajar con datos obtenidos de Facebook y analizarlos para obtener información relevante (topics o palabras clave) y obtener conclusiones y quizás alguna idea que pueda adaptarse a Twitter. Para ello se va a hacer uso del análisis semántico, en pos de poder asociar los temas o noticias que dejan los usuarios sobre un tema o marca o comentarios de la propia empresa o marca en su cuenta oficial con temas más generales (o meta-topics). Cabe destacar que, aunque cada vez existe un mayor número de trabajos dedicados a analizar este tipo de redes, el análisis semántico (como el llevado a cabo en este trabajo) de textos es todavía escaso, siendo lo más cercano trabajos parecidos a este sobre las técnicas de Topic Detection.

Dada la cantidad de información que se maneja hoy en día en los medios de comunicación y especialmente en la red, resulta imprescindible filtrar los datos que se obtienen (noticias, reflexiones u opiniones), para más tarde tratar toda esa información de la manera más adecuada, ordenarla y tenerla a disposición para sacar el máximo provecho de su contenido. Realizar todo este proceso sin ayuda de procesos automáticos sería costoso e *imposible*.

Este trabajo realiza un estudio con diferentes herramientas para comprobar el rendimiento de ellas sobre la fuente de origen del texto, en este caso Facebook, y comprobar las diferencias o similitudes entre ellas.

Palabras Clave

Facebook, red social, meta-topic, keyword, Topic Detection, análisis semántico

Abstract

Social networks have become increasingly important in recent years time. Among them is Facebook and Twitter, where users share opinions and other public contents.

These networks contain large amounts of information hiding in plain sight. In this project aims to work with data from Facebook and analyze them to obtain relevant information (topics or keywords) and get conclusions and perhaps some idea that can adapt to Twitter. To do this is to make use of semantic analysis in pursuit of associating topics or news that let users on a topic or brand or comments from the company or brand in their official has more general topics (or meta-topics). Note that, although an increasing number of works devoted to analyzing such networks exist, semantic analysis (as done in this study) of texts is still rare, being the closest thing works like this on Topic Detection techniques.

Given the amount of information used today in the media and especially in the network, it is essential to filter the data obtained (news, thoughts or opinions), later to treat all such information in the most appropriate way , order it and have it on hand to make the most of its contents. Perform this process without the help of automated processes would be expensive and impossible.

This paper makes a study of different tools to check the performance of them on the source text, in this case Facebook, and check the differences or similarities between them.

Key words

Facebook, social network, meta-topic, keyword, Topic Detection, semantic analysis

Índice

Índice Figuras

Índice Ecuaciones

Capítulo 1

Introducción, Motivación y Objetivos

Este capítulo muestra un resumen sobre el ámbito en el que se ha desarrollado este Proyecto de Fin de Carrera, su objetivo, y cómo está organizado este documento.

1.1. Entorno del Proyecto de Fin de Carrera

Facebook es un sitio web de redes sociales. Originalmente era un sitio para estudiantes de la Universidad de Harvard, pero se abrió a cualquier persona con una cuenta de correo electrónico. Su lanzamiento fue en 2004 creado por Mark Zuckerberg. Su infraestructura principal está formada por una red de más de 50 000 servidores que usan distribuciones del sistema operativo GNU/Linux usando LAMP. LAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Linux, el sistema operativo; En algunos casos también se refiere a LDAP.
- Apache, el servidor web;
- MySQL/MariaDB, el gestor de bases de datos;
- Perl, PHP, o Python, los lenguajes de programación.

El funcionamiento de Facebook es similar al de cualquier otra red social, aunque esta oración deberíamos formularla al revés, ya que es esta la red social que marca los antecedentes y las condiciones que deben cumplir las demás, y por tanto una de las más utilizadas por los usuarios, razón por la que se ha optado por adoptarla como foco de información para el desarrollo del trabajo realizado.

En Facebook existen dos tipos de cuentas: las de cualquier usuario particular normal y corriente y la que pueden abrir las empresas.
En las cuentas de empresas se permite expresar breves opiniones o comentar cosas por usuarios particulares lo que lo convierte en una fuente inagotable de información de opinión de gran variedad de usuarios de distinta edad, género y condición social [1].
Los usuarios suelen tener gustos y opiniones variadas por eso el proceso de extracción de información específica y útil sobre los gustos del usuario se vuelve muy complejo.

Este trabajo pretende obtener de las publicaciones de los usuarios en una determinada empresa o temática un tema general que comprenda varios temas (meta-topic o keywords) que identifique la página de la marca.

Para ello se realiza un estudio con varias herramientas de tratamiento de textos para Topic Detection. Se realizará un análisis cuantitativo y cualitativo para compararse entre ellas y obtener resultados de rendimiento. Con la información se puede investigar cómo se distribuye la información dependiendo de la temática e investigar las tendencias que predominan en cada momento. Todo esto puede aplicarse para llevar a cabo campañas de marketing más efectivas por parte de las empresas o productos mejor orientados a sus destinatarios.

1.2. Cometido del Proyecto de Fin de Carrera

Las redes sociales en formato digital han aparecido y se están extendiendo rápidamente, obteniendo mucha importancia a nivel social.
La cantidad de información que contienen es inmensa y es necesario filtrar y tratarla de manera sencilla y eficaz.
La posibilidad de realizar un estudio para obtener los mejores resultados en distintas herramientas y comparar su resultado es la motivación que ha hecho de este proyecto una opción a tomar.
Por otro lado se ha optado por el análisis semántico y la selección de keywords y topics, porque se busca organizar la información de una forma cercana para el usuario, analizándola por su contenido y no tanto por su forma, por lo que el análisis semántico se vuelve indispensable.

El número, la variedad y la complejidad de los proyectos Topic Detection se ha incrementado en los últimos años rápidamente, lo que hace que los procesos de desarrollo tengan que estandarizarse para conseguir proyectos que se puedan integrar, reutilizar e intercambiar en el futuro.
Los proyectos de Topic Detection se están convirtiendo en proyectos de ingeniería por lo que la motivación del trabajo es realizar un estudio de las herramientas actuales para revisar los resultados para incorporar el punto de vista de la ingeniería.

1.3. Objetivos y Metodología

El objetivo principal de este proyecto es realizar una valoración de diferentes herramientas de topic detection en comentarios extraídos de Facebook, realizar experimentos y obtener resultados para realizar una comparación y obtener una conclusión final. Para llegar a este objetivo se han realizado los pasos siguientes:

1.- Extraer los comentarios de una colección de ficheros que contienen además de cada comentario su autor y fecha de publicación.

2.- Agrupar los comentarios en ficheros de grupos de 1000, 2000, 5000, 10000 y 15000 comentarios aleatoriamente por cada empresa.

3.- Obtener los resultados para conjunto de agrupaciones de comentarios para todas las herramientas finales (Maui Indexer, Stanford NLP, Wandora y RapidMiner).

4.- Obtener los keywords o topics relevantes de cada resultado.

5.- Realizar comparación cuantitativa por tiempo de ejecución y volumen.

6.- Calcular medidas para realizar una comparación cualitativa de los resultados.

7.- Valoración final de las herramientas.

1.4. Estructura de la memoria

Inicialmente en un primer capítulo se realiza una introducción al estado actual en el tratamiento de textos. Analizar los diferentes modelos de procesado y clasificación e introducción a los sistemas de información y sistemas TDT.

A continuación un listado de las herramientas más importantes o más relevantes en el panorama actual de tratado de textos.
Inmediatamente después un análisis de las herramientas usadas en el trabajo y los experimentos sobre su funcionamiento y procesado de los comentarios.

Se comenta en otro capítulo las medidas y bases para la evaluación de los resultados para después en el capitulo siguiente mostrar los resultados obtenidos de los experimentos y compararlos cualitativa y cuantitativamente de manera independiente como de manera conjunta enfrentada a las herramientas entre sí.

Obtener conclusiones y divagar sobre trabajos futuros para finalizar.

Capítulo 2

Estado del Arte

En este apartado se introduce al estudio de Data Mining así como sus diferentes partes y los sistemas de Recuperación de Información (RI).
Los modelos de procesamiento de textos Text Mining y métodos de validación con respecto al análisis de redes sociales y el uso final en el proceso de Topic Detection and Tracking (TDT).

Gracias a la evolución de los ordenadores, la aparición de smartphones y la extensión de la red cada vez existe más información en formato digital, haciendo cada vez más notable la necesidad de mejorar el acceso a la información almacenada, con sistemas más rápidos y eficientes. Y tener un acceso más ordenado y directo a esta información por parte del usuario.

Aunque existen motores de búsqueda estas aplicaciones son muy lentas a la hora de procesar grandes cantidades de texto de gran longitud, y muchas veces no reconocen la información presente en los documentos. Por lo que el estudio de clasificar textos por temática o palabras clave se hace necesario.

2.1. Data Mining

Data Mining es el proceso de descubrir relaciones, patrones y tendencias, mediante el procesado de grandes cantidades de datos almacenadas en repositorios, utilizando tecnologías de reconocimiento de patrones, junto con técnicas matemáticas y estadística [2].

Realmente se trata de una etapa dentro de un proceso mayor llamado extracción de conocimiento en bases de datos (Knowledge Discovery in Databases o KDD). Lo que en verdad hace la Minería de Datos es reunir las ventajas de varias áreas como la Estadística, la Inteligencia Artificial, la Computación Gráfica, las Bases de Datos y el Procesamiento Masivo de Datos, principalmente usando como materia prima fuentes de información como las bases de datos o las ontologías.

Las principales fuentes de datos utilizadas son ficheros planos, bases de datos relacionales, base de datos de transacciones, bases de datos espaciales, series de tiempo, textos, literatura e incluso multimedia (video, audio) o datos en Internet. De ellos se pretende extraer información que abarca desde caracterización de entidades, discriminación, clasificación, agrupamiento, descubrir tendencias, calcular la desviación, detección de datos anómalos, etc.
Las técnicas de minería de datos son muy utilizadas en distintas áreas y tienen diversas aplicaciones. Evidentemente son muy útiles en investigación científica, pero también en telecomunicaciones o en la banca.

Las técnicas de Data Mining suelen dividirse en 5 pasos, de las cuales derivan después los pasos en el proceso TDT:

1. *Extracción de datos*: Consiste en obtener el conjunto de datos a analizar.

2. *Preprocesamiento de datos*: Preparar el conjunto de datos para ser analizados.

3. *Generación de los modelos*: Es el punto más importante del proceso. El modelo se crea para buscar los patrones en los datos.
   Normalmente se utilizan técnicas de Machine Learning u otras técnicas estadísticas para generar el modelo [2].

4. *Validación de los modelos*: La validación puede ser diferente según el modelo. Es usual utilizar la validación para clasificadores [2].

5. *Aplicación del modelo*: Obtener resultados de los datos y poder predecir el comportamiento de nuevas entradas.

2.1.1. Extracción de datos

En un primer momento los datos que se están analizando en este trabajo provienen de Facebook, aunque se pueden extender las conclusiones a Twitter, ya que son comentarios de usuarios que son muy parecidos a los que se pueden obtener de Twitter.
También existen aplicaciones para el análisis de las Redes Sociales, que permite a los investigadores extraer la información Facebook (Facebook API [3]).

2.1.2. Preprocesamiento de datos

El proceso de preprocesado consiste en una serie de pasos para simplificar la información contenida en los datos. Reducir el volumen de datos a un conjunto donde la información sea más relevante
Normalmente en este punto se encuentran los siguientes pasos:

1. Eliminar las Stop-Words (palabras de parada) y los caracteres especiales de las frases.
2. Generar una matriz término-documento con las keywords.
3. Utilizar una técnica de selección de características para elegir las palabras más relevantes para el análisis y reducir así el espacio de búsqueda.

2.1.3. Modelos Clásicos de Data Mining

Las técnicas de Machine Learning que se utilizan principalmente en Data Mining son técnicas de Clasificación y Clustering [2].
Las técnicas de clasificación buscan patrones dentro del conjunto de datos de forma supervisada, es decir, utilizan datos ya etiquetados para generar los modelos [2].
Las técnicas de clustering buscan los patrones de forma ciega, sin un etiquetado previo, y generan los modelos a partir de métodos estadísticos [2].

2.1.4. Sistemas de agrupamiento (Clustering)

Los sistemas de agrupamiento de documentos tienen la tarea de clasificar los documentos según sus propiedades intrínsecas en varios grupos (clusters). Mientras que en los sistemas de clasificación los documentos se clasifican según semejanza o relevancia con ciertas clases previamente especificadas, en los sistemas de agrupamiento se trata de buscar característica que permitan separar los documentos en grupos basándose en las propiedades internas de la colección. Idealmente los grupos deben estar completamente separados, pero algunas veces el solapamiento entre grupos es inevitable. El correcto funcionamiento de estos sistemas depende de las propiedades estadísticas de la colección. Generalmente estos sistemas se aplican en colecciones estáticas, aunque también se puede aplicar a colecciones que se incrementan en el tiempo.
Este tipo de algoritmos divide un conjunto de elementos en grupos que satisfacen las condiciones de homogeneidad (alta similitud entre los elementos de un mismo grupo) y separación (baja similitud entre elementos de grupos distintos).

Entre los algoritmos de agrupamiento más utilizados están el algoritmo de agrupamiento jerárquico, el de las k-medias y los mapas auto-organizativos.

Agrupamiento jerárquico

El agrupamiento jerárquico ordena los elementos de una población en base a un árbol de distancias que refleja la similitud que hay entre los elementos y grupos. Los algoritmos aglomerativos se inician asignando cada elemento individual a un grupo, se calculan las distancias de todos contra todos y los dos elementos más similares se unen para formar un nuevo grupo. Finalizado este proceso, se vuelve a recalcular la matriz de distancias considerando el nuevo grupo y se vuelven a unir los dos elementos más similares. Este proceso se repite hasta que se unen los dos últimos grupos. Por el contrario, los algoritmos divisivos comienzan con un solo grupo que engloba al conjunto total de elementos, y en cada paso se subdivide en grupos de menor tamaño hasta llegar a los elementos únicos [4]. Presentan las ventajas de que es una metodología simple y los resultados pueden ser fácilmente visualizados. Sin embargo, también pueden presentar ciertos problemas como es el que, al ir creciendo en tamaño, los vectores representativos de un grupo puede que no se asemejen a los elementos englobados en el mismo. Además, con este tipo de técnicas si se comete un error de asignación en estados iníciales del proceso este se arrastrará hasta el final.

K-medias

El algoritmo de k-medias es un algoritmo de agrupamiento clásico que divide un conjunto de elementos en un número predefinido de grupos. Este método requiere por tanto especificar el número de grupos (k) a priori. Dado un valor de k, el algoritmo de k-medias divide el conjunto de datos en k grupos minimizando la siguiente función:

$$E = \sum_{i=1}^{k} \sum_{O \in C_i} |O - \mu_i|^2$$

Ecuación 1: K-medias

donde O es un elemento en el grupo C_i y μ_i es el centroide (media de los elementos de un grupo) del grupo C_i. De forma resumida, este algoritmo trabaja asignando los datos de forma aleatoria a k grupos. A continuación los centroides de cada grupo son calculados y cada dato es asignado a su centroide más cercano formando k nuevos grupos. Este proceso es repetido hasta que se alcanza algún criterio de parada, usualmente cuando las variaciones de los centroides entre distintas iteraciones sean muy pequeñas o cuando se alcanza un número prefijado de las mismas.
El algoritmo de k-medias es rápido y sencillo, pero presenta también ciertas limitaciones para el análisis de datos de expresión como, por ejemplo, que normalmente el número de grupos no se conoce a priori. Además, este algoritmo no garantiza que se alcance un mínimo global en la función de optimización, por lo que los resultados obtenidos en muchas ocasiones pueden no ser óptimos.

Mapas auto-organizativos (SOM)

Los mapas auto-organizativos constituyen un método de agrupamiento basado redes neuronales desarrollado por Teuvo Kohonen. Un SOM asigna los elementos a una serie de vectores, o neuronas, dentro de una red que presenta una topología predefinida. El algoritmo de SOM fue introducido para análisis de datos de expresión por Tamayo et al. [5] y Toronen et al. [6] y tiene algunas propiedades que lo hacen interesante para este tipo de análisis: facilita la visualización e interpretación de datos multidimensionales en espacios usualmente bidimensionales, organiza los grupos de forma que los más cercanos en la red son los más parecidos y es relativamente más robusto al ruido en los datos que otros algoritmos como el de k-medias. Las desventajas de este método es que requiere determinar a priori el tamaño y la estructura del mapa, aunque este parámetro no es tan crítico como establecer el número de grupos en el algoritmo de k-medias.

2.1.5. Sistemas de Clasificación

La diferencia entre los algoritmos de clasificación y los de agrupamiento radica en que los primeros conocen a priori el número de grupos que se van a formar y utilizan esta información mientras que los segundos no. Debido a esto se considera a los algoritmos de clasificación como algoritmos de aprendizaje supervisado (es decir, que cuentan con información previa que les ayuda a resolver el problema).

Formalmente se pueden definir como una función en la que dada un conjunto de instancias del problema a resolver, devuelve la categoría a la que pertenecen (de un conjunto de categorías predefinidas). A pesar de necesitar conocer previamente el número de categorías son algoritmos muy utilizados y de gran utilidad. Incluso pueden combinarse con otros algoritmos que deducen el número de clases existentes dentro de un conjunto de instancias del problema (selección del rango de factorización).

Árboles de decisión

Los sistemas de aprendizaje basados en árboles de decisión son quizás el método más fácil de utilizar y de entender y por tanto el más aplicado. Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos, y son especialmente apropiados para expresar procedimientos médicos, matemáticos, lógicos, etc.
Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y, siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar.
Estos algoritmos se llaman algoritmos de partición o algoritmos de ”divide y vencerás”, donde la elección del criterio de partición puede llevar a un buen o mal resultado.

Naive Bayes (NB)

El clasificador considera cada característica como independiente del resto. Cada una de ellas contribuye a la información del modelo. Se basa en la Ley de Probabilidad de Bayes. [7]
La técnica de Naive Bayes es particularmente adecuada cuando la dimensionalidad de las entradas es alta. A pesar de su simplicidad, Naive Bayes a menudo puede superar a los métodos de clasificación más sofisticados.

2.1.6. Validación del Modelo

La evaluación de los modelos, en general, es muy variada. Normalmente se realiza una evaluación basada en las métricas de Precisión y Recall.
Precisión se utiliza para medir cuando una instancia que no pertenece al conjunto de clases se clasifica como parte del conjunto de la clase.
Recall mide la situación en la que una instancia está debidamente tipificada de acuerdo a su clase. F-measure es una métrica que equilibra estas medidas.
En capítulos siguientes se explicará más detalladamente estas medidas.

2.1.7. Perspectiva Semántica

Una de las aplicaciones más actuales del análisis de datos está enfocada a la comprensión semántica. Dentro de este campo destaca específicamente la web semántica [8], que es un enfoque reciente utilizado para proporcionar un nuevo tipo de uso de la web. Este enfoque se basa en las relaciones semánticas que existe entre los diferentes conceptos que se pueden encontrar en Internet. Con esta información, se trata de mejorar la experiencia del usuario mejorando los sistemas de acceso a la información. En nuestro objetivo este punto es primordial ya que los datos origen son comentarios web en una red social.

2.2. Text Mining

La Minería de Textos o Text Mining tiene como objetivo examinar una colección de documentos no estructurados escritos en lenguaje natural y descubrir información no contenida en ningún documento individual de la colección; en otras palabras, trata de obtener información sin haber partido de algo ([9]). Aunque se apoya en técnicas de minería de datos ([10]) al trabajar con textos, se invierte un mayor porcentaje del esfuerzo en el preprocesado de la colección de documentos, así se puede decir que la minería de textos es un área multidisciplinaria basada en la recuperación de información, minería de datos, aprendizaje automático, estadísticas y Procesamiento Natural de Lenguaje (NLP).

2.2.1. Introducción a procesamiento de texto

Existen las bases de Datos Relacionales (BDR) que se encargan de buscar un campo que contenga exactamente las palabras de la consulta y también existen los sistemas de procesamiento de documentos que busca las palabras de la consulta en cualquier parte del texto.
Mientras que en las BDR tienen un tipo de datos fijo, y en el caso de que sea texto, éste posee una longitud determinada, en los sistemas de procesamiento de documentos los campos de búsqueda (metadatos del documento o partes de un documento) son textos sin una longitud predeterminada.
Existen diferentes técnicas de procesamiento de documentos y aplicaciones que las practican. Estas aplicaciones se puede clasificar en dos familias: los sistemas de Recuperación de Información (RI) y los sistemas de Extracción de Información (EI).
La tarea principal de un RI consiste en buscar los documentos relevantes dentro de la colección que más se asemejen a la consulta del usuario y devolver estos al usuario. Si lo permite el sistema, los documentos recuperados se ordenarán según un valor de relevancia establecido por el sistema.
En este estudio se realiza el proceso a revés, clasificar texto (comentarios de usuarios) en temas o palabras clave. Pero es necesario entender el funcionamiento de un sistema RI.
En un sistema de RI cada fichero o documento se ve como una secuencia de posibles palabras o términos significativos.

Las técnicas de Procesamiento de Lenguaje Natural (NLP) son básicas en los sistemas EI, pero su uso en sistemas de RI se está extendiendo más, ya que se pueden aplicar bien en el lenguaje de consulta del usuario, como en la representación y clasificación interna de los documentos.

En 1997 surgió una nueva iniciativa en el campo de los sistemas de RI, los sistemas de seguimiento y de detección de sucesos en noticias de actualidad. A esta iniciativa se le denominó Topic Detection Tracking (TDT). En siguientes apartados completamos esta información.

2.2.2. Sistemas de Recuperación de la Información

Un sistema de Recuperación de Información (RI) es un sistema de procesamiento de documentos que trata de recuperar de una colección de documentos aquellos que se asemejan más a la consulta del usuario. Un sistema de RI se encarga tanto de la recuperación de documentos como de su almacenamiento y organización, al igual como ocurre en los Sistemas de Gestión de Bases de Datos. Pero hay que tener en cuenta que mientras en estos sistemas, los datos están totalmente estructurados, organizados expresamente para ser recuperados por el gestor de la base de datos, en los sistemas de RI los datos pueden ser de cualquier tipo y de cualquier longitud.

Un sistema de recuperación de datos simplemente comprueba si un dato existe en un fichero, es decir, busca documentos que casan con una palabra, mientras en los sistemas de RI se seleccionan aquellos documentos que coinciden parcialmente (búsqueda aproximada) o totalmente (búsqueda exacta) con los términos de la consulta del usuario, creándose una lista de documentos que en el caso de búsqueda aproximada se puede ordenar según un índice de relevancia entre la consulta y el documento.

La mayoría de sistemas de RI permiten que las consultas se expresen como una lista de palabras que el usuario espera encontrar en el documento y que para él lo caracterizan. Algunos sistemas permiten la utilización de operadores (principalmente boléanos), palabras o cadenas con comodines. Sin embargo pocos sistemas incluyen la búsqueda por estructura del documento puesto que esto requiere que el usuario conozca de antemano la estructura del documento, y además como cada colección de documentos posee su propia estructura, la tarea de reconocimiento de las distintas estructuras posibles añade más complejidad a los sistemas de indexación y recuperación.

Las conferencias TREC (Text Retrieval Conferences), con carácter anual por el National Institute of Standard and Technology de los Estados Unidos, están centradas en el desarrollo de los sistemas de RI. En ellas se ha comprobado que su efectividad está muy ligada a la consulta que hace el usuario y a la elección de los términos de indexación de los documentos. Por ello en las investigaciones actuales se está trabajando con el propósito de llegar a la utilización del Lenguaje Natural como lenguaje de consulta.

La gran cantidad de documentos almacenados actualmente en los ordenadores y la longitud de éstos son los dos grandes retos de los sistemas RI actuales, que dan lugar a que las aplicaciones desarrolladas teóricamente y probadas en pequeñas colecciones de documentos cortos, al aplicarse a los grandes repositorios de documentos completos existentes en la actualidad, no den los resultados esperados. Esta problemática se está estudiando en las últimas conferencias TREC y la investigación en la evaluación de buscadores web. [11]

Arquitectura de un sistema de RI

Según Shatkay et al. ([12]), un sistema de extracción de la información tiene tres o cuatro fases principales. La primera fase consiste en la tokenización, dividir el documento en bloques básicos. Estos bloques suelen ser palabras, oraciones o párrafos, en raras ocasiones se elige tener unos bloques más grandes (como capítulos o secciones). La segunda fase consiste en el análisis morfológico y léxico, asignar etiquetas PoS (Part of Speech) a las palabras, creación de sintagmas básicos (nominales o verbales) y desambiguación de palabras o expresiones. La tercera fase trata del análisis sintáctico, estableciendo la conexión entre las diferentes partes de cada oración, explicado en una sección previa. La cuarta fase consiste en el análisis de dominio, donde se combina toda la información extraída en las fases anteriores para describir las relaciones entre las distintas entidades.

Contemplaremos la investigación en recuperación de la información desde una perspectiva global, teniendo en cuenta tres características generales que constituyen la esencia de su evolución y existencia:

1. La recuperación de información es un campo interdisciplinar. A pesar de que son muchas las disciplinas directamente relacionadas (Informática, Ciencia de la Información, Documentación, Lógica, Lingüística, Psicología, Inteligencia Artificial, etc.), todas comparten el mismo objetivo: facilitar la búsqueda y obtención de información relevante que satisfaga las necesidades de información de los usuarios.

2. La investigación en recuperación de la información ha estado muy influida por la evolución y los avances producidos en las tecnologías de la información que, necesariamente, han introducido nuevos enfoques, procedimientos y métodos en la organización, almacenamiento y acceso a la información, así como en el uso de nuevos sistemas de recuperación de información.

3. La recuperación de la información ha desempeñado un papel esencial en la evolución de la Sociedad de la Información. Hay que recordar que la recuperación de información surge para buscar soluciones y dar una respuesta al problema de la explosión de información científica; en la actualidad, la World Wide Web, como medio de acceso a la información más utilizado, y a la extensión globalizada de las redes sociales juntamente con la facilidad para poder publicar en ellas ha provocado que uno de los principales problemas a los que se enfrenta cualquier persona es cómo localizar información pertinente ante el exceso de información existente. [13]

Un sistema RI generalmente se compone de una aplicación que se encargue de la organización, selección y presentación de documentos.
Esta aplicación está a su vez compuesta de:

- Un gestor del repositorio con un sistema de indexación de los documentos.
- Un proceso de emparejamiento de la consulta con cada uno de los documentos, que devuelve la semejanza entre cada documento y la consulta.
- Un proceso que organice los documentos relevantes, y los represente en una lista para que el usuario pueda extraer de esa lista los documentos que desea.

Modelos de Recuperación de Información

Uno de los problemas más importantes en los sistemas de RI es cómo discernir entre los documentos relevantes o no. Los sistemas de RI clásicos utilizan técnicas de búsqueda booleanas y de reconocimiento de patrones. Estas técnicas llamadas de búsqueda exacta son técnicas muy restrictivas.
Debido a los problemas que plantean los sistemas de búsqueda exacta se han desarrollado técnicas de búsqueda basadas en la información estadística, las denominadas técnicas de búsqueda aproximada. El sistema de búsqueda aproximada no hace un emparejamiento exacto y permite que el usuario especifique la importancia de cada uno de los términos.
La realización de las consultas resulta más compleja en estos sistemas.

Sistemas de RI de búsqueda exacta

En los sistemas de RI de búsqueda exacta se utiliza una correspondencia exacta entre los términos de las consultas y de los documentos.
En este sistema el primer documento de la lista no tiene por qué ser más interesante para el usuario, simplemente es el primero que ha encontrado relacionado con la consulta. Es decir, todos los documentos de la lista tienen la misma relevancia, no se tiene en cuanta la frecuencia de aparición, ni el orden o importancia de los términos de la consulta.

Destacan dos modelos de sistemas de RI por búsqueda exacta:

Búsqueda de Patrones

Los sistemas de RI por búsqueda de patrones utilizan técnicas de reconocimiento de patrones.
En estos sistemas la consulta puede ser una colección de palabras, cadenas con comodines o expresiones regulares.
El sistema intenta buscar los documentos que contengan el patrón de la consulta.

Indexación Booleana

Se trata de uno de los modelos de recuperación de información más simples que se conocen. Se fundamenta en el álgebra de Boole y en la teoría de conjuntos.
El problema de este modelo es que si encuentra una serie de documentos, no sabe ordenarlos según la relevancia que tenga cada uno.

El usuario proporciona un término o una combinación booleana de términos. De esta manera el grado de relevancia de un documento es binario, es decir, una información determinada es relevante o no lo es. El resultado es el conjunto de todos los documentos de la base de datos que satisfagan las restricciones de la consulta

Cuando un usuario realiza una consulta, se busca en la estructura de índices y se devuelven los documentos que contienen el término o combinación de términos que se buscan. Existen varios métodos para crear índices y usarlos. Aunque este tipo de estrategias tienen la ventaja de ser muy rápidas, tienen algunas limitaciones:

• El número de documentos recuperado puede llegar a ser prohibitivamente grande.

• Una parte substancial de los documentos recuperados puede ser irrelevante para el usuario.

• Muchos documentos que sí son relevantes pueden no ser devueltos.

El modelo booleano es muy popular, sobre todo debido a su sencillez y a que es una de las primeras ideas que surgen en el diseño de un sistema RI. Su sencillez hace que sea muy fácil de formalizar e implementar.

Sistemas de RI de búsqueda aproximada

Los estudios sobre el modelo booleano dieron lugar a que estos modelos se ampliaran para no desestimar un documento porque en él no aparecieran todos los términos de la consulta.

Modelo vectorial

También conocido como modelo de espacio vectorial, está basado en el modelo booleano, pero mejorado, de manera que se asigna a cada término de la consulta un peso que puede ser cualquier valor positivo.
Dentro de este modelo los documentos son representados utilizando un vector en el que se recogen las relaciones existentes entre el documento y sus características.
La consulta también se representa como un vector por lo que este modelo resulta perfecto para realizar la comparación entre documentos y consultas.

	t_1	t_2	t_3	...	t_j	...	t_m
d_1	w_{11}	w_{12}	w_{13}	...	w_{1j}	...	w_{1m}
d_2	w_{21}	w_{22}	w_{23}	...	w_{2i}	...	w_{2m}
..	..	..	..	..	..	..	..
d_i	w_{i1}	w_{i2}	$w_{i3}1$	...	w_{ij}	...	w_{im}
..	..	..	..	..	..	..	..
d_n	w_{n1}	w_{n2}	w_{n3}	...	w_{nj}	...	w_{nm}

Figura 1: Matriz de frecuencia de términos del modelo vectorial

Se elaboran vectores de términos a partir de los documentos seleccionando un conjunto de palabras que sea útil para discriminar unos textos de otros (se denominan términos o keywords). En los sistemas modernos todas las palabras del texto se consideran términos, excepto las stopwords o palabras vacías. Se puede enriquecer esto con procesos de lematización (stemming), etiquetado e identificación de frases.

A cada uno de los términos que aparecen en el vector hay que asignarle un peso en función de la frecuencia con la que aparece la palabra en el documento o en la colección de documentos entera.

Modelo Probabilístico

Una manera de relajar la dependencia entre los resultados recuperados y los términos explícitos de la consulta es usando el modelo probabilístico.
Este modelo se fundamenta en el cálculo de la probabilidad de que el documento sea relevante para la consulta realizada. Por tanto si cogemos un documento cualquiera entre un conjunto de documentos, existe una cierta probabilidad de que dicho documento sea relevante para la pregunta realizada.
Se tienen que analizar las características que hacen a un documento ser relevante.
Para calcular la relevancia se utilizan una serie de pesos dados a las características del documento.
Para saber la relevancia se usan índices de los términos que se conocen como descriptores con los pesos que se han establecido. Con esto se pretende recuperar los documentos en los que existen los mejores descriptores de los que el usa en la consulta.
Para el modelo probabilístico, los pesos de los términos son siempre binarios.

El modelo probabilístico se basa en el siguiente supuesto: dada una consulta *q* y un documento d_j en la colección, el modelo probabilístico trata de estimar la probabilidad de que el usuario encuentre a dicho documento relevante. El modelo asume que esta probabilidad de relevancia depende únicamente de la consulta hecha y del propio documento. De esta manera, se asume que existe un subconjunto de entre todos los documentos que el usuario quiere como respuesta a la consulta *q*. Ese conjunto *R* que forma la respuesta ideal maximiza la probabilidad de relevancia para el usuario. Todos los documentos que se encuentren en el conjunto *R* se dice que son relevantes para la consulta y los que no están en la consulta son no relevantes.
El problema es que no se dice de qué manera calcular la probabilidad de que un determinado documento sea relevante o no. Dada una consulta *q*, la relevancia de un documento d_j se calcula como indica la ecuación 2.

$$sim(d_j, q) = \frac{P(d_j\ relevante\ para\ q)}{P(d_j\ no\ relevante\ para\ q)}$$

Ecuación 2: Probabilidad de documento relevante en modelo probabilístico

La mayor ventaja de este modelo es que los documentos son ordenados de manera decreciente respecto a la probabilidad que tienen de ser relevantes.
Pero tiene una serie de desventajas como por ejemplo:

- La necesidad de suponer inicialmente la separación de documentos relevantes y no relevantes, es decir que se comienza adivinando y luego se refina esa apuesta iterativamente.

- El método no considera la frecuencia con la que un término aparece en un documento, sino que ve cada documento como un conjunto de términos (la información es binaria).

- Necesita presuponer que los términos son independientes.

Latent Semantic Indexing

Resumir el contenido de los documentos y las consultas a un conjunto de términos puede ocasionar problemas a la hora de recuperar información debido a que muchos documentos no relevantes pueden ser incluidos dentro del conjunto de respuesta por compartir términos con la consulta y a que documentos que sí son relevantes, pero que no tengan ninguno de los términos que aparecen en la consulta, no son recuperados.
La sinonimia y la polisemia son las dos principales causantes de estos problemas. Al hablar de sinonimia nos referimos al hecho de que existen diferentes maneras de llamar a una misma cosa. Los usuarios en diferentes contextos, con diferentes necesidades, conocimientos, hábitos lingüísticos describirán la misma información usando distintos términos. La sinonimia es la principal culpable de disminuir el valor de ”recall” de los sistemas de recuperación.
Por polisemia se entiende al hecho de que muchas palabras puedan tener más de un único significado. Una misma palabra usada en diferentes contextos o por distintas personas puede llegar a significar cosas completamente distintas. De esta manera, el hecho de que un determinado término aparezca en una consulta no significa necesariamente que un documento que contenga dicho término sea de interés. La polisemia hace que los sistemas de recuperación obtengan una baja ”precisión”.

El método Latent Semantic Indexing (LSI) utiliza la relación implícita que existe en términos y documentos, pero a nivel semántico, pretendiendo así mejorar la detección de aquellos documentos que sean relevantes en función de los términos que se hayan encontrado en la consulta. Se vale de un método matemático conocido como Singular Value Descomposition (SVD), cuyo cometido es el de factorizar matrices. En este caso se trata de una matriz de términos por documentos, que una vez factorizada representa la estructura semántica latente entre la colección de documentos y los términos contenidos. El motivo de usar SVD es el de reducir la dimensionalidad del espacio de términos, que terminan agrupándose como conceptos (ideas más generales que pueden englobar uno o más términos).
De esta manera se reducen los efectos de la sinonimia y la polisemia.

El método LSI, a pesar de ser una muy buena opción, tiene también sus desventajas:

- Es muy efectivo en colecciones pequeñas de documentos, pero no tanto en colecciones grandes.

- La transformación algebraica que se lleva a cabo hace que el método no sea capaz de devolver qué términos son responsables de la similitud de los documentos.

Modelo de redes neuronales

En un sistema de Recuperación de la Información, los vectores de documentos son comparados con los vectores consulta para calcular el grado de similitud entre ellos. Esto se hace capturando y pesando los términos índice que aparecen en los documentos y en las consultas y comparando los patrones de unos y otros. Como las redes neuronales son conocidas por ser buenas encontrando patrones, es natural considerar su uso como un modelo alternativo para la recuperación de información.

Una red neuronal empleada en recuperación de la información puede ser definida como ilustra la siguiente figura.

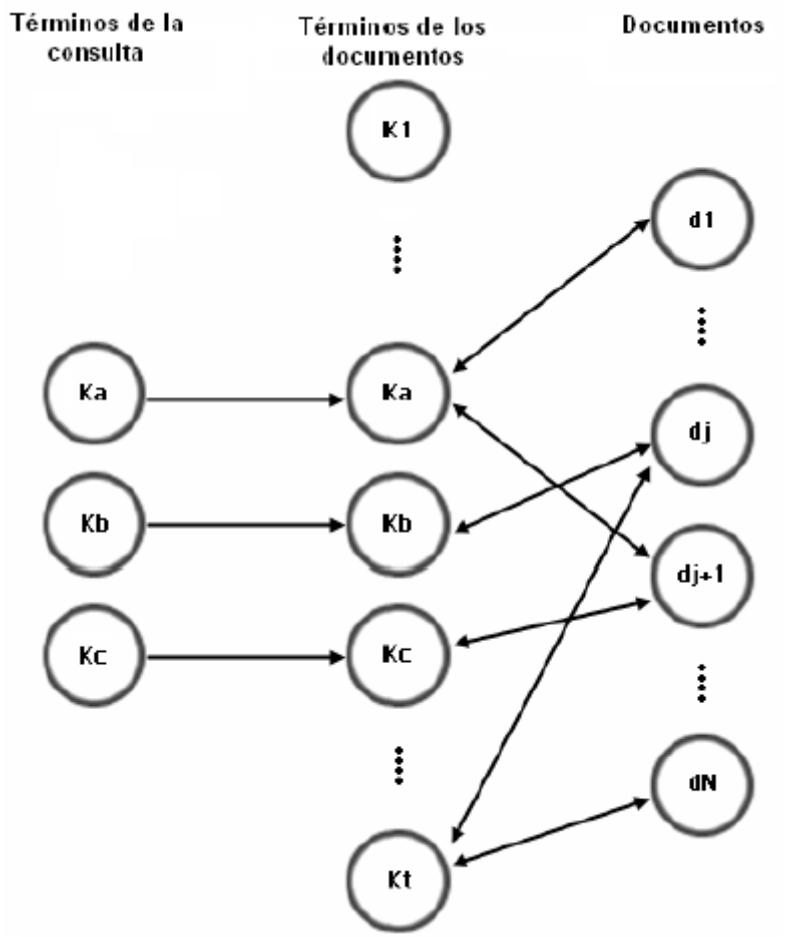


Figura 2: Modelo de red neuronal

Se observa que la red neuronal está compuesta por tres capas: una para los términos de la consulta, otra para los términos de los documentos y la tercera para los documentos mismos.

En este modelo los nodos de la primera capa (los términos de la consulta) son los que inician el proceso de inferencia enviando señales a los nodos de los términos de los documentos (segunda capa). Seguidamente, los nodos de la segunda capa propagan la señal (o no) hasta los nodos que representan a los documentos (tercera capa).
Así se completa la primera fase en la que una señal viaja desde los nodos de los términos de la consulta hasta los nodos de los documentos.

La red neuronal, sin embargo, no termina tras esta primera fase, sino que los nodos de los documentos generan nuevas señales y las propagan hacia atrás, hacia los nodos de los términos de los documentos (esa es la razón por la cual las aristas que conectan la segunda y a tercera capa son bidireccionales).
El proceso se repite recursivamente mientras la señal se hace cada vez más débil, hasta que llega un momento en el que el proceso de activación se termina parando.

2.2.3. El procesamiento del Lenguaje Natural en RI

Esta sección introduce las disciplinas involucradas en el procesamiento de texto así como con las técnicas y métodos que usan. Empezamos con técnicas generales de Procesamiento Natural de Lenguaje (NLP) y Text Mining.

En 1999 comenzó una discusión sobre las mejoras de la utilización de las técnicas de NLP en los sistemas de RI. En la actualidad la mayoría de investigadores están a favor de que la contribución de las técnicas avanzadas del NLP es en realidad pequeña y además la efectividad de los sistemas de RI está estrechamente relacionada con la formulación de las consultas. Si una consulta no está muy bien formulada, los errores que producen los sistemas de NLP a menudo pueden empeorar la eficacia de los sistemas de RI. Además, las consultas son difíciles de procesar si se tienen en cuenta todos los sinónimos, y los emparejamientos de palabras, lo que produce una explosión lingüística.

Pero ello no significa que los sistemas de NLP no sean útiles en RI, muy al contrario, se ha probado que las técnicas básicas de NLP, como la extracción de raíces, o técnicas más avanzadas como detección de multi-términos y nombres propios, detección de relaciones de sinonimia y expansión de consultas juegan un papel muy importante en los sistemas de RI.

Las líneas de investigación del NLP aplicado a los sistemas de RI son básicamente los siguientes:

- La interacción basada en el significado (búsqueda conceptual).

- Respuesta a preguntas concretas (no búsqueda de documentos).

- Creación automática de resúmenes como respuesta a las consultas.

- Integración de información.

- Creación de consultas altamente descriptivas, precisas y elaboradas.
- Multilingüismo.

Las implementaciones lingüísticas básicas provenientes de los sistemas de NLP para mejorar los sistemas de RI son:

Segmentación del texto en vocablos (Tokenizing). En algunos idiomas esta tarea es muy sencilla ya que existen separadores entre las palabras. Pero en idiomas como el japonés o el chino, para extraer las palabras del texto se requiere la ayuda de un diccionario y la utilización de patrones.

Extracción de raíces. Se suele utilizar para unificar los términos con variantes morfológicas. La falta de un dominio específico o del conocimiento del contexto da lugar a que estos sistemas provoquen fallos en la recuperación (varias palabras con significados distintos a veces tienen la misma raíz).

Utilización de listas de palabras de parada. El estudio de los sistemas de RI ha demostrado que la presencia de las palabras con poco valor semántico o demasiado frecuentes influyen poco en su efectividad. Por ello los indexadores de sistemas de RI, poseen colecciones de palabras de este tipo, de modo que cuando una palabra pertenece a esta lista no se indexa. Con ello se reduce el espacio de búsqueda y el tamaño del índice.

Algunos sistemas utilizan algunas implementaciones de técnicas de NLP un poco más complejas como son:

Identificación de frases. El objetivo es utilizar frases como unidades de indexación. La creación de frases como unidades, se basa en presuponer que algo ha pasado con anterioridad en las frases procedentes, de modo que cuando hay elementos cohesivos entre varias sentencias se forma una unidad.

Identificación de nombres de entidades. Estos pueden ayudar a identificar nombres propios, nombres de lugar y organizaciones. Para ello se aplican técnicas de análisis de patrones, a partir de la aplicación de reglas (manuales o creadas mediante un sistema de aprendizaje) o bien mediante modelos de Harkov (que requieren un conjunto de entrenamiento de documentos etiquetados).

Extracción de conceptos. Es una versión más general de la extracción de nombres de entidades. Se intentan identificar nombres de ciudades, países, provincias, títulos, fechas, monedas, porcentajes, nombres químicos, etc. La obtención de esta información es similar a la de nombres de entidades, el problema a resolver es determinar cuándo se deben utilizar los conceptos y cuándo un sistema de RI los requiere.

Procesamiento de Lenguaje Natural: Técnicas Generales

Las técnicas de NLP cubren todos los aspectos y etapas necesarias para convertir el lenguaje escrito o hablado en información que pueda ser usada por otros humanos o agentes automatizados.
A continuación se explican las operaciones comunes de procesamiento de texto usadas por los sistemas típicos de text mining.

Tokenización

Este primer paso en el análisis de texto es el proceso de separar el texto en unidades, los denominados tokens. Los tokens pueden variar su granularidad en función de las necesidades. De esta manera, la tokenización se puede dar en distintos niveles: el texto puede ser dividido en capítulos, secciones, párrafos, frases, palabras, sílabas o fonemas. Existen muchos algoritmos diferentes para cualquier nivel de tokenización aunque generalmente el texto suele fragmentarse en frases o palabras y en algunos sistemas en sílabas.

Existe otro tipo de tokenización especial, los n-gramas (n-grams), que son subsecuencias de n elementos (caracteres) de un texto dado. Los n-gramas se emplean a menudo en sistemas de reconocimiento de patrones para determinar la probabilidad de que una palabra dada aparezca en un texto o en el proceso de recopilación de información cuando es necesario encontrar documentos similares dado un documento y una base de datos de documentos de referencia.

Eliminación de stopwords (palabras de parada)

Las palabras que son más frecuentes en los textos de una colección no son buenos discriminantes y se denominan stopwords. Artículos, preposiciones y conjunciones, así como algunos verbos, adverbios y adjetivos son candidatos naturales para formar parte de la lista de stopwords. Son característicos de cada lenguaje por lo que se requiere detectar el idioma de cada documento tratado. La eliminación de stopwords permite reducir el tamaño de la estructura de indexación que se use.

Lematización

El propósito de la lematización o stemming es obtener un único término de indexación a partir de las diferentes variaciones morfológicas de una palabra (por ejemplo, representar ”analysis”, ”analyzer” o ”analyzing” mediante ”analy”). Frecuentemente, una palabra no aparece exactamente en un documento, pero sí alguna variante gramatical de la misma, como plurales, gerundios, sufijos de tiempo verbal, etc. Este problema puede resolverse con la sustitución de las palabras por su raíz (stem).
Un stem es la porción de una palabra que resulta de la eliminación de sus afijos (prefijos y sufijos). Los stems son interesantes ya que permiten reducir variantes de la misma raíz gramatical a un concepto común.

Consecuentemente, el stemming permite reducir el tamaño de la estructura de indexación ya que el número de términos índice se reduce. Además, permite ampliar la definición de la información que poseemos o la consulta que se pretende satisfacer con las variantes morfológicas de los términos usados, mejorando así el performance de recuperación.
En eliminación de afijos, la parte más importante es la eliminación de sufijos porque la mayoría de las variantes de una palabra se generan con su introducción. El algoritmo más popular de eliminación de sufijos es el algoritmo de Porter [14]. Este algoritmo usa una lista para la detección de sufijos. La técnica se basa en aplicar una serie de reglas a los sufijos de las palabras del texto.

Part of Speech

Consiste en el uso de etiquetas que representen conjuntos de categorías de palabras, basándose en el papel que las palabras pueden desempeñar en la frase en la que aparecen. El etiquetado Part of Speech (PoS) es la anotación de las palabras con su etiqueta correspondiente en función del contexto de la frase. Las etiquetas almacenan información del contenido semántico de la palabra.

Análisis sintáctico

Es el proceso de determinar la estructura sintáctica completa de una frase.
Los sistemas que llevan a cabo este tipo de análisis toman como entrada una secuencia de tokens extraídos del texto original. La salida suele ser un árbol sintáctico, cuyas hojas corresponden con las palabras del texto y cuyos nodos internos representan estructuras sintácticas, identificadas por etiquetas gramaticales, como: sustantivo, verbo, sujeto, predicado, etc. Actualmente no existe un sistema lo suficientemente eficiente que analice sintácticamente un texto sin ningún tipo de restricción. Los algoritmos estándar suelen consumir muchos recursos y no son lo suficientemente robustos.

2.3. Detección y Seguimiento de Temas (TDT)

El objetivo de la Detección y Seguimiento de Temas (TDT) es desarrollar tecnologías que buscan, organizan y estructuran en varios idiomas, noticias y materiales textuales. El programa de TDT comenzó en 1997 con un estudio piloto con un pequeño conjunto de investigadores que identificaron tecnologías para la organización de los textos de las noticias de forma automática.
La comunidad continuó reuniéndose para diseñar los componentes principales del proyecto, las tareas de investigación experimental, y los recursos necesarios para la investigación.
Los principales participantes fueron DARPA (Defense Advanced Research Projects Agency), la Universidad Camegie Mellon, Dragon Systems y la Universidad de Massachusetts.

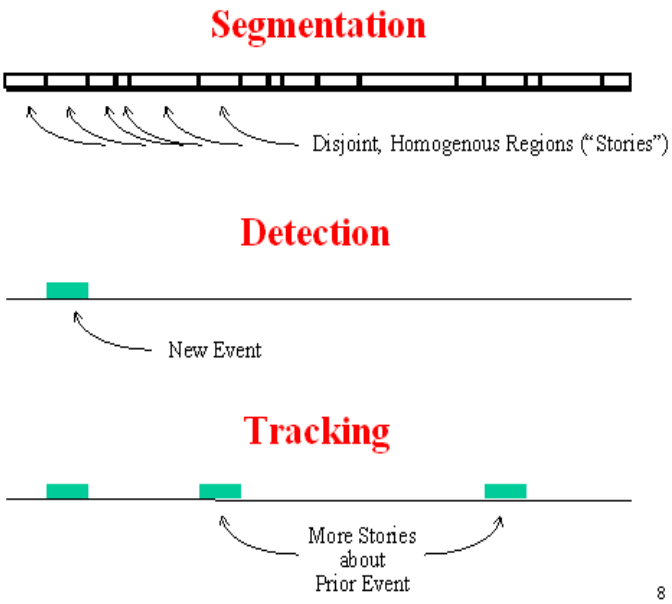


Figura 3: Tareas del estudio piloto TDT

Tal vez el concepto más importante en TDT es que los sistemas operativos del futuro van a procesar datos de forma continua a medida que se recogen.
La mayor parte de la investigación sobre la recuperación de textos y organización de la información se ha centrado en archivos de texto estáticos. En contraste, las tecnologías de TDT operan con los datos recogidos en tiempo real y de una variedad de fuentes y en multitud de idiomas.
El segundo concepto fundamental a la TDT es la noción de un evento, o lo que denominamos *tema*. Durante el estudio piloto y los siguientes años, la comunidad ha seleccionado y definido cinco tareas de investigación que simulan sistemas TDT. Las tareas fueron nombradas: Seguimiento del Tema (Topic Tracking), Detección de Enlace (Link Detection), Detección de temas (Topic Detection), Detección primera historia (First Story Detection) y Segmentacion de Historia (Story Segmentation).

El Instituto Nacional de Estándares y Tecnología (NIST) administra tres evaluaciones abiertas de tareas de TDT desde 1998. La Página web NIST TDT [15] contiene información acerca de las evaluaciones, así como numerosos artículos y ponencias presentadas en las convenciones TDT que NIST celebrada después de cada evaluación.

2.3.1. Introducción y Fases TDT

Los sistemas para el seguimiento y la detección de sucesos en las noticias (TDT – Topic Detection and Tracking) comenzaron en 1997 soportada por el DARPA dentro del programa TIDES (Translingual Information Detection, Extraction and Summarization).
La investigación para la Detección y Seguimiento de Temas (TDT, Topic Detection and Tracking) (Allan, 2002) se centra en aplicaciones cuyo objetivo es el de mantener un seguimiento de eventos de interés (categorías, historias, temas), en una colección creciente (dinámica) de historias multimedia: flujos de noticias, periódicos en línea (Mori et al., 2006), bitácoras colaborativas, etc. De esta forma, la aplicación es capaz de recibir nuevos elementos, detectar si pertenecen a una categoría existente o si es una nueva historia (nueva temática), y mantener un seguimiento de los mismos

Es dentro de estos sistemas dónde se encuadra el estudio del proyecto ya que trata de detectar sucesos en un conjunto de comentarios extraídos de Facebook (obtenidos de los perfiles de las marcas). Su objetivo es la detección de temas y agrupación y seguimiento de las noticias.

En TDT se han definido cinco tareas de investigación:

1. *Segmentación de noticias (Store Segmentation).*
La segmentación de historias consiste en la segmentación del flujo de datos procedente de una fuente en historias cohesionadas según el tema que tratan.
Consiste en extraer cada noticia de la colección.

2. *Seguimiento de sucesos (Topic Tracking).*
El objetivo de un sistema de "seguimiento" de eventos es el de seguir la pista de eventos que el usuario designa como de interés para historias futuras.
Consiste en clasificar las noticias en un conjunto de sucesos predeterminados. El problema se resuelve con técnicas de clasificación supervisada donde el sistema conoce a priori los sucesos de interés.

3. *Detección de temas (Topic Detection).*
Consiste en buscar los distintos sucesos que aparecen en las noticias y agrupar las noticias que hablan sobre el mismo tema. Este es un problema de clasificación no supervisada, donde se trata de organizar o agrupar automáticamente las noticias sobre el mismo suceso.
El objetivo de la detección es la de agrupar historias que tratan sobre el mismo tema. Mientras que los algoritmos de agrupamiento suelen trabajar en modo global, el agrupamiento en TDT se realiza incrementalmente, lo que implica que sólo podemos tomar decisiones de agrupamiento sobre historias futuras en un rango temporal limitado (p.e. 3, 4 o 5 días)

4. *Creación de temas nuevos (First Story Detection).*
El sistema debe decidir si un documento representa un nuevo tema.
La tarea es una abstracción de un sistema TDT que alerta al usuario cuando en el flujo de datos aparece un nuevo tema.

5. *Enlazar noticias (Store Link Detection).* La tarea de detección de relaciones evalúa un sistema de TDT que detecta si dos historias están relacionadas, es decir, si comparten el mismo tema.
Esta tarea no trata de dividir los documentos en conjuntos ortogonales, se permite que un documento hable de distintos temas, por lo que un documento puede pertenecer a varios grupos.

2.3.2. Segmentación de noticias (Store Segmentation)

La tarea de segmentación de historias evalúa tecnologías que detectan cambios en la historia. El sistema segmenta automáticamente el texto transcrito en historias de estilo TDT.
En TDT, una historia es una "cohesión de segmentos de noticias que incluye dos o más cláusulas declaradas independientes sobre un único evento". La noción de historia excluye explícitamente los anuncios de ser historias, y por lo tanto los sistemas no son evaluados entre comerciales consecutivos.
En TDT, la segmentación de la historia es una tecnología basada en la recuperación de la historia.
Esto implica que todo discurso transcrito de forma automática tendrá que ser segmentado en historias. Como se ha comentado previamente, TDT es multilingüe, la tarea de segmentación no es una excepción. En el caso de los textos en mandarín, en vez de requerir las traducciones al inglés, los sistemas de segmentación funcionan en ortografías nativas.

El rendimiento del sistema de segmentación usa el modelo de detección de costes, pero la derivación de detección fallida y la probabilidad de falsa alarma es muy diferente comparada a otras tareas TDT.
El rendimiento del sistema es juzgado por determinar qué tan bien calcula los límites de la historia junto con los límites de la referencia. La unión será juzgada con un intervalo de evaluación, nominalmente 15 segundos, que es el barrido de los datos de entrada. La técnica es una derivación del método propuesto por Beeferman, et al. [16] Se elige el intervalo de evaluación que sea suficientemente largo para incluir todos los límites calculados que puedan razonablemente estar asociados con un verdadero límite de referencia, pero lo suficientemente corto como para excluir asociaciones y múltiples límites de referencia (es decir, historias enteras).

La evaluación se realiza mediante el barrido del intervalo de evaluación a través del origen de datos de entrada y la corrección de la segmentación en cada posición del intervalo:

1. Si hay un límite de historia y un límite de referencia dentro del intervalo, entonces se juzga la segmentación como correcta.
2. Asimismo, si no hay límite de historia ni un límite de referencia dentro del intervalo, entonces se juzga la segmentación como correcta.
3. Sin embargo, una detección fallida se declara si no hay límite de historia dentro de un intervalo que contiene un límite de referencia.
4. Por otra parte, se declara una falsa alarma cuando un límite de historia existe dentro de un intervalo que no contiene un límite de referencia.

Las condiciones de evaluación para las tareas de segmentación son el lenguaje del material, la forma de difusión de las noticias y el período de aplazamiento de decisiones, medido en segundos.

2.3.3. Seguimiento de sucesos (Topic Tracking)

Los sistemas de seguimiento de tema TDT detectan historias que tratan sobre temas previamente conocidos. Un tema es "conocido" por su asociación con las historias que hablan de él.
Los sistemas son probados por su capacidad de encontrar historias on-topic dentro del resto del corpus.
Los desarrolladores deben tener en cuenta tres incidencias al realizar el diseño del sistema.
En primer lugar, los sistemas de seguimiento deben entrenar y probar cada tema de forma independiente. Los sistemas no pueden hacer uso de la definición de cualquier otro tema, lo que haría presumiblemente la tarea más fácil.
La independencia, la iteración de entrenamiento, la parte de evaluación del corpus utilizado para la formación de los modelos de sistemas, difieren de un tema a otro. Dado que el número de historias evaluadas es diferente de un tema a otro, la función de costes de detección tema es la ponderación del rendimiento del sistema. La independencia del tema tiene una importante ventaja. Dado que el protocolo de evaluación crea temas ortogonales, las historias que discuten varios temas se evalúan por separado para cada tema y por lo tanto se manejan más eficientemente.

La segunda incidencia de diseño del sistema es la normalización de la puntuación de decisión a través de los temas. Las puntuaciones de decisiones deben tener la misma media a través de los temas, por lo que para un ejemplo de puntuación de decisión de 15.0 para una historia y un tema indica la misma cantidad de evidencia que apoya una decisión sobre el tema para otra historia y otro tema. Matemáticamente, no sólo hacer que las medias de las puntuación de decisión tengan que coincidir, sino también sus varianzas.

Tener en cuenta que esta tarea sería mucho más sencilla si se permitiera a los sistemas hacer uso de otros temas de evaluación para la normalización de la puntuación, sin embargo, la formulación de la tarea como tal hace que los sistemas tengan problemas en la fiabilidad de la evidencia.

La tercera incidencia de diseño del sistema requiere que los sistemas de seguimiento sean multilingües.
Los sistemas deben realizar un seguimiento de temas en todos los idiomas del corpus independientemente del idioma de entrenamiento. Sin duda, esta es una tarea de enormes proporciones y requiere una infraestructura considerable. Para hacer esta tarea más accesible a los pequeños investigadores, la evaluación del corpus incluye traducciones al inglés para textos en mandarín.
Los sistemas de seguimiento se evalúan utilizando la función de coste normalizado y la curva DET, la curva DET (Detection Error Tradeoff) ha sido desarrollada para apreciar el funcionamiento del detector de forma más sencilla debido a que se dibuja en el gráfico la desviación normal en ambos ejes, dando un tratamiento uniforme para ambos tipos de error y usa una escala logarítmica para ambos ejes [17].

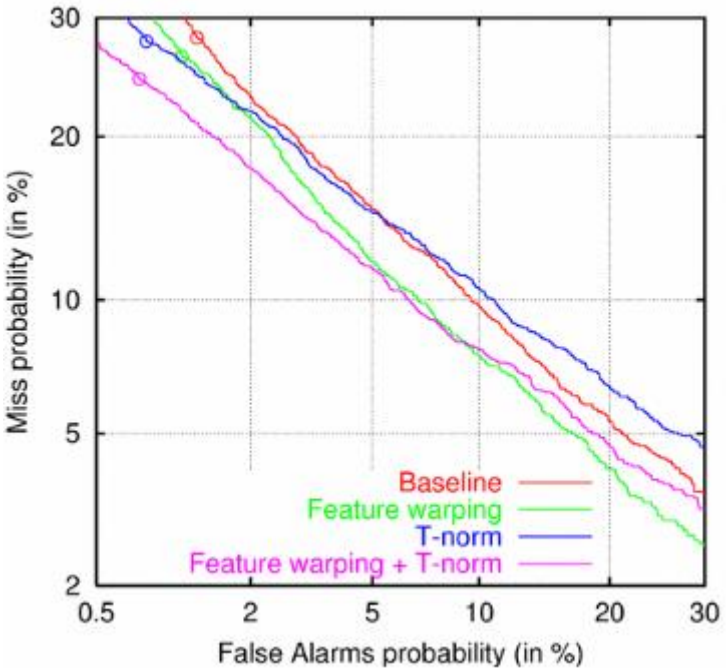


Figura 4: Ejemplo de curva DET

Hay muchas condiciones experimentales determinadas en el plan de evaluación, realizado por los desarrolladores y NIST, para la oportunidad de descomponer el rendimiento del sistema en los factores que se cree que afectará al rendimiento del sistema. La evaluación TDT 2000 tiene las siguientes condiciones: el número de historias de entrenamiento, el número de ejemplos negativos en historias de entrenamiento, el idioma de las historias de entrenamiento, la forma de la difusión de los datos de noticias, y referencia versus límites automáticos de la historia.

2.3.4. Detección de temas (Topic Detection)

Los sistemas TDT de detección de enlace detectan cuando un par de historias discuten sobre el mismo tema (es decir, las historias están relacionadas ("linked") por un tema común). Estos sistemas responden SI o NO a la siguiente pregunta "¿Estas dos historias habla sobre un tema en común?" y se anota la salida cuando las respuesta es SI. Las decisiones reales y las puntuaciones de decisión se utilizan para calcular la curva DET.

Esta tarea puede ser pensada como el núcleo a partir del cual los sistemas de seguimiento y detección del tema pueden ser construidos. La tarea de detección de enlace está relacionada con el seguimiento del tema, con una historia de entrenamiento, pero en lugar de realizar un seguimiento de las historias a través del tiempo, así las submuestras de la tarea de detección de enlace sean más eficientes. De lo contrario, un sistema tendría que evaluar N*(N-1)/2 pares de historias.
Hay ventajas en el paradigma de detección de enlace. Tal como se define, la tarea no requiere esfuerzo anotador para definir temas. El rendimiento puede ser evaluado utilizando los juicios humanos sobre pares de historias aleatorios, en cuanto a si discuten o no el mismo tema sin una declaración formal de un tema. Dado que el espacio no tiene que ser organizado en grupos ortogonales de historias, la manipulación de las noticias sobre múltiples temas no es un problema.

Otra ventaja es la capacidad para separar el rendimiento de pares de historias del mismo idioma. Puesto que el sistema juzga cada pareja de historias de forma independiente, la primera historia de la segunda.
La tarea es más flexible que la tarea de segmentación porque existen disposiciones para los sistemas para aprovechar los períodos de aplazamiento, (especificar cantidad de datos del futuro que pueden ser procesados antes de tomar decisiones sobre la historia actual).
Hay relativamente pocas condiciones de evaluación definidas por el plan de evaluación. Para la evaluación TDT 2000, esas condiciones eran la forma de difusión de los datos de noticias y el período de aplazamiento.

2.3.5. Creación de temas nuevos (First Story Detection)

La tarea de detección de primera historia (First Story Detection FSD) evalúa las tecnologías que detectan la primera historia para discutir el tema. Este caso especial de detección de tema se centra en el aspecto específico de detección de tema asociado con información de detección, es decir, saber cuándo comenzar un nuevo clúster. Los parámetros de la tarea son esencialmente los mismos que en la detección de tema. La verdadera diferencia está en las salidas del sistema.

La salida de los sistemas FSD en una decisión real, será SI o NO, en respuesta a la pregunta: "¿Esta historia discute un nuevo tema?" y una puntuación de decisión cuando la respuesta es SI. Aunque hay relativamente pocas primeras historias en un corpus, la evaluación del desempeño de esta tarea utiliza curvas DET.
Al igual que la detección de tema, la evaluación FSD supone que las primeras historias siempre discuten un solo tema. Las anotaciones de TDT de temas refutan esta suposición, por lo que la evaluación ignora primeras historias que son ambiguas, es decir, historias conocidas que discuten un tema visto anteriormente.

A diferencia de otras tareas, las historias etiquetadas como breves menciones de un tema se consideran como potenciales para no ser las primeras historias. Sin embargo, no se utilizan como candidatas a primera historia.

Para la evaluación TDT 2000, FSD fue estrictamente una tarea en idioma inglés. La restricción era una decisión pragmática hecha por la comunidad para racionalizar la evaluación. La tarea tiene las condiciones de evaluación adicionales que implica la forma de difusión de las noticias, la referencia versus segmentación automática de historia, y el período de aplazamiento de decisiones.

2.3.6. Enlazar noticias (Store Link Detection)

La tarea de detección tema evalúa tecnologías que detectan temas previamente desconocidos. Al igual que en la tarea de seguimiento, los temas son definidos por la asociación de historias que tratan sobre el tema. Sin embargo, la detección tema no tiene a priori conocimiento del tema. Por lo tanto, los sistemas han de entender que constituye un tema, y este entendimiento debe ser independiente de aspectos específicos del tema. La tarea es multilingüe y por lo tanto debe construir clusters por cada idioma.

Los sistemas detectan grupos de historias que tratan sobre el mismo tema. El concepto de agrupación se aplica fácilmente a las noticias, pero la evaluación del rendimiento es difícil porque las historias con frecuencia discuten varios temas.

Este fenómeno no sólo significa que los grupos por tema dependen de historias previamente procesadas, sino también que la descomposición del rendimiento en subconjuntos es engañoso.

El protocolo de evaluación debe abordar la cuestión de la independencia del tema. Las historias de varios temas se declaran no puntuables a pesar de que los sistemas realizan el agrupamiento de todas las historias. Por lo tanto, múltiples historias pueden influir en un sistema, pero no contribuyen al error de las medidas.

Hay una serie de condiciones de evaluación definido por el plan de evaluación.
Para la evaluación TDT 2000, esas condiciones eran el idioma de origen (inglés, mandarín y ambos inglés y mandarín), la forma de difusión de las noticias, referencia versus límites automáticos, y el período de aplazamiento de decisiones.

2.4. Métodos de evaluación de los resultados

Cuando se aplica algún tipo de análisis textual sobre una colección de documentos determinada o, más importante, cuando se desarrolla una herramienta nueva, es fundamental saber si los resultados obtenidos son fiables. Dado que es imposible conocer todos los posibles casos con los que se puede encontrar una herramienta de estas características (por ejemplo, todos los posibles artículos que pueden aparecer) y, por lo tanto, evaluar los resultados de manera anticipada no es viable, lo más razonables es medir la efectividad de una determinada herramienta comparándola con otra técnica candidata que haga el mismo tipo de análisis, utilizando en ambos casos el mismo dominio. El dominio consiste en un corpus anotado o etiquetado que está compuesto por elementos textuales. Además de eso, será necesario una medida o métrica para denotar la efectividad del sistema ejecutad sobre ese dominio.
Una buena manera para evaluar tanto sistemas de recuperación de la información como sistemas de extracción de la información es midiendo los valores de recall y precisión. Estas medidas se detallan en capítulos siguientes.

Capítulo 3

Herramientas y Algoritmos

Existe una gran cantidad de herramientas utilizadas en procesos de Data Mining y Topic Detection. A continuación se presentan algunas de las herramientas más relevantes e importantes.

3.1. Listado herramientas actuales

Gephi 

Gephi [18] es un software de visualización y análisis de grafos orientado a todo tipo de redes y sistemas complejos y grafos dinámicos y jerárquicos. Dispone de varios algoritmos implementados para el análisis de Redes Sociales, como PageRank, HITS, etc.
También presenta algoritmos para mejorar la visualización del grafo, utilizando diferentes layouts y es capaz de calcular diferentes métricas del grafo como su grado (power-law), betweenness, closeness, densidad, longitud de la trayectoria, diámetro, modularidad o coeficiente de clustering

Matlab 

MATLAB ® [19] es un lenguaje de alto nivel con un entorno interactivo para la computación numérica, visualización y programación. Posee varias herramientas para analizar datos, desarrollar algoritmos y crear modelos y aplicaciones.

Graphviz 

Graphviz [20] (Graph Visualization Software) es un software open source de visualización de grafos. Puede representar diagramas, grafos y redes. Se aplica para llevar a cabo análisis de redes, bioinformática, ingeniería del software, diseño de webs, machine learning y puede llevar a cabo visualizaciones para otros dominios técnicos.

Maui 

Maui [21] es un programa que identifica de manera automática el tema principal de documentos. Para ello hace uso de tags, keywords, keyphrases, descriptores, indexación de términos o títulos de artículos en Wikipedia entre otros. También puede ser utilizado como extractor de terminologías e indexador de topics (temas) semi-automático.

Wikipedia Miner

Wikipedia Miner [22] es un complejo de herramientas diseñadas para aprovechar la información latente en Wikipedia mediante análisis semántico. Provee acceso a la estructura de Wikipedia, así como técnicas de medición para ver cómo están relacionados entre sí diversos términos o eliminar la ambigüedad de algunos de estos.

JUNG

JUNG [23] (the Java Universal Network/Graph Framework) es una librería Java que provee de algunas herramientas para el modelado de grafos y redes, así como su análisis y visualización. Ha sido diseñado para soportar gran variedad de representaciones y herramientas analíticas para datasets complejos. También dispone de un framework para visualización con diferentes layouts.

Apache Mahout

La librería de machine learning Apache Mahout ™ [24] está diseñada para construir librerías escalables para machine learning. Provee de varias herramientas de machine learning para clustering, clasificación y filtrado colaborativo, implementadas sobre Apache Hadoop [25] utilizando el paradigma map/reduce.

Stanford TMT

Stanford Topic Modeling Toolbox [26] es un conjunto de herramientas de modelado de textos para científicos u otras personas que desean realizar análisis sobre conjunto de datos que tienen un importante componente textual. Desafortunadamente este software no se sigue desarrollando ni dando soporte.

Octave

Octave [27] es un lenguaje interpretado de alto nivel para la computación numérica. Proporciona extensas capacidades gráficas para la visualización y manipulación de datos. El lenguaje Octave es similar a Matlab, por lo que muchos programas son fácilmente portables.

Wandora

Wandora [28] es una aplicación con el objetivo de extracción de información, basado en Topic Maps y Java. Wandora tiene varias opciones de almacenamiento de datos, capacidad de importación y exportación y servidor incorporado.

RapidMiner 

RapidMiner [29] (anteriormente, YALE, Yet Another Learning Environment) es un programa informático para el análisis y minería de datos. Desarrollado en Java y multiplataforma.
Permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico. Puede usarse a través de un GUI o por línea de comandos.

Weka 

Weka [30] es una colección de algoritmos de machine learning para tareas de data mining. Contiene herramientas para el pre-procesado de datos, clasificación, regresión, clustering, reglas de asociación y visualización. Es software open source bajo la GNU General Public License.

R 

R [31] es un lenguaje y un entorno para computación estadística y gráfica. Es parte del proyecto GNU. Proporciona varias técnicas estadísticas (modelado, test estadísticos, análisis de series temporales, clasificación, clustering, etc) y gráficas, es altamente extensible.

Rattle Rattle

Rattle [32] ofrece una interfaz sencilla y lógica para la minería de datos.
Se trata de una aplicación de minería de datos basada en código abierto utilizando lenguaje R e interfaz gráfica Gnome.

Relfinder 

Relfinder [33] es una herramienta de visualización de grafos, donde se representan las relaciones existentes entre dos términos. Está basado en el framework de open source Adobe Flex y funciona con cualquier dataset de RDFs estandarizado para proporcionar acceso mediante consultas SPARQL.

PSPP GNU PSPP

PSPP [34] es una aplicación de software libre para el análisis de datos. Se presenta en modo gráfico y está escrita en el lenguaje de programación C.
Es una herramienta potente que puede utilizarse para exploración de análisis de datos, pruebas de hipótesis y preprocesamiento y visualización de datos.
Se puede utilizar con línea de comandos o interfaz gráfica de usuario.

3.2. Herramientas finales

El listado anterior explica brevemente las distintas herramientas más importantes y más usadas para el procesado de textos, pero solamente algunos están más capacitados para la extracción de temas y mejor enfocados a los resultados que buscamos.

Después de analizar y probar las diferentes herramientas a nuestra disposición se ha llegado a la conclusión, en un principio, que las herramientas elegidas para la realización de los experimentos y la comparación final fueran las siguientes herramientas o algoritmos:

- Maui
- Stanford NLP
- Wandora
- Rapidminer
- Weka

La selección de estas herramientas o algoritmos se debe por su sencillez en el manejo para un usuario novato, por tener una interfaz gráfica intuitiva y rápida de usar, por la similitud de los resultados de salida para poder realizar una comparación final entre herramientas. Por ser licencias GNU de software libre.

Después de realizar los primeros experimentos de test para comprobar el funcionamiento de cada herramienta y decidir las herramientas finales pasamos a realizar pruebas con mayor volumen de datos por lo que la herramienta de Weka no era capaz de soportar y lo hacía vulnerable a gran cantidad de texto, finalmente se ha decidido descartar esta herramienta como una de las herramientas finales.

Sin embargo Maui nos ha brindando la posibilidad de obtener dos resultados diferentes realizando el algoritmo sobre diferentes puntos de vista. El primero sobre el propio texto de entrada a lo que en adelante se denominara como resultado Maui KEY, y el segundo sobre artículos de Wikipedia, que en adelante en la memoria se denominara como resultado Maui WIKI.

Finalmente los resultados obtenidos que se evaluarán serán a partir de las siguientes herramientas o algoritmos.

- Maui KEY
- Maui WIKI
- Stanford NLP
- Wandora
- Rapidminer

Capítulo 4

Análisis de las herramientas finales

4.1. El Algoritmo KEA

Primero antes de hablar del algoritmo de Maui debemos explicar el algoritmo de Kea del cual se basa. El algoritmo de extracción de frase clave Kea es simple y eficaz, (Frank et al., 1999). Utiliza el algoritmo Naïve Bayes de aprendizaje para la formación y la extracción de frase clave. Una implementación está disponible desde el proyecto de Biblioteca Digital de Nueva Zelanda [35].

Kea se basa en el trabajo de Turney (2000), que fue el primero en tratar este problema como un problema de aprendizaje supervisado a partir de ejemplos. Otros habían utilizado anteriormente heurísticas para extraer frases clave de un documento (Krulwich y Burkey, 1996), o métodos tales como redes neuronales (Muñoz, 1996), o la heurística de información mutua (Steier y Belew, 1993), para descubrir una gran lista de frases de dos palabras. Ha habido también una gran cantidad de investigación relacionada a la generación o extracción de información de resumen de texto (por ejemplo Brandow et al., 1994; Johnson et al., 1993; Kupiec et al., 1995), pero esto, en general, son intentos de extraer frases completas en lugar de palabras clave.

En un ejemplo para un artículo en el que el autor ha proporcionado sus propias palabras clave para identificar el artículo, la salida de Kea, las palabras clave extraídas de forma automática, tendrá similitud con las palabras clave elegidas por el autor. Un inconveniente de Kea, además de la elección de varias buenas frases clave, también elige algunas palabras claves totalmente ajenas al tema principal. A pesar de estas anomalías, las listas extraídas automáticamente parecen proporcionar una descripción razonable de los documentos. En el caso en que no se dispusiera de palabras clave especificadas por el autor, las opciones de Kea serían un recurso valioso para alguien que se encuentra con un artículo la primera vez.

El objetivo con Kea es proporcionar información o datos útiles que antes no existían. Permitiendo extraer resúmenes razonables de documentos de texto, dando una valiosa herramienta para los diseñadores y los usuarios de las bibliotecas digitales.

La siguiente sección detalla el diseño del algoritmo en el que después está basado el código de Maui que es una de las herramientas usadas en este proyecto. A continuación, se explica los de predicción por el modelo generado por Kea y mostrar cómo se utiliza para evaluar una frase clave candidata y realizando varios experimentos diseñados para probar la eficacia de Kea.

El algoritmo de extracción de Kea tiene dos etapas:

1. Entrenamiento: crear un modelo para la identificación de palabras clave, usando documentos de formación donde se conocen las palabras clave del autor.

2. Extracción: elegir palabras clave de un nuevo documento, utilizando el modelo anterior.

El proceso se describe en la Figura 5. Ambas etapas eligen un conjunto de frases candidatas de sus documentos de entrada y, a continuación, calculan los valores de ciertos atributos (llamados recursos o medidas) para cada candidato.

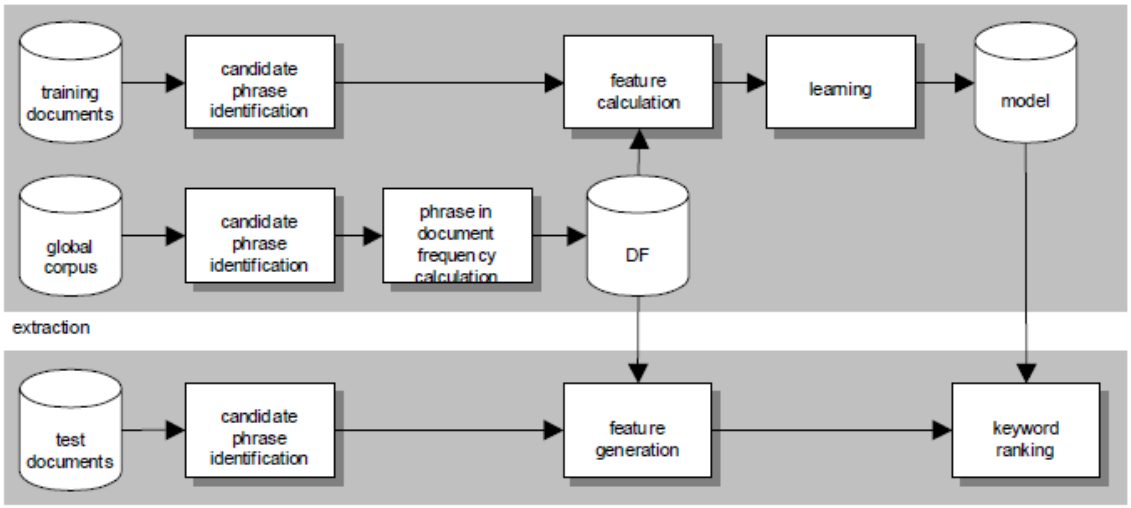


Figura 5: Esquema de etapas del algoritmo de KEA

4.1.1. Frases candidatas

Kea elige frases candidatas en tres pasos. En primer lugar, limpia el texto de entrada, a continuación, identifica candidatos, y, finalmente, clasificación.

Limpieza de entrada

Recibe los archivos de entrada en código ASCII y determina los límites iníciales de cada frase. El flujo de entradas se divide en tokens (secuencias de letras, dígitos y períodos internos), y después se realizan unas comprobaciones de limpieza:

- Signos de puntuación, paréntesis, y números se sustituyen por los límites de la frase.
- Se eliminan apóstrofes.
- Palabras con guiones se dividen en dos.
- Se eliminan los restantes caracteres no simbólicos, como son los tokens que no contienen letras.

El resultado es un conjunto de líneas y cada secuencia de tokens contiene al menos una letra.

Identificación de la frase

Kea entonces considera todas las subsecuencias en cada línea y determina cuáles de estas es más candidata. Las siguientes reglas son básicas.

1. Las frases candidatas se limitan a una determinada longitud máxima (generalmente tres palabras).

2. Las frases candidatas no pueden ser nombres.

3. Las frases candidatas no pueden comenzar o terminar con una palabra de lista de palabras de parada.

La lista de palabras de parada contiene 425 palabras en nueve clases sintácticas (conjunciones, artículos, partículas, preposiciones, pronombres, verbos anómalos, adjetivos y adverbios).

Clasificación

El último paso en la determinación de frases candidatas es usar el método iterativo de Lovins. Esto implica el uso de un clasificador Lovins clásico (1968) para descartar cualquier sufijo, y repetir el proceso en la subfrase que queda hasta que no haya más cambios. Por ejemplo, la frase *cut elimination* se vuelve *cut elim*.

Durante la clasificación se permite tratar diferentes variaciones en una frase como la misma. Por ejemplo, *proof net* y *proof nets* son esencialmente la misma, pero sin el clasificador tendrían que ser tratadas como diferentes frases. Por tanto por ejemplo las frases *cut-elimination* y *cut elimination*, y *proof nets* and *proof net*, se consideran equivalentes.

Finalmente, se compara las palabras clave de la salida de Kea con las palabras clave del autor. Consideramos una frase clave especificada por el autor se ha identificado con éxito si, es lo mismo que una frase clave generada por máquina.

Se conservan las palabras sin clasificar para cada frase, para su presentación al usuario en caso de que la frase pueda llegar a ser una frase clave.

4.1.2. TF × IDF

Principalmente el algoritmo calcula una característica durante el entrenamiento y la extracción por cada palabra candidata. Esta característica es *TF × IDF* que compara la frecuencia del uso de una frase en un documento particular con la frecuencia de la frase en uso general. El uso general está representado por el número de documentos que contienen la frase a lo largo del corpus. La frecuencia de una frase en el documento indica cuanto de común es (las frases más raras son más propensas a ser frases clave). El algoritmo de Kea y Maui construyen un archivo de documento de frecuencia para este propósito usando el corpus de los documentos. Se generan frases candidatas a partir de todos los documentos utilizando el método descrito anteriormente. El archivo almacena cada frase y su conteo de número de documentos en los cuales aparece.

Term Frequency (TF)

Se calcula contando el número de veces que la palabra aparece en el documento dividido entre el número total de palabras contenidas en él.

$$\mathbf{TF_{i,d}} = \textit{term count /number of words}$$

Ecuación 3: Cálculo de TF

Inverse Document Frequency (IDF)

Se calcula dividiendo el número total de documentos de la colección entre el número de documentos que contienen la palabra, *y se calcula el logaritmo de ese valor*:

$$\mathbf{IDF_i = \log (}\textit{number of documents in collection / number of documents containing word i)}$$

Ecuación 4: Cálculo de IDF

Con atención podemos comprobar una peculiaridad en el denominador de esta ecuación: si la palabra a buscar no está contenida en ningún documento, tenemos una división entre cero. Para evitarla, se acostumbra sumar 1 al denominador:

$$\mathbf{IDF_i = \log (}\textit{number of documents in collection / 1 + number of documents containing word i)}$$

Ecuación 5: Cálculo de IDF mejorado

IDF sirve para penalizar a palabras que estén en muchos documentos, y de la misma manera, para valorar más a palabras que estén en pocos. ¿Por qué? Porque se parte de la premisa que una palabra que no sea tan común proporciona más información (es más valiosa) que una que es más común (y por ende, que tiende a aparecer en un mayor número de documentos).

El resultado final TF×IDF para la frase *P* en documento *D* es:

$$\mathrm{TF{\times}IDF} = \frac{\mathrm{freq}(P, D)}{\mathrm{size}(D)} \times -\log_2 \frac{\mathrm{df}(P)}{N}$$

Ecuación 6: Cálculo final de TFxIDF

Donde:

1. freq(*P*,*D*) es el número de veces que P aparece en D
2. size(*D*) es el número de palabras en D
3. df(*P*) es el número de documentos que contienen P
4. *N* es el número de documentos.

El segundo término en la ecuación es el logaritmo de la probabilidad de que esta frase aparece en cualquier documento (negada porque la probabilidad es menor que uno). Si el término no se encuentra en el documento, *df (P)* y *N* se incrementan en uno antes de que el término es evaluado, para simular su aparición en el corpus (para evitar la división por cero antes mencionada).

4.2. El Algoritmo Maui Indexer

Los detalles del algoritmo de Maui son muy complejos. Sólo se proporciona en este documento una breve descripción para proporcionar una mejor base para la visualización de los resultados de la evaluación.

Maui-indexer fue desarrollado por Olena Medelyan como parte de su tesis doctoral (Witten et al, 1999). Como ya se ha mencionado Maui se basa en el algoritmo de extracción de frase clave Kea y aparte de otras características también permite la asignación de temas a los documentos sobre la base de términos de Wikipedia.

El algoritmo de Maui mejora a Kea con un marco de aprendizaje máquina con conocimiento semántico recuperado de Wikipedia, nuevas características, y un nuevo modelo de clasificación. Se evalúa Maui utilizando conjuntos de etiquetas asignadas a los mismos documentos y mostrar su consistencia.

Tener en cuenta que los conjuntos de datos debe ser más pequeños para evaluar la consistencia de los indexadores humanos, porque tales conjuntos necesitan ser creados específicamente para el experimento.

Basado en el sistema Kea funciona en dos etapas: selección de candidato y filtrado o clasificación. Se aplica al etiquetado automático. En la etapa de selección de candidatos, Maui determina primero secuencias textuales definidos por límites ortográficos y divide estas secuencias en tokens. Creando n-gramas hasta una longitud máxima de tres palabras, que no pueden comenzar o terminar con una palabra de parada. Para reducir el número de candidatos, aquellos que aparecen solo una vez se descartan. Esto acelera la formación del proceso de extracción sin afectar los resultados. En la etapa de filtrado varias características se calculan para cada candidato, que son entrada a un modelo de aprendizaje máquina para obtener la probabilidad del candidato. La arquitectura de Maui se asemeja a la de muchos otros sistemas de extracción de frases clave supervisados.

Maui es de código abierto y está disponible su descarga.

4.2.1. Características para clasificación

A continuación se describen las tres principales características generalmente utilizadas en la clasificación del modelo para determinar si es una frase con probabilidades de ser una etiqueta. Las tres características también son utilizadas en Kea (Frank et al., 1999).

1. TF×IDF

Combina la frecuencia de una frase en un documento en particular con su frecuencia inversa de ocurrencia de uso general (Salton y McGill, 1983). Esta puntuación es alta para frases raras que aparecen con frecuencia en un documento y, por lo tanto es más probable que sea significativo.

2. Posición de la primera aparición.

Es calculado como la distancia relativa de la primera ocurrencia de la etiqueta del candidato desde el inicio del documento. Los candidatos con muy alto o muy bajo valor es probable que sean etiquetas, porque aparecen en partes del documento como la apertura, como título, resumen, tabla de contenidos, introducción o en las secciones finales del documento, tales como conclusión y lista de referencias.

3. Frases clave

Se cuantifica la frecuencia de una frase clave aparece como una etiqueta en la formación correspondiente. Enfoques de marcado automático utilizan la misma información: Mishne (2006) y Sood et al. (2006) sugieren automáticamente etiquetas previamente a documentos similares. Sin embargo en Maui (como en Kea) esta característica es solo un componente del modelo general. Así, si un candidato nunca aparece como una frase clave en el corpus de entrenamiento, todavía puede ser extraída si sus otros valores de características son los suficientemente significativos.

Están son las características que usa Maui para obtener frases clave cuando no se usa un dataset predefinido y solamente usando los propios documentos a tratar, lo que denominamos anteriormente como resultados Maui KEY.

Existe otra característica que usaremos también en los experimentos que es buscar frase clave en base de Wikipedia. Es la probabilidad de una frase de ser un eslabón en el corpus de Wikipedia. Se divide el número de páginas de Wikipedia en la que la frase aparece como ancla de un enlace por el número total de páginas de Wikipedia que lo contienen. Multiplicando este número por la frecuencia de la frase.
En este proyecto Maui también fue configurado para realizar indexación con Wikipedia. Esto significa que Maui realiza la extracción utilizando como frase clave Wikipedia para mejorar algunos pasos del algoritmo. Estos resultados son denominados en la memoria Maui WIKI.

4.2.2. Fases del algoritmo

El primer paso de Maui se divide en las siguientes fases:

• **Fase A** recibe el texto del documento como entrada. Su salida es un conjunto de segmentos de texto (oraciones completas o sus partes), siendo cada uno una secuencia de tokens de palabras que contienen al menos una letra.

• **Fase B** extrae todas las subsecuencias de muestras de longitud n (n-grama). Para cada n-grama Maui determina entonces si se trata de un candidato adecuado. Si Maui está configurado para trabajar con indexación con Wikipedia, como es el caso de una parte de los resultados, los candidatos se identifican con valor frase clave Wikipedia sobre un determinado umbral, es decir, aquellos que son propensos a aparecer como anclas en Wikipedia.

• En la **Fase C** los n-gramas están fusionados a un conjunto de temas de los candidatos. Aquí Maui utiliza *Wikipedia Miner* para recuperar artículos de Wikipedia y eliminar la ambigüedad a aquellos artículos que son más similares al contexto inequívoco.

• **Fase D** normaliza las posiciones de la ocurrencia de los candidatos por parte de la longitud del documento y las frecuencias de ocurrencia por el número de candidatos.

Es importante señalar que en el caso de indexación con Wikipedia los candidatos son los artículos de Wikipedia, o más preciso sus títulos. Esto no supone una gran restricción en los temas posibles. Con un tamaño de 1,1 millones de artículos de Wikipedia cubre un enorme conjunto los temas posibles. Además la Wikipedia tiene un gran número de páginas que solamente redireccionan, es decir, un punto a otros artículos que permiten cubrir una gran cantidad de sinónimo.

En la Fase C en la etapa de generación de candidato también se permite a Maui tratar con los candidatos ambiguos. Si un candidato tiene múltiples significados sólo el significado más probable, teniendo en cuenta su contexto en el documento, se utilizará como tema. Debido a la desambiguación el tema no puede estar contenido en el documento en la misma forma exacta, pero al menos uno de sus sinónimos o significados tiene que aparecer en el texto del documento para que Maui lo considere como candidato.

En los resultados experimentales de Maui WIKI se usa un dataset de Wikipedia predefinido con temática de naturales y ciencias sociales.

Después de la generación del candidato, Maui calcula características para cada uno de los candidatos. Las siguientes funciones se calculan durante la etapa de generación de candidato:

- *Term Frequency (TF)* – Descrito anteriormente.

- *Primera ocurrencia* – Descrito anteriormente.

- *Última ocurrencia* - La posición de la última ocurrencia para cada candidato en relación con el número de palabras en el documento.

En la etapa de formación de Maui también crea tablas con los siguientes valores:

- n_t es el número de documentos que contienen cada candidato.

- N es el número total de documentos.

- m_t es la frecuencia de un tema que aparece en los conjuntos de temas asignados manualmente.

Estas características y valores se utilizan en el cálculo de las siguientes características:

- *Inverse Document Frequency (IDF)* - Descrito anteriormente.

- *TFxIDF* - Descrito anteriormente.

- *Propagación* = última ocurrencia - primera aparición.

- *Dominio de frases clave* es 0 si el tema candidato nunca aparece en un conjunto tema asignado manualmente basado en m_t.

- *Wikipedia Frase Clave* involucra al candidato contra las anclas que aparecen en el corpus Wikipedia. Los valores se calculan previamente durante la generación del candidato.

Y en el caso de Maui configurado para Wikipedia:

- *Frecuencia Inversa Wikipedia* se calcula mediante la recuperación del artículo más probable Wikipedia para el candidato actual (a menos que el candidato sea un artículo de Wikipedia en sí) y contando el número de sus enlaces entrantes.

- *Total frases clave Wikipedia* es la suma de los valores *Wikipedia Frase Clave* sobre todos los n-gramas que fueron asignados al artículo de Wikipedia correspondiente al candidato dado.

4.3. Stanford Natural Language Processing Group

Stanford CoreNLP [26] proporciona un conjunto de herramientas de análisis de lenguaje natural que puede tomar como entrada texto sin procesar y dar formas de base a las palabras, partes de la oración, si son nombres de empresas, personas, etc., se normalizan fechas, horas y cantidades numéricas, y marcar la estructura de las oraciones en términos de frases y dependencias de palabras.
Su objetivo es hacer que sea muy fácil de aplicar un montón de herramientas de análisis lingüístico en un fragmento de texto. A partir de texto plano, puede ejecutar todas las herramientas en él con sólo dos líneas de código. Está diseñado para ser altamente flexible y extensible.

La distribución básica proporciona archivos de modelo para el análisis en inglés, pero el motor es compatible con los modelos para otros idiomas.
Stanford CoreNLP está escrito en Java y licenciado bajo la GNU General Public License (v3 o superiores, en general código de Stanford NLP es GPL v2+, pero CoreNLP utiliza varias bibliotecas con licencia de Apache, por lo que el conjunto de aplicaciones es v3+). Esta licencia GPL, permite muchos usos libres, no puede realizarse en uso de software propietario que se pueda distribuir a otros. Requiere Java 1.8+.

En este estudio usaremos la herramienta Stanford Topic Modeling Toolbox (TMT), un conjunto de herramientas de modelado de temas para los científicos sociales y otras personas que desean realizar análisis sobre conjuntos de datos que tienen un importante componente textual. Desafortunadamente, este software ya no sigue en desarrollo ni tiene soporte. He comprobado personalmente que con Java 8 no funciona.

Stanford Topic Modeling Toolbox es una colección gratuita de herramientas de modelado de tema, y una parte del conjunto de herramientas de Stanford Natural Language Processing Group. Esta herramienta puede aceptar y generar valores separados por tabuladores y separados por comas, y está diseñado para trabajar en combinación con programas de hojas de cálculo como Excel.

La caja de herramientas cuenta con la capacidad de:

- Importar y manipular texto de celdas en Excel y otras hojas de cálculo.
- Modelos de entrenamiento (LDA, etiquetado LDA, y nuevo PLDA) para crear resúmenes de texto.
- Seleccionar y configurar parámetros (como el número de temas) a través de un método impulsado por los datos.
- Generar salidas compatibles con Excel para el seguimiento del uso de la palabra a través de los temas, el tiempo y otras agrupaciones de datos.

El Stanford Topic Modeling Toolbox (TMT) fue escrito en el grupo de Stanford NLP por Daniel Ramage y Evan Rosen, lanzado por primera vez en septiembre de 2009. Actualmente existe la versión 0.4.0. TMT fue escrito durante 2009-10 en lo que hoy es una versión muy antigua de Scala, el uso de una biblioteca de álgebra lineal que también ya no se desarrolla. Algunas personas todavía lo utilizan y les resulta una pieza de software amigable para LDA y etiquetado LDA.

A continuación se explica el modelo de LDA en el que se basa el algoritmo interno usado por la herramienta de Stanford.

4.3.1. Algoritmo Latent Dirichlet Allocation (LDA)

En estadística, *Latent Dirichlet Allocation (LDA)* es un modelo generativo que permite que conjuntos de observaciones puedan ser explicados por grupos no observados que explican por qué algunas partes de los datos son similares. Por ejemplo, si las observaciones son palabras en documentos, presupone que cada documento es una mezcla de un pequeño número de categorías y la aparición de cada palabra en un documento se debe a una de las categorías a las que el documento pertenece. LDA es un ejemplo de modelo de categorías y fue presentado como un modelo en grafo para descubrir categorías por David Blei, Andrew Ng y Michael Jordan en 2002. [36]

Asignación de Dirichlet Latente (LDA) es un modelo probabilístico generativo de un corpus. La idea básica es que los documentos se representan como mezclas aleatorias sobre temas latentes, donde cada tema es caracterizado por una distribución de más palabras.

1. Elegir $N \sim$ Poisson (ξ).
2. Elegir $\theta \sim$ Dir (α).
3. Para cada N palabra w_n:
    a. Elegir un tema $z_n \sim$ Multinomial (θ).
    b. Elegir una palabra w_x de $p(w_n \mid z_n, \beta)$, una probabilidad multinominal condicionada del tema z_n.

En primer lugar, la dimensionalidad *k* de la distribución de Dirichlet (y por tanto la dimensionalidad del tema variable *z*) se supone conocida y fija. En segundo lugar, las probabilidades de palabra son parametrizadas por una matriz β de dimensiones $k \times V$ donde $\beta_{ij} - p(w^j - 1 \mid z^i - 1)$, que por ahora tratamos como una cantidad fija que ha de ser estimada.
Por último, la hipótesis de Poisson no es crítica para cualquier cosa que sigue y las distribuciones de longitud del documento más realistas pueden ser utilizadas según sea necesario. Además, tener en cuenta que *N* es independiente de todas las otras variables de generación de datos (θ y *z*). [37]

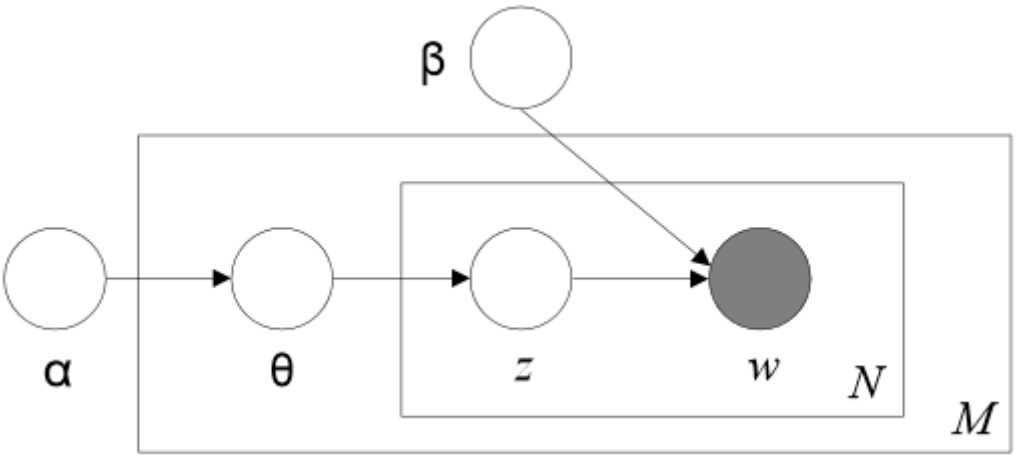


Figura 6 Representación gráfica del modelo de LDA. Las cajas son "placas" que representan réplicas. La placa exterior representa los documentos, mientras que la placa interior representa la elección repetida de temas y palabras dentro de un documento.

El modelo de LDA se representa como un modelo probabilístico gráfico en la Figura 6. Como la figura deja claro, hay tres niveles a la representación LDA. Los parámetros α y β son parámetros de nivel en corpus, suponiendo ser muestreados una vez en el proceso de generación de un corpus. Las variables θ_d son las variables de nivel de documento, muestreado una vez por documento. Finalmente, las variables z_{dn} y w_{dn} son variables a nivel de palabra y se toman muestras una vez por cada palabra en cada documento.

Estructuras similares a la mostrada en la Figura 6 se estudian a menudo en la modelización estadística bayesiana, donde se denominan como *modelos jerárquicos* (Gelman et al., 1995), o más precisamente como *modelos jerárquicos independientes condicionalmente* (Kass y Steffey, 1989). Tales modelos son también a menudo denominados *modelos Bayes empíricos paramétricos*, un término que se refiere no sólo a la estructura de un modelo en particular, sino también a los métodos utilizados para la estimación de parámetros en el modelo (Morris, 1983). Se adopta Bayes para estimar los parámetros tal como α y β en implementaciones sencillas de LDA.

En el esquema de la figura 7, se puede ver el algoritmo de uso, donde *k* representa el número de tópics, el codebook es el conjunto de todas las palabras posibles, *Pwd* contiene el número de veces que se ha repetido cada una de las palabras por cada uno de los documentos, *Pwz* es *P(w|z)* y muestra la probabilidad que tiene cada una de las palabras de formar parte de un topic y, *Pzd* que es *P(z|d)* e indica la probabilidad que tiene un documento de formar parte de un topic.

Figura 7: Esquema de uso del LDA

4.3.2. Stanford Topic Modeling Toolbox GUI

La herramienta GUI descargable en la web de Stanford permite abrir scripts para el estudio y modelado de un conjunto de textos. Los scripts utilizados en el proyecto también están disponibles en la misma web.

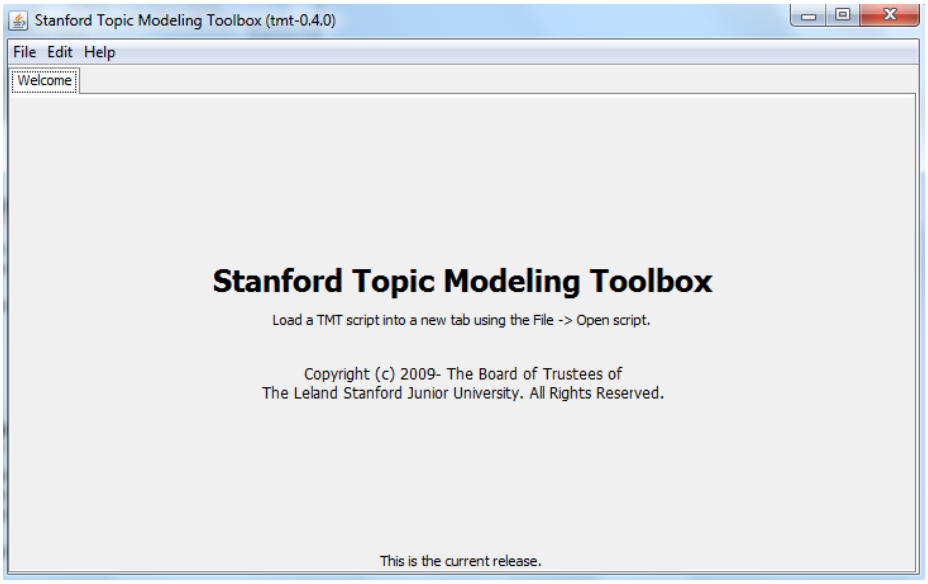


Figura 8: Ventana 1 Stanford Topic Modeling Toolbox GUI

Los scripts se han configurado de manera que se obtenga el mismo número de resultados que el resto de algoritmos para realizar una mejor comparación de resultados en precisión y recall además de tiempo de ejecución.
Se cargan los script y deben ser ejecutados en orden ya que en cada ejecución es necesaria la salida de la ejecución anterior, configurando los directorios de salida de datos para que el siguiente paso pueda utilizarlos. Hasta llegar al paso final y obtener los resultados reales a evaluar.

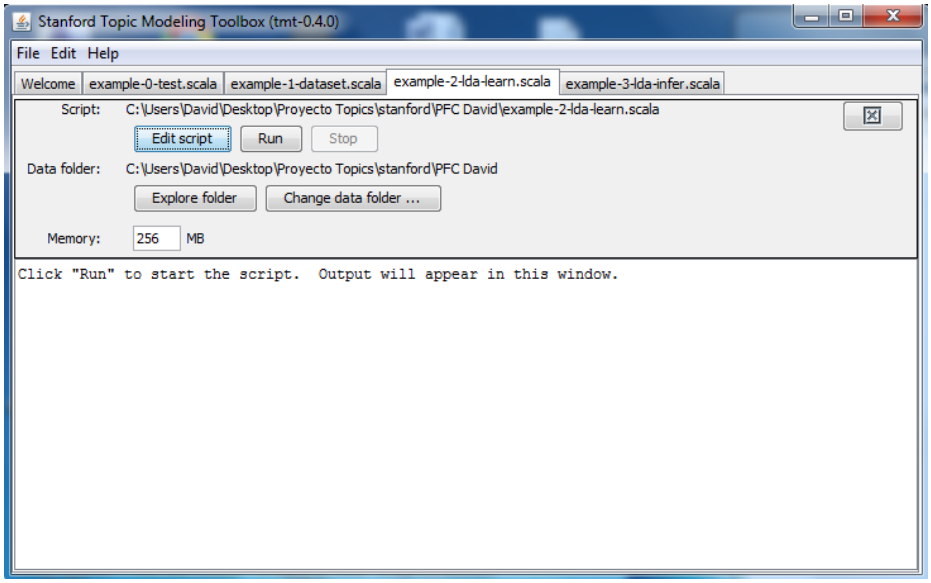


Figura 9: Ventana 2 Stanford Topic Modeling Toolbox GUI

Los resultados de Stanford Topic Modeling Toolbox vienen dados por orden de relevancia, siendo el primer keyword o topic el más relevante.

Esta herramienta también permite realizar estudios de texto diferenciando por fechas aunque en este proyecto no se ha realizado así y se ha considerado todos los comentarios comprendidos en el mismo año, pero la herramienta nos permite obtener resultados por fecha y realizar comparaciones y que topics destacan por períodos como por ejemplo en época de Navidad o en verano, anotación que haremos en el apartado de líneas futuras.

4.4. Wandora

Wandora es una aplicación de propósito general para la extracción, gestión y publicación de información basada en *Topic Maps* y *Java Swing* [28]. Es una aplicación de escritorio que dispone de una interfaz gráfica de usuario, permite gestionar los mapas en distintas capas, incorpora varias opciones de almacenamiento de datos (en memoria, en una base de datos relacional o mediante un servicio web, aún experimental), importa y exporta datos en diversos formatos, permite la importación de bases de datos SQL y de tópicos y asociaciones desde el portapapeles, además de facilitar estadísticas, permitir búsquedas, mostrar distintas visualizaciones gráficas y publicarlos en formato web. Sumado a ello, la mayor potencia de Wandora es la gran colección de opciones de extracción de datos que incorpora, que se ha visto ampliada recientemente con la extracción de datos desde registros MARC. En la Figura 10 se muestra el catálogo de extractores de que dispone.

Figura 10: Diferentes extractores implementados en Wandora

Dispone además de un pluging para Firefox que permite la utilización de algunos de estos extractores mientras se navega. La publicación en web se puede realizar de diversas formas entre las que se encuentran la creación de páginas web estáticas, directamente a partir de su servidor http embebido o mediante sus módulos para Drupal o Joomla.

Posee una documentación extensa que permite familiarizase con su funcionamiento y tiene también ciertas limitaciones pues utiliza una versión ligeramente reducida del modelo. Como usuario se ha utilizado el extractor más sencillo y con resultados eficaces, AlchemiAPI, aunque con la desventaja que no puede trabajar con textos muy grandes o extensos, en esto se parece al código Maui.

Wandora es una aplicación *open source* desarrollada desde el 2006. Su licencia es GNU GPL. Por ser *open source* solamente es posible usar el recurso gratuitamente. La aplicación está escrita en lenguaje de programación Java y es necesario Java Runtime Environment (JRE) en el escritorio Java versión 7, personalmente probamos la herramienta con la actual versión Java 8 pero no abría la interfaz, posiblemente versiones más modernas de Wandora puedan ejecutarse en Java 8.

La herramienta está programada a través de estructuras frásticas; es decir que obedece a las reglas básicas sobre las cuales construimos frases. Para ello entonces desarrollamos la unión de topics mediante asociaciones, en donde los topics serán los elementos informativos fundamentales sobre los que se habla. La herramienta no sólo permite manejar los topics de la información que deseemos estudiar o sistematizar, sino que nos permite manejar las representaciones que se han hecho de dichos topics.

Así, lo que proporciona Wandora es la posibilidad de construir un modelo de topics. Asimismo, es posible extraer información de diferentes documentos y relacionarla con otros que tengan el mismo rango de información o estén en el mismo campo semántico.
Es decir, los *topic maps* se constituyen en una herramienta que permite modelar el mundo de la información estableciendo relaciones reales con el mundo externo al modelo, esto es, el nuestro, que es un mundo basado en la información, en su procesamiento y en su manejo.

A continuación se explica brevemente el extractor utilizado en el proyecto a partir de la herramienta de Wandora.

4.4.1. Extractor AlchemyAPI

En particular para nuestro proyecto se ha usado un clasificador de keywords para obtener las palabras clave de cada subgrupo de comentarios del conjunto total de experimentos. El clasificador usa AlchemyAPI y es necesaria una conexión a internet.

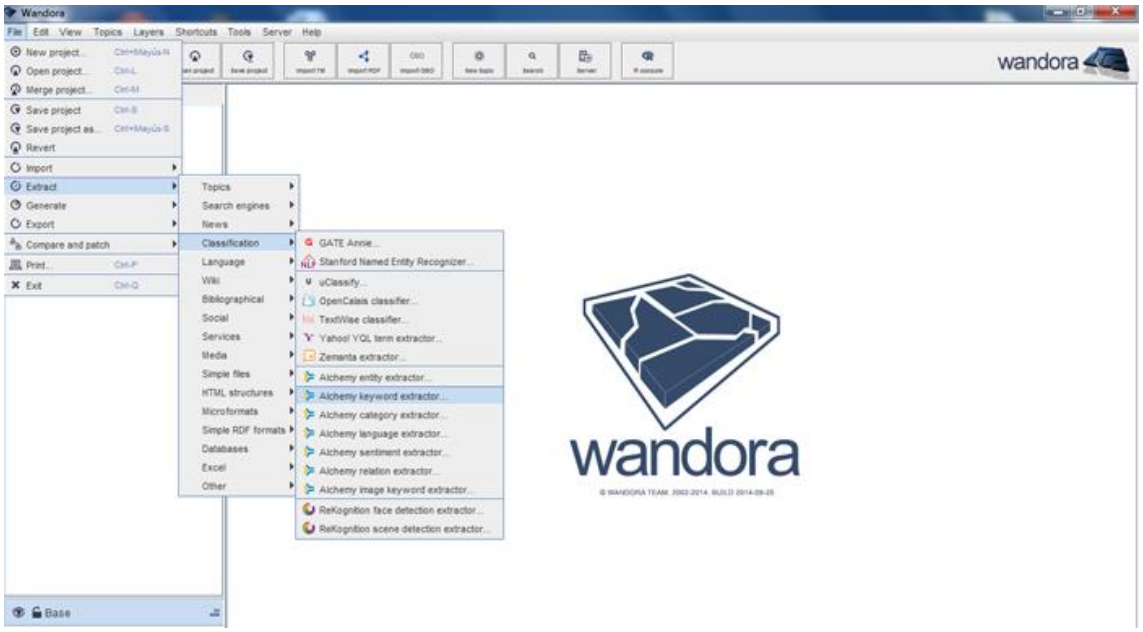


Figura 11: Opción AlchemyAPI en Wandora

Las empresas se enfrentan a una avalancha de datos de todo tipo: comentarios de los clientes, información sobre la competencia, los correos electrónicos de los clientes, los tweets, notas de prensa, registros legales y documentos producto/ingeniería.

¿El problema? Hay toneladas de este tipo de información, principalmente en la red (más del 85% de los datos del mundo no están estructurado), y los volúmenes de información que sigue creciendo. Para muchas empresas, es imposible emplear recursos humanos para leer todo y obtener los datos importantes.

Empleando lingüística compleja, estadística y algoritmos de redes neuronales, AlchemyAPI es capaz de leer y entender el texto, a tasas enormes de velocidad. Si se dispone de 1.000 documentos u 10 mil millones, AlchemyAPI puede digerir rápidamente toda esta información.

En el caso de API de extracción de palabras clave trabaja en URLs, documentos HTML y texto plano. Al igual que todas las otras características, AlchemyAPI detecta automáticamente el idioma del contenido y luego realiza el análisis correspondiente.

El algoritmo principal de extracción de palabras clave de AlchemyAPI emplea algoritmos estadísticos sofisticados y tecnología de procesamiento de lenguaje natural para analizar su contenido e identificar las palabras clave relevantes. Extracción de palabras clave se apoya en más de media docena de idiomas diferentes, lo que permite incluso el contenido de un idioma extranjero a ser clasificado y etiquetado [38].

Para usar AlchemyAPI es necesario ingresar unas keys que nos devuelve la web oficial al registrarnos y facilitando la información por la que queremos dicha clave de acceso, en nuestro caso para experimentos académicos. Los resultados de AlchemyAPI vienen dados por orden de relevancia, siendo el primer keyword el más relevante, en este caso es igual a los resultados obtenidos por Stanford que también devuelve por orden de relevancia.

4.5. RapidMiner

RapidMiner es una plataforma de software que proporciona un entorno integrado para aprendizaje automático, minería de datos, minería de texto, análisis predictivo y análisis de negocios. Se utiliza para aplicaciones de negocios e industriales, así como para la investigación, la educación, la formación, la creación rápida de prototipos y desarrollo de aplicaciones y es compatible con todos los pasos del proceso de minería de datos, incluyendo los resultados de la visualización, validación y optimización [39].

RapidMiner es de código abierto distribuido con licencia AGPL y versión gratuita, aunque dispone de versiones de pago por la empresa de mismo nombre. Actualmente existe la versión 6.1 pero en este proyecto se utilizó la versión 5.3.

RapidMiner, antes conocido como YALE, se desarrolló a partir de 2001 por Ralf Klinkenberg, Ingo mierswa, y Simon Fischer en el departamento de Inteligencia Artificial de la Universidad Técnica de Dortmund.

RapidMiner proporciona esquemas y modelos y algoritmos de aprendizaje Weka y R scripts que se pueden utilizar a través de extensiones.

RapidMiner está escrito en el lenguaje de programación Java. Y proporciona una interfaz gráfica de usuario para diseñar y ejecutar flujos de trabajo de análisis. Se ejecuta en todos los sistemas operativos.

Realiza transformación de datos, modelado de datos, y métodos de visualización de datos con acceso a una lista completa de fuentes de datos, incluyendo Excel, Access, Oracle, IBM BD2, Microsoft SQL, Sybase, Ingres, MySQL, Postgres, SPSS, dBase. Permitiendo trabajar con fuentes de datos de gran tamaño.

Lo más característico hasta ahora no mencionado es que permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico.

4.5.1. Modelos y operadores

RapidMiner para la ejecución de todas sus funcionalidades pone a disposición del usuario un set de módulos, en los que se asignan los operadores de tratamiento de datos. Existen por ejemplo un modulo de importación o un módulo de exportación, pero los módulos más importantes y usados en este proyecto son el módulo de modelado y el módulo de procesado de texto.

El módulo de modelado de RapidMiner (Modeling), está dividido en siete secciones principales:

1. Classification and regression (53 operadores)
2. Attribute Weighting (21 operadores)
3. Clustering and Segmentation (13 operadores)
4. Association and ítem set mining (6 operadores)
5. Correlation and Dependency Computation (8 operadores)
6. Similarity Computation (4 operadores)
7. Model Aplication (13 operadores)

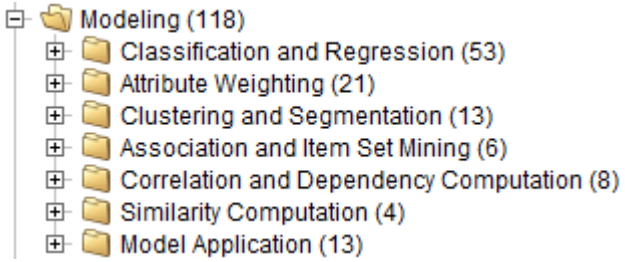


Figura 12: Secciones del modulo de modelado en RapidMiner

De las cuales algunas carpetas están subdividas en sistemas de modelado más específicos

El módulo de procesado de textos de RapidMiner (Text Processing), está dividido en seis secciones principales, además de operadores de origen de datos:

1. Tokenization (1 operador)
2. Extraction (5 operadores)
3. Filtering (12 operadores)
4. Stemming (7 operadores)
5. Transformation (11 operadores)
6. Utility (1 operador)

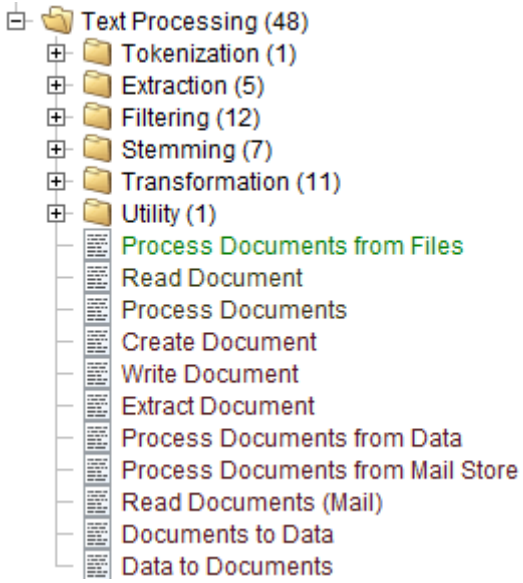


Figura 13: Secciones del modulo de procesado de textos en RapidMiner

Capítulo 5

Criterios utilizados para la evaluación de los algoritmos

Las frases clave proporcionan un breve resumen del contenido de un documento. Como grandes colecciones de documentos tales como las bibliotecas digitales se generalizan, el valor de dicha información resumida aumenta.

Palabras clave y frases clave son particularmente útiles porque se pueden interpretar de forma individual y de forma independiente uno del otro. Pueden ser utilizados en los sistemas de recuperación de información como descripciones de los documentos devueltos por una consulta, como base para índices de búsqueda, como una forma de consulta para una colección, y como una técnica de agrupación documento.

Además, las frases clave pueden ayudar a los usuarios tener una idea del contenido de una colección, proporcionan puntos de entrada de información, facilitan el documento visualmente con las frases importantes; y ofrecen un medio poderoso para medir similitud entre documentos. (Por ejemplo Gutwin *et al*., 1999; Witten, 1999).

Las frases clave suelen ser elegidas de forma manual. En muchos contextos académicos, autores asignan frases clave a los documentos que han escrito. Índices profesionales suelen elegir las frases de un predefinido "vocabulario controlado" para el dominio en cuestión. Sin embargo, la gran mayoría de documentos vienen sin frases clave, y asignarlos manualmente es un proceso tedioso que requiere el conocimiento de la materia. El uso de técnicas automáticas de extracción es potencialmente de gran beneficio.

Hay dos enfoques fundamentalmente diferentes para el problema de generar automáticamente frases clave para un documento: asignación de frase clave y extracción de frase clave. Ambos usan métodos de aprendizaje, y requieren entrenamiento con un conjunto de documentos con frases clave asignadas.

La asignación de frase clave busca seleccionar las frases de un vocabulario controlado que mejor describa un documento. Los datos de entrenamiento asocian un conjunto de documentos con cada frase en el vocabulario, y construye un clasificador para cada frase. Un nuevo documento se procesa por cada clasificador, y se le asigna la frase clave de cualquier modelo que clasifica de forma positiva (por ejemplo, Dumais et al, 1998). Las únicas frases clave que se pueden asignar son los que ya se han visto en los datos de entrenamiento.

La extracción frase clave, no utiliza un vocabulario controlado, pero en su lugar elige palabras clave en el texto en sí. Emplea técnicas de recuperación léxica e información para extraer frases del texto del documento que es probable que lo caractericen (Turney, 2000). En este enfoque, los datos de entrenamiento se usan para ajustar los parámetros del algoritmo de extracción.

5.1. El conjunto de datos

Para evaluar los algoritmos se necesitan bases de datos adecuadas y deseables.

1. El conjunto de datos debe ser lo suficientemente grande. Se dispone de una gran cantidad de comentarios sacados directamente de Facebook para realizar los experimentos.

2. El conjunto de datos debe estar disponible gratuitamente en formato legible por máquina. Cada herramienta tiene una entrada de datos determinada.

3. Los datos disponibles eran comentarios de Facebook que contenían información de la fecha, autor, id y el mensaje del comentario. Se ha realizado un filtro para obtener solamente la parte del mensaje de cada comentario.

4. El conjunto de datos debe contener una amplia variedad de documentos.

De estas propiedades la número 4 es crítica, ya que sería muy difícil evaluar un algoritmo totalmente sin abarcar todos los temas posibles. Pero es imposible encontrar un conjunto de datos que tenga todas las propiedades deseadas. El conjunto de datos de evaluación para este proyecto cumple con las propiedades lo más cerca posible. Alguna diferencia a estas propiedades puede tener un impacto significativo en la tarea de "la indexación de frase clave". Por lo tanto es importante tener en cuenta cómo cualquier determinado conjunto de datos difiere de las propiedades deseadas y cómo esto influye en el rendimiento de los algoritmos.

Pero como se quiere realizar un estudio sobre el uso de los algoritmos en comentarios cortos obtenidos de Facebook, comprobar la eficiencia de cada algoritmo y destacar alguno en particular por sus ventajas, es suficiente.

El tema de los comentarios es desconocido, pueden ser de la misma temática o diferentes temas mezclados, de esta manera los resultados de los algoritmos y herramientas se compararán para ver si el resultado da una referencia exacta sobre el tema general para que un usuario que a priori desconoce la temática de los comentarios pueda tener una idea clara y básica sobre el contenido de los comentarios.

Se disponen dos tipos de comentarios para realizar los experimentos.

El primer tipo son comentarios por usuarios independientes a la marca o empresa que escriben en el perfil de Facebook de una empresa o cliente.

Se disponen de comentarios de usuarios en diferentes perfiles de clientes, por ejemplo Youtube (pelicula Iron Man), Starbucks y diferentes juegos de la empresa King (Candy Crush, Pet Rescue Saga, etc). Los comentarios mezclan varios idiomas destacando inglés, español, francés o portugués. Después de haber realizado un filtro quitando signos de puntuación, interrogración, exclamación o caracteres extraños como #, ~, *, etc... Para dejar los comentarios lo más limpios y puros posibles.

Se comenzó el proyecto con los siguientes comentarios de usuarios mostrados en la siguiente tabla, aunque posteriormente se utilizarían solo para algunos clientes, concretamente de los que se disponía mayor volumen de datos.

Por usuarios		
Cliente	**Num Palabras**	**Num Comentarios**
BubbleWitchSaga	78484	7092
CandyCrushSaga	6262746	677404
DiamondDiggerSaga	71390	8970
FarmHeroesSaga	1734078	193255
PappaPearSaga	1535171	161514
PepperPanicSaga	282160	29941
PetRescueSaga	3339367	372973
PyramidSolitaireSaga	476050	31135
Repsol	35358	2506
Starbucks	1906329	226898
Youtube	2069067	271359

El segundo tipo de comentarios a disposición son los comentarios que escribe la empresa KING en su propio perfil de Facebook en forma de noticias, avances o avisos de sus juegos, posiblemente escritos por sus community manager. Comentarios oficiales y mejor redactados que los comentarios de un usuario particular, pero en menos cantidad de volumen. Estos comentarios se encuentran en cada uno de los perfiles de sus juegos.

Marca empresa KING		
Cliente	**Num Palabras**	**Num Comentarios**
BubbleWitchSaga	1813	62
CandyCrushSaga	7841	310
DiamondDiggerSaga	2052	87
FarmHeroesSaga	7268	320
PappaPearSaga	10269	428
PepperPanicSaga	4673	210
PetRescueSaga	9982	384
PyramidSolitaireSaga	2999	118

5.2. Introducción a los criterios de evaluación

Para evaluar un algoritmo de indexación de tema necesitamos dos cosas.

Primero necesitamos buenos conjuntos de datos en la que los experimentos se pueden ejecutar. Estos han sido presentados anteriormente como un conjunto de comentarios obtenidos de Facebook.

En segundo lugar, en todos los experimentos de todas las herramientas estudiadas, se calcula criterios en los que los resultados de los experimentos pueden compararse entre sí. En esta sección se presentan por primera vez y se explican los diferentes criterios de evaluación.

Los criterios utilizados para la evaluación de los resultados se dividen en dos: criterios cuantitativos y criterios cualitativos.

5.2.1. Criterios Cuantitativos

Los criterios cuantitativos no dan resultados globales porque depende del dispositivo u ordenador donde se ejecuten los experimentos, variando sensiblemente. Pero si pueden ayudarnos a comparar las herramientas de una manera subjetiva.

Volumen de datos

El primer criterio cuantitativo es la capacidad de volumen de datos que puede procesar un algoritmo o herramienta sin ningún tipo de bloqueo o error durante ejecución por falta de memoria.

Este criterio se realiza más por fuerza bruta, llegando a los límites de cada algoritmo o herramienta forzando el error de memoria y obteniendo el volumen total que puede albergar el proceso.

Normalmente este criterio está asociado a la capacidad de memoria del dispositivo donde se realizan los experimentos, por lo que los datos obtenidos no son globales pero sí el criterio ayuda a obtener una idea aproximada de que algoritmo puede procesar mayor número de textos.

Runtime

El tiempo de ejecución es un criterio crítico al comparar los algoritmos. Se determina a qué casos problemáticos de los algoritmos puede ser aplicado. Un algoritmo que toma un largo tiempo para ejecutar es menos útil en una aplicación práctica. Si un conjunto de documentos cambia con frecuencia el algoritmo también se tendría que volver a ejecutar con frecuencia. Así que si un algoritmo tiene una terminación larga de tiempo de los temas para el nuevo documento no estaría disponible durante mucho tiempo para que los usuarios pueden trabajan con la colección de documentos.

También el tiempo de ejecución de los algoritmos mostrará la forma en que se pueden integrar en un sistema más complejo. Si un algoritmo tarda mucho tiempo para completarse, se puede utilizar como un preprocesador, por ejemplo, en un motor de búsqueda. Por otro lado, si sólo necesita un tiempo muy corto para completase se podría utilizar cuando un usuario interactúa con el sistema.

También la complejidad de tiempo nos haría responder a la pregunta de si un algoritmo puede ser utilizado en un sistema interactivo. Al igual que el criterio anterior los resultados no son globales ya que los experimentos se realizaron en el mismo ordenador, y en otros dispositivos el resultado sería diferente. Pero al realizarse en el mismo dispositivo podemos obtener conclusiones y comparar las herramientas y sacar una idea globalizada con los criterios cuantitativos.

5.2.2. Criterios cualitativos

A diferencia de los criterios cuantitativos, los criterios cualitativos si obtienen resultados globales, siempre y cuando se usen los mismos textos origen, independientemente del dispositivo donde se ejecuten las herramientas. Nos permite sacar una idea objetiva para poder comparar los resultados de los experimentos.

Precisión y Recall

Dos de los tres criterios que se usarán son la precisión y recall (exhaustividad). Precisión y recall son dos métricas que pueden utilizarse para medir el rendimiento de recuperación de información o patrón de algoritmos de reconocimiento. En la recuperación de la información el algoritmo devuelve un conjunto de resultados como una respuesta a una consulta proporcionada por el usuario.

Los buscadores de Internet, como www.google.com y us.yahoo.com, realizan tareas de recuperación de información. Como entrada reciben una consulta suministrada por el usuario (una cadena de texto). Dada esta consulta hay páginas web de Internet relevantes y no relevantes para el usuario. Se espera que el motor de búsqueda pueda devolver un conjunto de resultados que contiene en la medida de lo posible las páginas relevantes y en lo menos posible las páginas no relevantes. Para un motor de búsqueda también el ordenamiento del conjunto de resultados es importante, es decir, las páginas más relevantes deben presentarse primero para el usuario. Sin embargo, para la definición de precisión y recall el orden del conjunto de resultados es irrelevante [40].

Para definir la precisión y recall que tenemos que definir formalmente la tarea de recuperación de la información:

• Q es el conjunto de todas las consultas posibles.

• I es un conjunto de posibles elementos que pueden ser devueltos por el algoritmo de recuperación de información.

• La función A representa el algoritmo de recuperación de información

$$A : Q \rightarrow \mathscr{P}(I)$$
$$A : q \mapsto R_q$$

Ecuación 7: Representación del algoritmo de recuperación de información

La función devuelve subconjuntos de posibles artículos.

• Por cada consulta q ∈ Q hay conjuntos:

- $I_q \subseteq I$ es el conjunto de todos los elementos que son relevantes para la consulta.
- $\tilde{I}_q \subseteq I$ es el conjunto de todos los elementos que no son relevantes para la consulta.
- $R_q \subseteq I$ es el conjunto devuelto por el algoritmo de recuperación de información A, definido anteriormente.

• Para cada consulta q ∈ Q y cada artículo i ∈ I solamente una de las siguientes puede ser verdad:

$$i \in I_q$$
$$i \in \tilde{I}_q$$

Ecuación 8: Representación de la existencia de cada artículo en un conjunto

Por ejemplo, un ítem puede ser relevante o no relevante pero no ambos al mismo tiempo.

Para medir el desempeño de la recuperación de la información de un algoritmo A podemos utilizar la precisión y recall.

Para una consulta q la precisión y el recall se definen de la siguiente manera:

• $C_q = \{i \in I : I \in I_q \cap R_q\}$. Cq es el conjunto de todos los elementos relevantes en el conjunto de resultados de la consulta q.

• La precisión de p es la proporción entre el número de artículos relevantes devueltos a la cantidad total de los artículos devueltos:

$$p = \frac{|C_q|}{|R_q|}$$

Ecuación 9: Cálculo del criterio de evaluación precisión

• El recall r es la relación entre el número de artículos relevantes devueltos al número total de artículos pertinentes:

$$r = \frac{|C_q|}{|I_q|}$$

Ecuación 10: Cálculo del criterio de evaluación recall

Ambas medidas (precisión y recall) siempre se deben considerar conjuntamente a la hora de juzgar el desempeño de un algoritmo ya que es fácil maximizar uno de ellos. Tener en cuenta los dos siguientes casos extremos:

1. El resultado R_q devuelto por el algoritmo A es un conjunto de elementos. Si $R_q = \{i\}$ y $I \in I_q$ entonces la precisión sería como máximo 1.0, debido a que el algoritmo no devuelve ningún artículo definido como no relevante. Pero esto sólo lo haría un valor de recall de $r = 1/|I_q|$, debido a que el algoritmo sólo devuelve un elemento relevante.

2. Los resultados R_q devueltos por el algoritmo son siempre $R_q = I$. Recall tendría un máximo de 1.0 ya que todos los productos relevantes son devueltos. Pero sería a la precisión de un valor de $r = |I_q|/|I|$, porque todos los productos no relevantes son también devueltos.

Ambos casos anteriores no son deseables. En el primer caso el algoritmo puede fallar al devolver un gran número de información relevante dada una tarea del mundo real. En el segundo caso, el algoritmo no filtra cualquier información irrelevante, por lo que un usuario tendría que llevar a cabo la búsqueda por sí mismo.

La definición anterior define la precisión y el recall sólo en una consulta. En este trabajo para juzgar el desempeño de un algoritmo en un conjunto de consultas podemos guardar la precisión y el recall de cada consulta y luego calcular el promedio y la desviación estándar de estos valores.

F-measure

También hay una medida que combina la precisión y el recall en uno. Normalmente se calcula esta *F-measure* durante la etapa de extracción [41].

F-measure es una medida que combina la precisión y el recall:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Ecuación 11: Cálculo del criterio de evaluación F-measure

Hay varias razones por las que la F-measure puede ser criticada en circunstancias particulares debido a su sesgo como una métrica de evaluación. También es conocida como la medida F_1, porque el recall y la precisión se ponderan de manera uniforme.

Es un caso especial de la medida general F_β (para los valores reales no negativos de β):

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

Ecuación 12: Cálculo del criterio de evaluación F-measure con β

Otras dos medidas de uso común son la medida F_2, en la cual el peso del recall es mayor que la precisión y la medida $F_{0.5}$, que pone más énfasis en la precisión que en el recall.

La *F-measure* fue derivada por van Rijsbergen (1979) de manera que F_β "mide la eficacia de la recuperación con respecto a un usuario que se conecta β veces dando la importancia a recall como a la precisión". Se basa en la medida de la eficacia de van Rijsbergen:

$$E = 1 - \frac{1}{\frac{\alpha}{P} + \frac{1-\alpha}{R}}$$

Ecuación 13: Cálculo de la eficacia de van Rijsbergen

Su relación es

$$F_\beta = 1 - E$$

Ecuación 14: Relación de F-measure con β y la eficacia de van Rijsbergen

donde

$$\alpha = \frac{1}{1+\beta^2}$$

Ecuación 15: Relación entre α de van Rijsbergen con β de F-measure

Desviación típica

La desviación típica o desviación estándar (denotada con el símbolo σ o s, dependiendo de la procedencia del conjunto de datos) es una medida de dispersión para variables de razón y de intervalo. Se define como la raíz cuadrada de la varianza de la variable [42].

Se calcula aplicando la ecuación XYZ, que se resume en los siguientes pasos:

1. Calcula la media (el promedio de los números).

2. Sumatorio de todas las variables elevadas al cuadrado y dividido por el total de variables de la distribución.

3. Al resultado anterior resta la media elevada al cuadrado, con esto obtener la varianza y finalmente aplicar raíz cuadrada [43].

$$\sigma = \sqrt{\frac{\sum x_i^2}{N} - \bar{x}^2}$$

Ecuación 16: Cálculo de la desviación típica de una distribución de datos

Elevar al cuadrado para que los grandes valores se destaquen. Pero elevarlas al cuadrado hace que la respuesta sea muy grande, así que se aplica la raíz cuadrada y así la desviación típica es mucho más útil.

Esta variable ayuda a conocer la desviación que presentan los datos en su distribución respecto de la media aritmética de dicha distribución.

Intervalo de confianza

En el contexto de estimar un parámetro poblacional, un intervalo de confianza es un rango de valores (calculado en una muestra) en el cual se encuentra el verdadero valor del parámetro, con una probabilidad determinada.

La probabilidad de que el verdadero valor del parámetro se encuentre en el intervalo construido se denomina **nivel de confianza**, y se denota 1-α. La probabilidad de equivocarnos se llama **nivel de significancia** y se simboliza α.

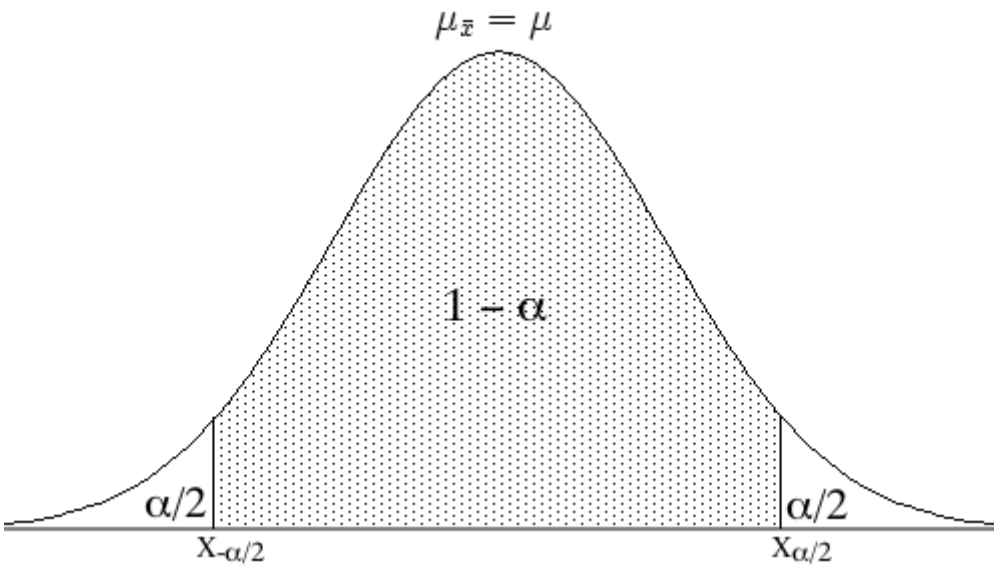


Figura 14: Distribución de los datos limitados por nivel de confianza

La fórmula para calcular el intervalo de confianza a partir de la medias es

$$\bar{x} - Z\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{x} + Z\frac{\sigma}{\sqrt{n}}$$

Ecuación 17: Cálculo de intervalo de confianza

donde Z es el valor crítico de la distribución normal estandarizada

En esta memoria se ha calculado el intervalo de confianza para el 95% ya que es uno de los más utilizados. De esta manera el nivel de confianza es igual a (1- α)*100%, si el nivel de confianza es 95% entonces α = 0.025.

Para calcular el valor de Z sabiendo que 1- α es 0,0975 buscamos ese valor en la tabla Z (anexo A) y se ha obtenido el cruce de la parte derecha zona gris 1.9 y de la parte de arriba 0.06, por lo que Z = 1,96.

Finalmente aplicando la ecuación 16 con la desviación típica y la raíz del total de datos se obtiene el intervalo de confianza.

Capítulo 6

Evaluación y comparación de las herramientas

En este capítulo se detalla la obtención de los resultados y el funcionamiento de cada herramienta para obtenerlos, así como los pasos previos y los pasos finales por cada experimento para realizar una valoración final y una evaluación conjunta.

6.1. Obtención y limpieza de los comentarios

Inicialmente al comenzar el trabajo se disponía de un gran conjunto de información de varias marcas y clientes. Comentarios obtenidos de Facebook tanto de usuarios particulares como de la propia empresa.

Los comentarios se encontraban en gran cantidad de archivos de texto plano, cada comentario tenía la siguiente forma:

```
id: 10152170316869498_36498750
from:   [Mariana Cavalcanti,    id: 100000546034670]
message:    'Add me. Need friends!'
created_time:   2013-08-18T21:56:28+0000
like_count: 29
```

Figura 15: Formato de cada comentario origen

Se han reunido todos los comentarios de todos los archivos de texto plano en un único archivo que contiene toda la información. Un archivo para cada tipo de cliente con toda la información.

De esta manera la parte más importante de cada comentario es la parte de *message* por lo que se ha realizado un filtro para obtener solamente el comentario en sí de cada conjunto de información de comentario. Así se ha creado un único archivo por cada cliente solamente con todos los mensajes comentario.

Aunque como se ha explicado en capítulos anteriores la mayoría de los algoritmos internamente realiza un filtro de los comentarios de toda palabra o frase sin aporte sintáctico o semántico, antes de comenzar con las pruebas para conocer el alcance y funcionamiento de las herramientas se ha realizado una limpieza de los comentarios una vez agrupados en cada uno de los ficheros para cada cliente.

Esta limpieza ha consistido en eliminar todo signo de puntuación: punto, coma, punto y coma, dos puntos, comilla simple y doble, signos de interrogación y exclamación, paréntesis, corchetes, u otro tipo de símbolos como almohadilla #, arroba @, porcentaje %, etc…. Además se ha eliminado todo tipo de código que hacía referencia a emoticonos, ya que en los comentarios de Facebook se puede poner emoticonos pero en los ficheros de texto plano de partida en el proyecto estos emoticonos se encuentran por su código en texto unicode.

Así los comentarios han quedado limpios, lo más posible a contener solamente palabras y no tener ningún posible impedimento o distorsión en los resultados debido a la simbología que no iba a aportar nada a los resultados finales.

6.2. Primeros experimentos de test y primer contacto

Al principio se ha realizado una labor de investigación sobre las posibles herramientas a usar en los experimentos. Después de realizar pequeñas pruebas con textos pequeños para comprender el funcionamiento y manejo de las herramientas inicialmente se ha dispuesto de las siguientes herramientas:

- Maui
- Stanford NLP
- Wandora
- Weka

El algoritmo de Maui nos ha proporcionado dos tipos diferentes de resultados debido a dos tipos diferentes de configuración. La primera usando la propia entrada de datos sobre sí misma al igual que el funcionamiento del resto de herramientas evaluadas, y la segunda usando como fuente de resultados una base de datos de Wikipedia, en este proyecto se ha predispuesto una base de datos de artículos de ciencias naturales y sociales.

Los resultados obtenidos por el algoritmo de Maui con la configuración por defecto, que es el mismo funcionamiento que el resto de herramientas, usando los propios comentarios de entrada como parte del proceso de entrenamiento se ha denominado Maui KEY.
Los resultados obtenidos por el algoritmo de Maui con la configuración de utilizar una base de datos de Wikipedia para buscar el tema a partir de los comentarios se ha denominado Maui WIKI.

Se han realizado, después de los experimentos preliminares para comprobar el funcionamiento de la herramienta, experimentos más reales con los comentarios obtenidos en el primer punto de la evaluación. De esta manera la herramienta Weka no era capaz de devolver resultados con textos medianamente grandes y se obtenía fallos de memoria y el consecuente bloqueo de la herramienta. La herramienta Weka es una gran herramienta para el procesado de textos pero por el inconveniente del volumen de datos que no era capaz de soportar se ha decidido prescindir de su utilización debido a que no podía compararse sus resultados con el resto de herramientas.

Debido a la caída de Weka de la lista de herramientas iniciales se ha buscado un reemplazo para tener el máximo posible de resultados para las comparaciones y evaluaciones en la memoria. La herramienta RapidMiner ha reemplazado su puesto que se encontraba en la reserva de siguientes candidatos a avaluar.

La lista final de herramientas, algoritmos y diferentes configuraciones que se han ejecutado para obtener resultados coherentes entre sí y por tanto poder realizar una evaluación uniforme es la siguiente:

- Maui WIKI
- Maui KEY
- Stanford NLP
- Wandora
- RapidMiner

6.3. Modelado de los comentarios para la evaluación

Una vez definidas las herramientas finales se ha comenzado con los experimentos con todos los datos origen a disposición. Se ha encontrado el primer inconveniente que las herramientas no soportan grandes bloques de comentarios, realizando pruebas se ha llegado a la conclusión de realizar bloques de un máximo de alrededor de 15000 comentarios. De esta manera algunos clientes solamente tienen un bloque por tener menos comentarios de 15000 y otros clientes llegaban a tener 45 bloques de comentarios debido a su mayor volumen de datos origen.

Se han realizado los experimentos con los bloques disponibles de cada cliente y se ha calculado las medidas de precisión y recall. Pero de esta manera para algunos clientes se ha obtenido una o dos veces el cálculo de precisión y recall y para otros clientes veinte o cuarenta veces, por lo que esta manera de realizar los experimentos se ha descartado al no existir una uniformidad entre clientes para realizar comparaciones.

Finalmente se ha decidido coger los cuatro clientes con el mayor volumen de datos para realizar los experimentos.

Por usuarios		
Cliente	**Num Palabras**	**Num Comentarios**
CandyCrushSaga	6262746	677404
PetRescueSaga	3339367	372973
Starbucks	1906329	226898
Youtube	2069067	271359

Se ha creado para cada cliente grupos de comentarios obtenidos aleatoriamente del conjunto global. En concreto 50 grupos de comentarios para cada cliente. Además se han creado conjuntos con diferente volumen de datos para comprobar y expresar mejor el funcionamiento de las herramientas a medida que el volumen de datos aumenta.
Así se han creado 50 grupos de 1000 comentarios, 50 grupos de 2000 comentarios, 50 grupos de 5000 comentarios, 50 grupos de 10000 comentarios y 50 grupos de 15000 comentarios, cada grupo con comentarios aleatorios dentro del conjunto global de cada cliente.

6.4. Ejecución de las herramientas

6.4.1. Maui Indexer

La primera herramienta a evaluar es el algoritmo de Maui, basado en el algoritmo de Kea.

El código de Maui ha sido abierto y ejecutado con el programa Eclipse SDK. Al principio el código no había podido ser ejecutado por la falta de algunas librerías compatibles, en la búsqueda de las librerías necesarias se ha encontrado librerías compatibles para la indexación por Wikipedia, y también para la ejecución correcta del algoritmo.

Se ha configurado un directorio de entrada con el fichero que contiene los comentarios a evaluar, el contenido de este fichero varía por cada experimento con los diferentes grupos de comentarios, y se ha obtenido un fichero de salida con los resultados. El algoritmo de Maui devuelve 10 posibles temas en cada experimento. No fue encontrado en la parte del código donde se habría podido configurar esta variable para aumentar o disminuir el número de temas por experimento. De esta manera y al ser la primera herramienta evaluada, el resto de herramientas han sido configuradas para obtener el mismo número de temas por experimento para poder realizar una evaluación lo más igualada posible para comparar las herramientas y sus resultados entre sí.

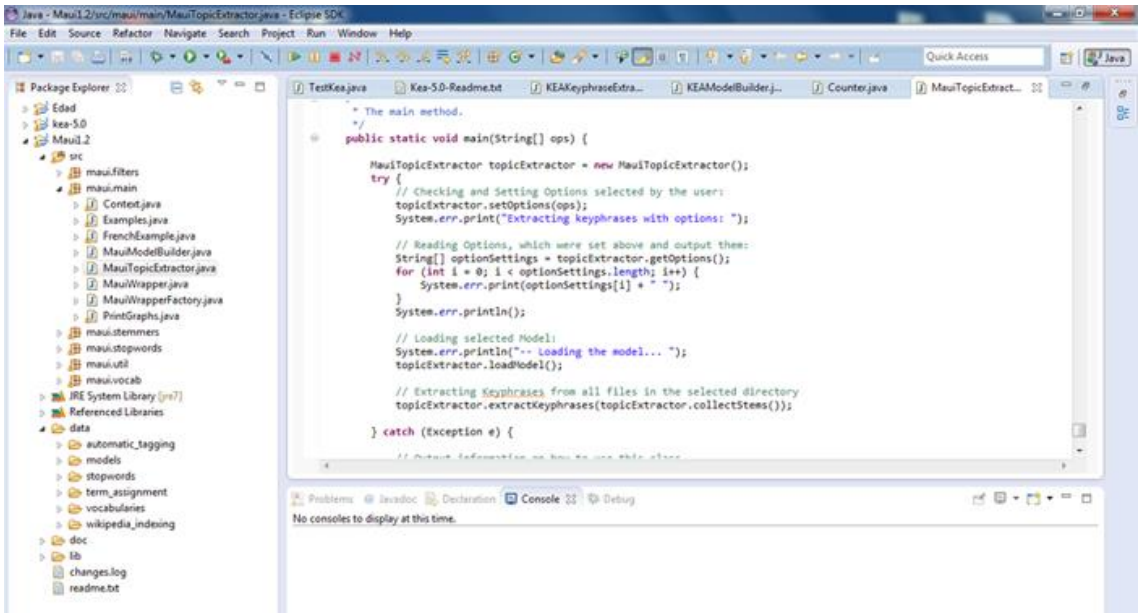


Figura 16: Visualización del código de Maui en Eclipse SDK

6.4.2. Stanford NLP

La siguiente herramienta a evaluar ha sido Stanford Topic Modeling Toolbox GUI de la que se ha mencionad en capítulos anteriores y mostrado en las figuras 8 y 9.

Se han abierto scripts que se han obtenido de la página oficial con la herramienta y han sido ejecutados en orden debido a que cada script genera un directorio o archivo que scripts posteriores usan para realizar los cálculos para dar un resultado final. En estos scripts se han configurado ciertos valores para obtener un resultado acorde con la idea final del proyecto y generalizada para todas las herramientas.

El código del script, donde se ha configurado los parámetros más importante s de salida, se muestra en la figura 16, donde en la línea 1 se escribe el nombre del archivo csv de entrada que debe encontrarse en la misma altura de directorio que el script o scripts.
En la línea 16 es donde se ha configurado que el número de topics por experimento en la salida sea 10.
Mención de la línea 3 donde se configura si los topics del resultado tengan un mínimo de caracteres. Inicialmente se ha configurado a topics de mayor o igual a cuatro caracteres, pero realizando pruebas se ha descubierto que en el caso del cliente *PetRescueSaga* el topic relevante *pet* se perdería y en el caso del cliente Youtube el topic *man* también, sabiendo que los comentarios de Youtube giran alrededor de la película Iron Man. Finalmente se ha configurado que el mínimo para cada topic devuelto sea de tres caracteres.

```
01. val source = CSVFile("pubmed-oa-subset.csv") ~> IDColumn(1);
02.
03. val tokenizer = {
04.   SimpleEnglishTokenizer() ~>            // tokenize on space and punctuation
05.   CaseFolder() ~>                        // lowercase everything
06.   WordsAndNumbersOnlyFilter() ~>         // ignore non-words and non-numbers
07.   MinimumLengthFilter(3)                 // take terms with >=3 characters
08. }
09.
10. val text = {
11.   source ~>                              // read from the source file
12.   Column(4) ~>                           // select column containing text
13.   TokenizeWith(tokenizer) ~>             // tokenize with tokenizer above
14.   TermCounter() ~>                       // collect counts (needed below)
15.   TermMinimumDocumentCountFilter(4) ~>   // filter terms in <4 docs
16.   TermDynamicStopListFilter(30) ~>       // filter out 30 most common terms
17.   DocumentMinimumLengthFilter(5)         // take only docs with >=5 terms
18. }
```

Figura 17: Código script configuración Stanford

La salida de los resultados es devuelta en un archivo csv en un directorio que se configura en los scripts.
Esta herramienta permite realizar una detección de topic mucho más concreta. En la línea 12 de la figura 16 se configura la columna de cada comentario, pero además permite en otras columnas meter la fecha o el autor del comentario y así realizar un análisis teniendo en cuenta estas columnas.
Esto nos dejaría realizar un análisis de los comentarios por ejemplo en agrupaciones de fechas, no era el fin de este proyecto, pero se podía haber obtenido los topics en tiempo, por meses, o por estaciones, o incluso por fechas destacadas como en verano, semana santa o navidades, y realizar una evaluación de topics.

6.4.3. Wandora

Con la herramienta de Wandora se ha usado el clasificador AlchemyAPI, en la figura 11 se ha mostrado el camino de menús para abrir este clasificador.

Este clasificador permite cargar los datos desde fichero o directamente desde la herramienta en la pestaña Raw mostrada en la siguiente figura.

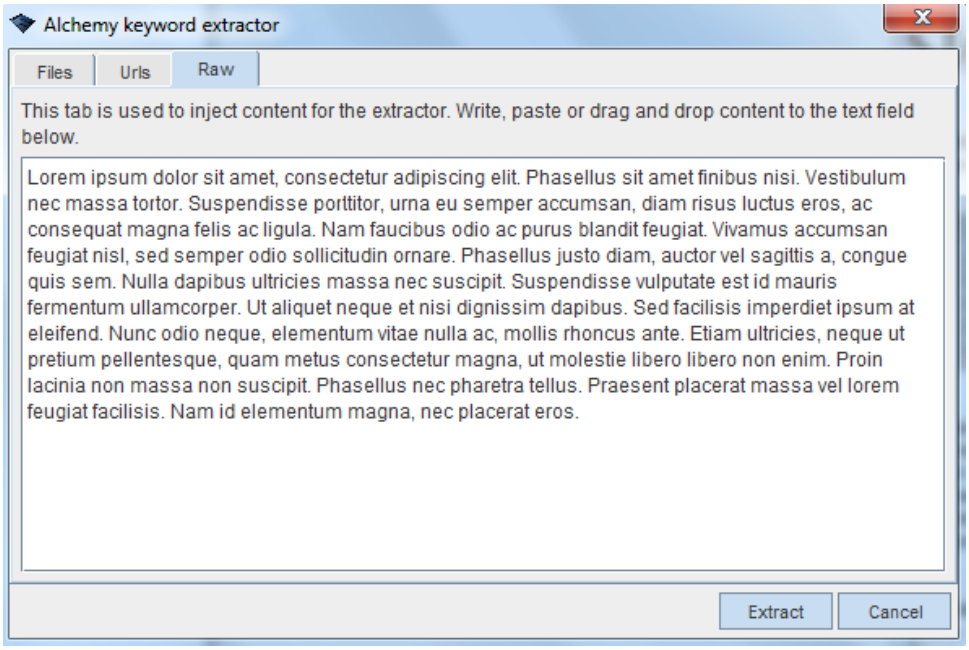


Figura 18: Ventana de introducción de dato a AlchemyAPI

Para obtener los resultados es necesario ingresar una clave que se puede obtener de la página oficial [38] para fines académicos registrando un correo electrónico. Para obtener los resultados es necesario tener conexión a internet porque realiza peticiones a sus bases de datos utilizando su algoritmo interno.

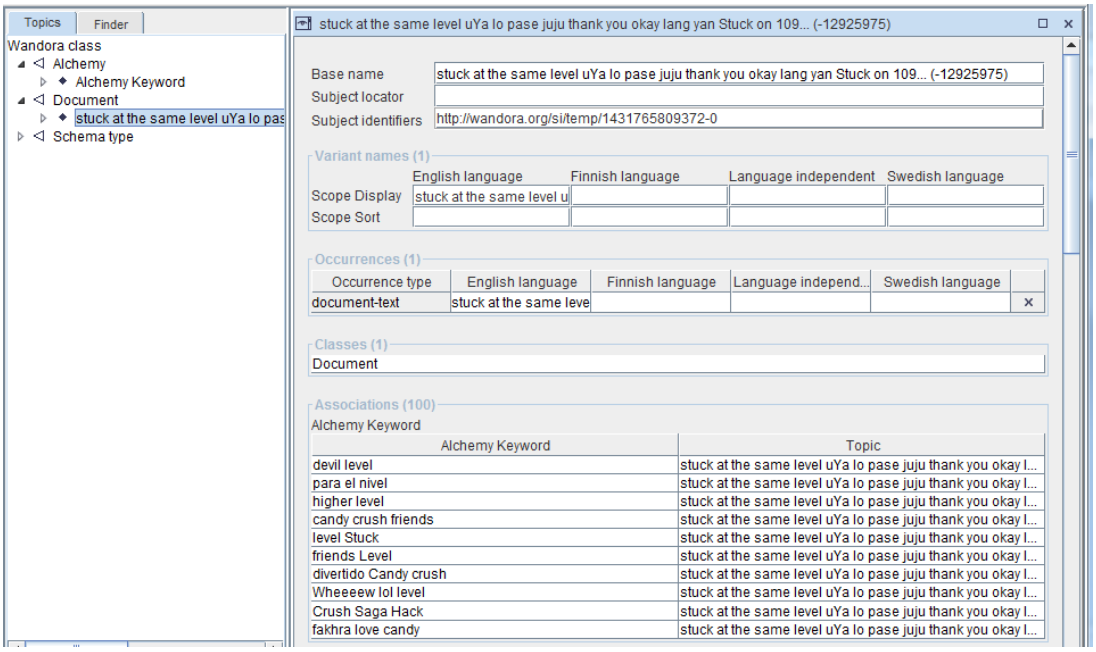


Figura 19: Ventana de resultados de AlchemyAPI

En la figura 19 se ha mostrado los resultados de Wandora con el clasificador AlchemyAPI para cada experimento. La desventaja de esta herramienta es que a medida que vas haciendo experimentos se guardan en memoria en la parte izquierda en el apartado de Document, porque utiliza cada vez más recursos del ordenador, esto hace que poco a poco la herramienta sea más lenta no obteniendo los resultados sino en mostrarlos, porque una vez obtenido el resultado se debe seleccionar el experimento en la parte izquierda y en mostrar los resultados en la parte derecha se vuelve cada vez más lento.

Esta herramienta al igual que Stanford tiene algún extra exclusivo que en esta memoria no se ha utilizado para las evaluaciones finales. Permite obtener resultados clasificando el texto de entrada por idiomas, por categorías predefinidas dentro de la herramienta o clasificar en porcentaje si el texto predonima sentimiento positivo o negativo.

6.4.4. RapidMiner

Con la herramienta RapidMiner se ha construido un modelo a partir de operadores para crear un clasificador de textos y obtener topics similares al resto de herramientas para realizar la evaluación final.

Primero se ha creado un operador para recoger el dato origen de entrada a partir de un archivo guardado en un directorio del ordenador.

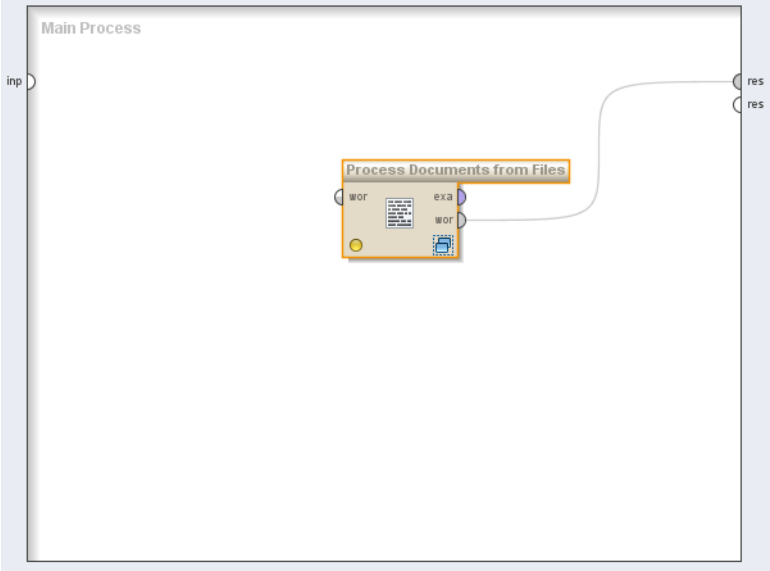


Figura 20: Operador RapidMiner para recoger los datos de entrada

Dentro del operador de recogida de datos de entrada se ha construido el modelo que procesa dichos datos. Se ha compuesto de cinco operadores alineados como muestra la figura 21.

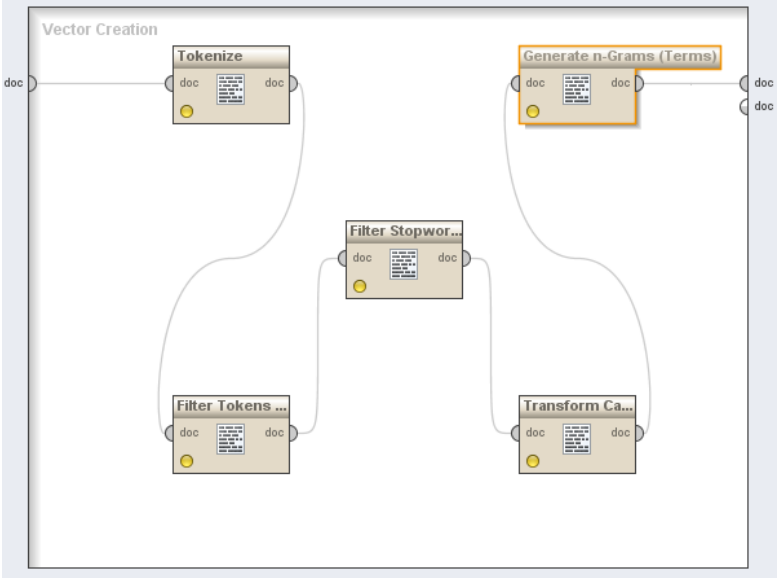


Figura 21: Esquema de operadores del modelo clasificador

Cada operador trabaja con la salida del operador anterior.
El primer operador se ha configurado para obtener diez topics por experimento y se ha encargado de tokenizar el texto de entrada.
El segundo operador se ha configurado para que el mínimo de caracteres por topic obtenido sea mayo o igual a tres caracteres.
El tercer operador se ha encargado de eliminar las palabras de parada.
El cuarto operador se ha encargado de eliminar la diferencia entre plural y singular y ha transformado todo el texto a minúsculas.
El quinto operador se ha encargado de obtener el resultado final con n-gramas con una longitud máxima de dos, por ejemplo un resultado de n-grama de longitud dos será candy_crush, además de obtener candy y crush individualmente pero de longitud uno.
El resultado final se muestra en una tabla por orden de relevancia.

6.5. Cálculo de criterios de evaluación

6.5.1 Criterios cualitativos

La primera herramienta analizada ha sido el algoritmo de Maui que ha devuelto 10 candidatos a topic y esto ha influenciado para que el resto de herramientas se hayan configurado para obtener el mismo número de resultados de salida por experimento.
Por cada salida de 10 topics se han destacado los topics relevantes que verdaderamente han hecho relación a la temática de los comentarios.
¿Pero que topic es relevante o no solamente sabiendo el cliente al que pertenecen los comentarios? Como para cada tipo de persona el resultado es subjetivo y un topic que para una persona es relevante y para otra persona no puede serlo se ha realizado un compendio.

Una vez se han extraído todos los resultados de cada uno de los 50 grupos por bloque de comentarios para cada cliente, se ha contado con la opinión de tres personas que independientemente entre ellas han declarado que topics devueltos por cada herramienta eran o no relevantes. Después se ha cogido sus opiniones y se ha declarado que un topic es relevante si al menos dos de las tres personas han declarado que el topic era relevante.

Han existido topics que se han considerado relevantes porque dos de las tres personas así lo han declarado y topics que se han descartado como relevantes porque solamente a una persona así le ha parecido. Pero la mayoría de los topics relevantes han sido declarados así porque las tres personas a la vez lo han declarado.

Estos topics que no han dejado lugar a duda de su relevancia e importancia sobre la temática de los comentarios generalmente han sido el propio nombre o parte del nombre del cliente, en el caso de los juegos de la marca KING, o han sido el producto estrella de la temática como el caso del café en Starbucks o la película de Iron Man en el caso de Youtube. Aunque esto se desarrolla en apartados siguientes.

Una vez se ha obtenido un listado con los topics relevantes que en alguna ocasión han aparecido en la salida de los resultados para cada cliente, se ha hecho un recuento del total de topics relevantes diferentes que han aparecido durante los experimentos por cada bloque de 50 conjuntos de comentarios. Es decir, cinco totales de topics relevantes para el conjunto completo de 1000, 2000, 5000, 10000 y 15000 comentarios.

Con los resultados totales de topics se han calculado los criterios cualitativos, precisión, recall y F-measure. Con un ejemplo sencillo suponiendo solamente dos experimentos se explica su cálculo. Si las salidas de los dos experimentos han sido la siguiente:

update, **game**, **stuck on level**, **candy crush**, waiting, pls, **play**, **crush**, man, **play level**

candy crush, **crush**, days, **stuck on level**, **game**, **play**, mi blez, waiting, **level**, **friends**

En verde se ha marcado los topics relevantes, entonces en el ejemplo se ha obtenido 8 topics relevantes diferentes. Además todas las herramientas se han configurado para obtener diez topics por experimentos. Con estos datos se ha calculado la precisión y el recall.

La precisión es la división del número de relevantes obtenido en el experimento entre el total de topics obtenido en el resultado, que en este caso y en nuestra memoria ha sido siempre diez. Para el ejemplo anterior la precisión para el primer resultado será 6/10 = 0.6, y para el segundo resultado será 7/10 = 0.7, precisiones altas ya que la mayoría del resultado se ha considerado de importante relevancia.

El recall es la división del número de relevantes obtenido en el experimento entre el total de relevantes diferentes que han aparecido en todo el conjunto de experimentos, en el ejemplo anterior de dos experimentos los topics relevantes no repetidos ha sido 8. Entonces el recall para el primer resultado será 6/8 = 0.75, y para el segundo resultado será 7/8 = 0.875.

F-measure se calcula para cada experimento utilizando la ecuación 11, explicada en el capítulo anterior, usando la precisión y recall calculados previamente por experimento.

Se ha explicado con un ejemplo sencillo de dos experimentos el cálculo de precisión y recall, en el proyecto se realiza sobre el conjunto de 50 grupos con comentarios aleatorios. Por lo que se han calculado 50 medidas de precisión, recall y F-measure por cada conjunto total de 1000, 2000, 5000, 10000 y 15000 comentarios.

Para obtener una única medida de cada criterio cualitativo se ha realizado la media de los 50 valores obtenidos, y se ha calculado la desviación típica y el intervalo de confianza. Así se ha obtenido un único valor de cada criterio para cada conjunto total de 1000, 2000, 5000, 10000 y 15000 comentarios, y que es tal y como se muestra en las gráficas de cada criterio para cada herramienta o algoritmo y agrupadas por cada cliente para realizar una comparación y evaluación de los resultados.

Por último para cada valor se ha calculado su desviación típica utilizando la media en el cálculo. Y también se ha calculado el intervalo de confianza, se han utilizado las formulas detalladas en el capitulo anterior.

De esta manera finalmente se han obtenido los valores finales de los criterios de evaluación cualitativos, precisión, recall y F-measure, y que se han mostrado en gráficas en los siguientes apartados con una tabla con la información de la media, representada en el gráfico, con su desviación típica, y otra tabla con el valor de intervalo de confianza.

6.5.2 Criterios cuantitativos

El primer criterio cuantitativo es el volumen de datos que puede soportar la herramienta para dar un resultado correcto y no interrumpir su resultado por algún fallo o error de memoria.

Este criterio se ha realizado a prueba y error, a fuerza bruta aumentando el volumen de datos de entrada hasta conseguir el número de comentarios aproximado para hacer fallar el algoritmo o herramienta.

El segundo criterio cuantitativo es el tiempo de ejecución de cada herramienta, que sirve para valorar además de que algoritmo devuelve mejor resultado y además es más rápido en devolver ese resultado.

El tiempo de ejecución se ha medido en segundos y se ha obtenido una medida de tiempo por cada experimento, al igual que la precisión o recall, y al igual que estos criterios cualitativos se ha realizado una media con todos los tiempos por cada grupo de 50 experimentos de cada bloque de 1000, 2000, 5000, 10000 y 15000 comentarios. Para esta media se ha calculado la desviación típica y los intervalos de confianza de la misma manera que en el cálculo de la precisión o recall. Estos valores se han representado en siguientes apartados en gráficas de runtime con una tabla con la media, representada en la gráfica, y con la desviación típica y otra tabla con el intervalo de confianza por cada valor para cada grupo todas las herramientas por cliente en la misma gráfica para una mejor comparación y evaluación.

6.6. Resultados y evaluaciones finales

En este apartado se analizarán los resultados de los criterios calculados. Se exponen gráficos con los criterios precisión, recall, F-measure y runtime para cada uno de los cuatro clientes finales. Cada gráfico consta de dos tablas, la primera con la media de todos los experimentos y con su desviación típica, en la segunda tabla la media se ha representado como la variable x junto a su intervalo de confianza de 95%.

Primero se mostrarán las gráficas seguidas de cada criterio de evaluación para realizar una comparación además de entre herramientas por cliente una comparación del resultado entre clientes.

Después una pequeña evaluación de las herramientas por volumen de datos independientemente del dato entre clientes.

Y finalmente una conclusión y evaluación final de las herramientas finales seleccionadas.

Las gráficas de los criterios cualitativos tienen el eje vertical en porcentaje siendo el máximo el 100%. En las gráficas del criterio de runtime el eje vertical esta medido en segundos. En todas las gráficas el eje horizontal indica el volumen de comentarios aleatorios de los experimentos. Representando en cada gráfica la media de todos los experimentos para cada criterio.

Analizando el resultado de topics relevantes para cada herramienta encontramos patrones similares. Para los clientes de CandyCrushSaga y PetRescueSaga los topics más repetidos son el nombre del juego: candy, crush, pet, rescue. También el topic game o play es muy frecuente. Y como resultado inesperado el topic stuck on level o help level, también muy repetido, y que nos ha llevado a la conclusión de que los comentarios mayoritariamente son para pedir ayuda o guía para completar un nivel del juego.

Para el cliente Starbucks el topic más repetido es coffee y el propio nombre de la marca, seguido de todos los tipos de café.

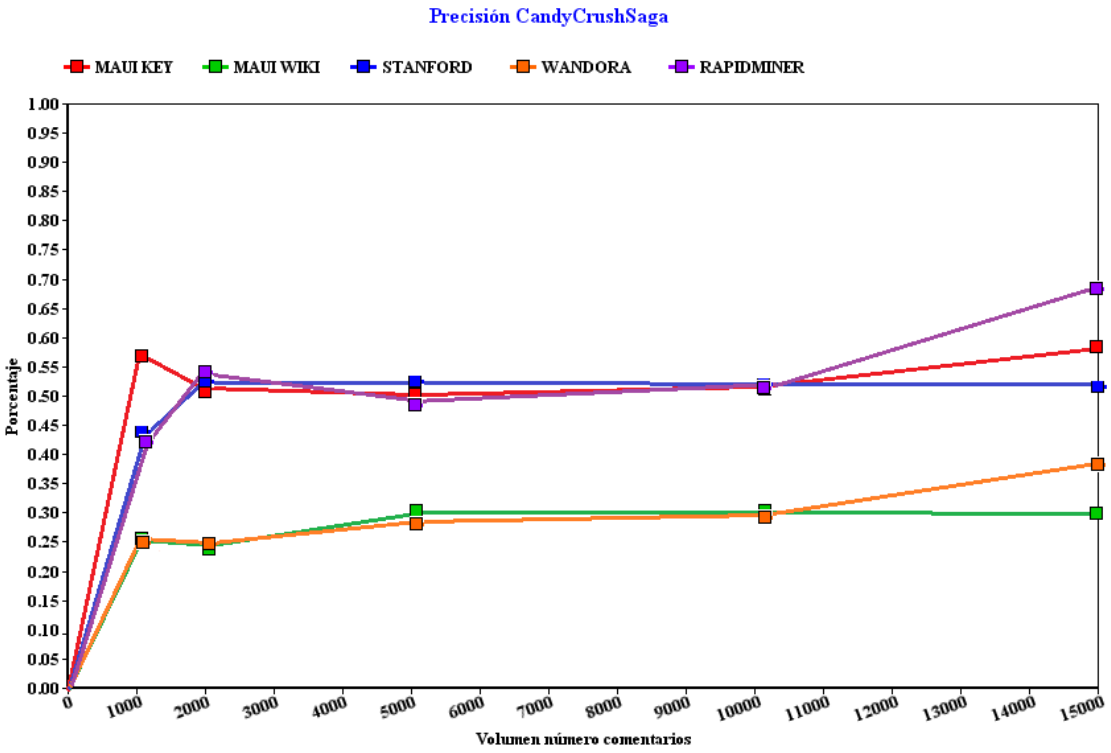
Para el cliente Youtube, comentarios de la película Iron Man se han obtenido topics relevantes del propio nombre de la película y del nombre del personaje principal Tony Stark así como el topic Mandarin, por lo que se ha concluido que los comentarios pertenecen a la tercera entrega de la saga.

A continuación se muestra las gráficas de evolución de las herramientas a mayor número de comentarios en el experimento para los criterios cualitativos.

La precisión para el cliente CandyCrushSaga parece tener dos vertientes.
La primera vertiente las herramientas Maui KEY, Stanford y RapidMiner han obtenido precisiones parecidas obteniendo mayor precisión a mayor volumen de comentarios.

Para Maui KEY y RapidMiner para un volumen de 15000 comentarios por experimento la precisión aumenta, pero para Stanford parece que la precisión se mantiene estable a partir de conjuntos de 2000 comentarios.
La segunda vertiente tiene la misma forma que la primera pero con una precisión a la mitad, para las herramientas de Maui WIKI y Wandora. Para Wandora la precisión aumenta a medida que tenemos mayor volumen de datos mientras que Maui WIKI se mantiene más o menos igual.
Maui KEY ha obtenido mejor medida de precisión que Maui WIKI pero con mayor desviación típica trabajando bajo el mismo algoritmo. De hecho Maui WIKI ha obtenido la menor desviación típica junto con el algoritmo empleado por Stanford.

RapidMiner muestra principios de obtener las mejores precisiones aunque con unas desviaciones típicas un poco altas, significando que los resultados de topics relevantes son un poco variantes entre experimentos.
Para Stanford, Maui KEY y RapidMiner las precisiones indican que más o menos la mitad de los topics de los resultados son relevantes, mientras que para Wandora o Maui WIKI solamente un cuarto de los topics del resultado son relevantes (Figura 22).

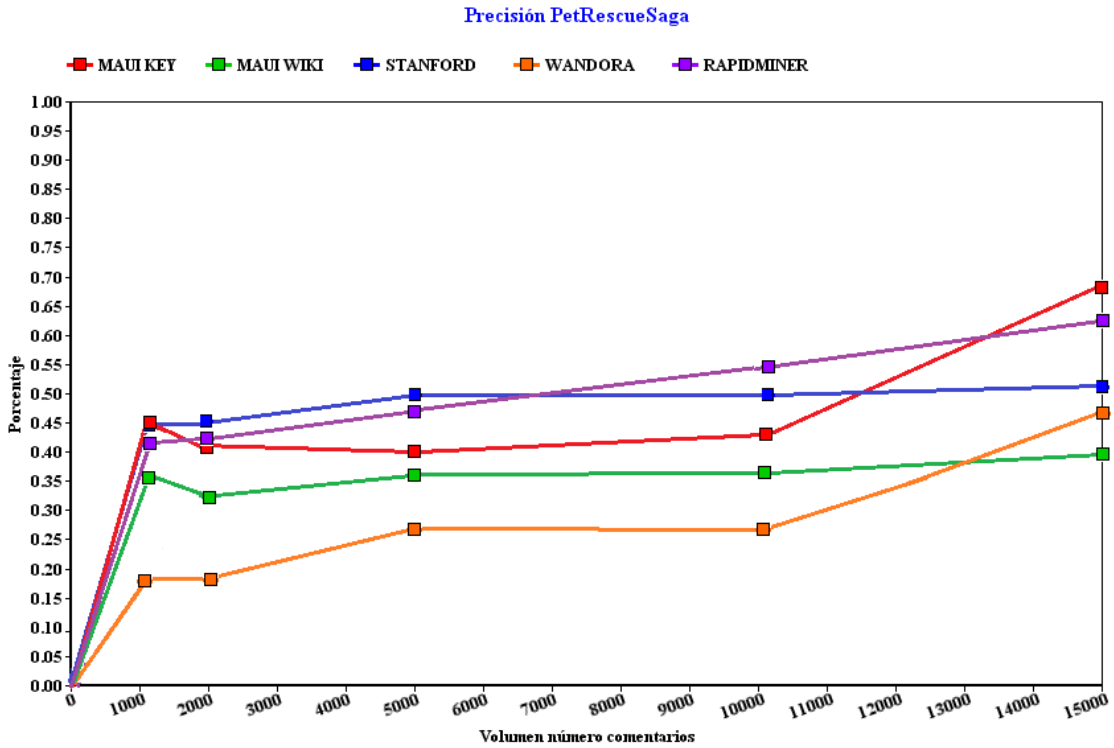


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,56 ± 0,18	0,255 ± 0,09	0,43 ± 0,1	0,25 ± 0,14	0,41 ± 0,1
Grupos 2000 comentarios	0,51 ± 0,17	0,246 ± 0,085	0,52 ± 0,07	0,248 ± 0,15	0,54 ± 0,09
Grupos 5000 comentarios	0,5 ± 0,146	0,29 ± 0,08	0,52 ± 0,1	0,27 ± 0,15	0,48 ± 0,09
Grupos 10000 comentarios	0,53 ± 0,056	0,3 ± 0,07	0,528 ± 0,075	0,28 ± 0,176	0,52 ± 0,12
Grupos 15000 comentarios	0,6 ± 0,22	0,29 ± 0,09	0,51 ± 0,078	0,38 ± 0,22	0,68 ± 0,15

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,05	x ± 0,025	x ± 0028	x ± 0,039	x ± 0,028
Grupos 2000 comentarios	x ± 0,048	x ± 0,023	x ± 0,02	x ± 0,04	x ± 0,025
Grupos 5000 comentarios	x ± 0,04	x ± 0,023	x ± 0,028	x ± 0,042	x ± 0,025
Grupos 10000 comentarios	x ± 0,04	x ± 0,02	x ± 0,021	x ± 0,048	x ± 0,035
Grupos 15000 comentarios	x ± 0,06	x ± 0,025	x ± 0,02	x ± 0,06	x ± 0,043

Figura 22: Gráfico evolutivo de precisión para CandyCrushSaga

La precisión para el cliente PetRescueSaga tiene similitudes con CandyCrushSaga, Maui KEY ha bajado en precisión y Maui WIKI ha subido en precisión respecto a CandyCrushSaga y Maui KEY sigue aumentando su precisión a mayor número disponible de comentarios, siendo el que ha conseguido mejor resultado. Stanford y RapidMiner se comportan muy similares, con RapidMiner aumentando la precisión a mayor volumen de comentarios y Stanford manteniéndose estable. Las desviaciones típicas se comportan de manera similar a CandyCrushSaga posiblemente debido a que son dos juegos de la misma empresa y tienen el mismo tipo de comentarios por parte de los usuarios.
Ahora RapidMiner tiene menos desviación y aunque a mayor volumen le supera la precisión de Maui KEY con casi un 70% en volumen de 15000 comentarios, la diferencia es pequeña, entonces la desviación hace que RapidMiner sea mejor (Figura 23).

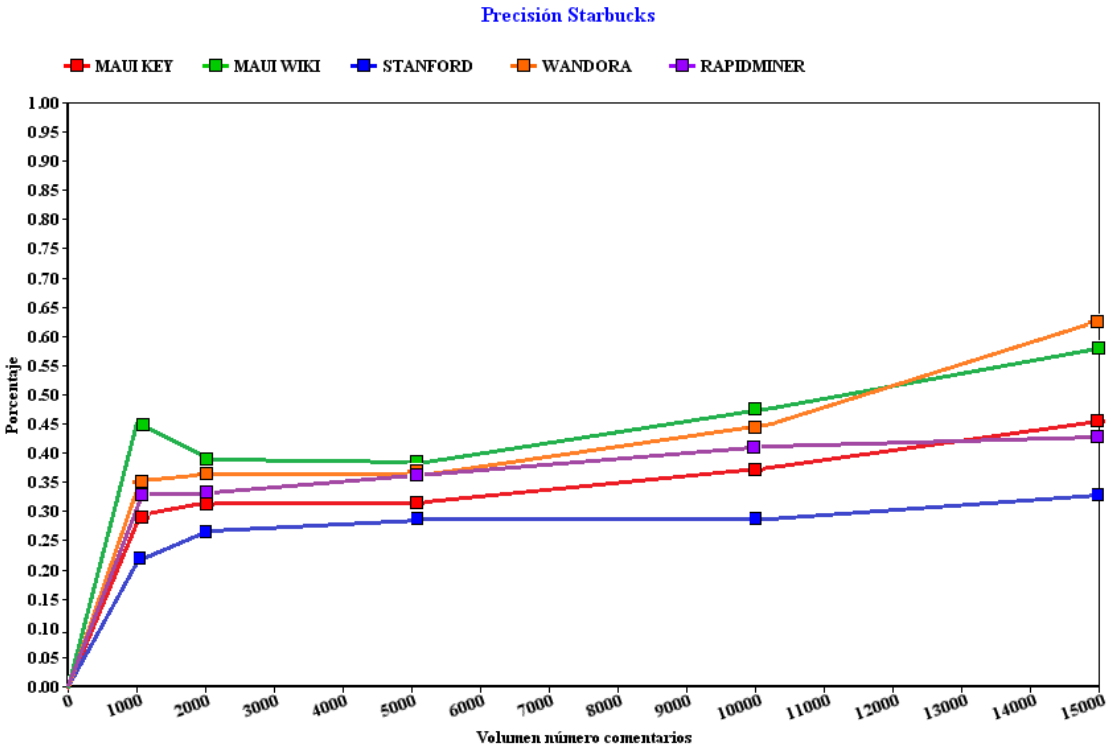


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,456 ± 0,14	0,35 ± 0,1	0,45 ± 0,08	0,18 ± 0,13	0,416 ± 0,08
Grupos 2000 comentarios	0,416 ± 0,16	0,33 ± 0,08	0,45 ± 0,12	0,18 ± 0,13	0,428 ± 0,096
Grupos 5000 comentarios	0,4 ± 0,15	0,36 ± 0,08	0,49 ± 0,07	0,268 ± 0,15	0,46 ± 0,075
Grupos 10000 comentarios	0,428 ± 0,16	0,368 ± 0,09	0,49 ± 0,05	0,26 ± 0,17	0,55 ± 0,06
Grupos 15000 comentarios	0,69 ± 0,17	0,38 ± 0,07	0,5 ± 0,1	0,45 ± 0,14	0,63 ± 0,06

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,038	x ± 0,027	x ± 0,022	x ± 0,035	x ± 0,023
Grupos 2000 comentarios	x ± 0,04	x ± 0,022	x ± 0,033	x ± 0,035	x ± 0,026
Grupos 5000 comentarios	x ± 0,04	x ± 0,022	x ± 0,021	x ± 0,04	x ± 0,02
Grupos 10000 comentarios	x ± 0,04	x ± 0,027	x ± 0,015	x ± 0,05	x ± 0,016
Grupos 15000 comentarios	x ± 0,05	x ± 0,021	x ± 0,027	x ± 0,038	x ± 0,016

Figura 23: Gráfico evolutivo de precisión para PetRescueSaga

En el cálculo de la precisión para el cliente Starbucks se han obtenido diferencias respecto a los dos clientes anteriores. Todas las herramientas tienen un espectro parecido y sufren una subida de precisión a mayor número de comentarios. Wandora es la herramienta que mejor precisión ha obtenido. Curiosamente las dos herramientas que para los clientes anteriores habían obtenido las menores precisiones ahora son las dos que mayores precisiones obtienen para comentarios de Starbucks, Wandora y Maui WIKI. Y Maui KEY y Stanford tienen el mismo comportamiento pero las precisiones son más bajas, la única herramienta que mantiene su comportamiento con precisiones similares a los comentarios de los juegos es RadidMiner. Comparando desde el mismo algoritmo Maui WIKI es mejor que Maui KEY para comentarios de Starbucks.
Para Stanford la precisión ha bajado respecto a los juegos a menos de un cuarto de topics relevantes por experimento (Figura 24).

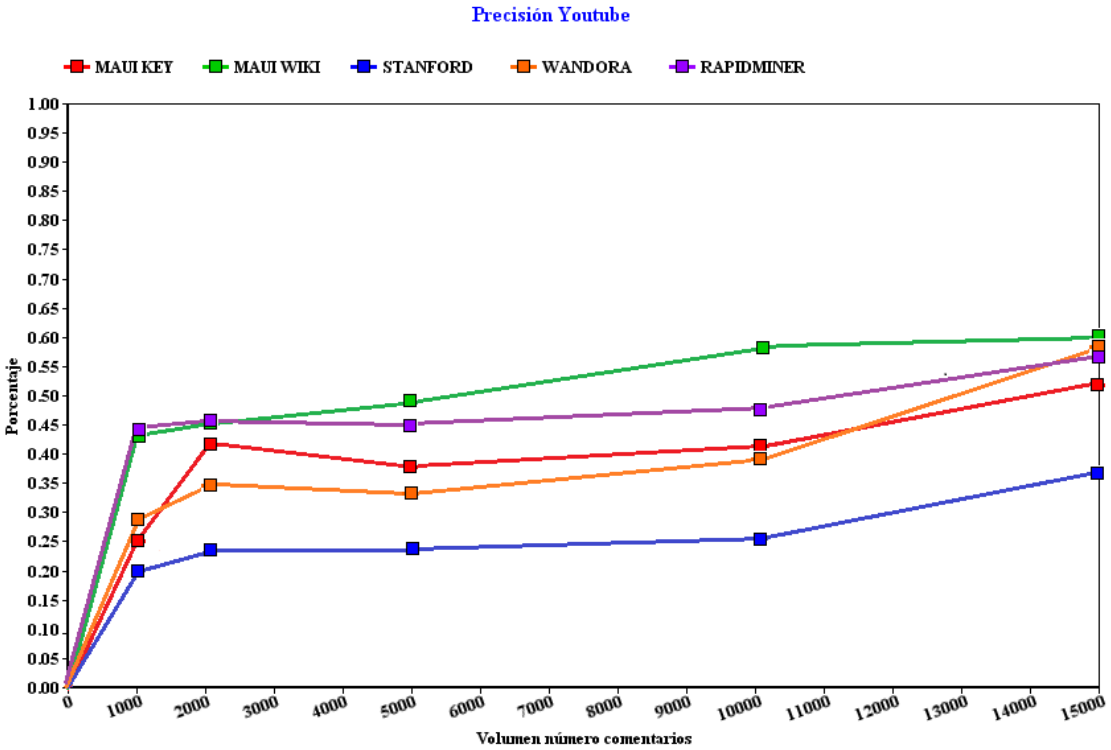


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,29 ± 0,14	0,45 ± 0,12	0,21 ± 0,1	0,35 ± 0,12	0,32 ± 0,09
Grupos 2000 comentarios	0,32 ± 0,19	0,38 ± 0,12	0,26 ± 0,12	0,36 ± 0,24	0,32 ± 0,05
Grupos 5000 comentarios	0,32 ± 0,19	0,37 ± 0,12	0,28 ± 0,1	0,36 ± 0,2	0,36 ± 0,11
Grupos 10000 comentarios	0,36 ± 0,19	0,48 ± 0,095	0,28 ± 0,12	0,45 ± 0,14	0,38 ± 0,17
Grupos 15000 comentarios	0,46 ± 0,12	0,57 ± 0,12	0,31 ± 0,13	0,64 ± 0,17	0,42 ± 0,18

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,04	x ± 0,035	x ± 0,028	x ± 0,035	x ± 0,024
Grupos 2000 comentarios	x ± 0,05	x ± 0,035	x ± 0,035	x ± 0,066	x ± 0,013
Grupos 5000 comentarios	x ± 0,05	x ± 0,035	x ± 0,028	x ± 0,055	x ± 0,03
Grupos 10000 comentarios	x ± 0,05	x ± 0,026	x ± 0,035	x ± 0,038	x ± 0,047
Grupos 15000 comentarios	x ± 0,035	x ± 0,035	x ± 0,036	x ± 0,047	x ± 0,05

Figura 24: Gráfico evolutivo de precisión para Starbucks

Para los comentarios de Youtube la precisión también aumenta a mayor número de comentarios en el experimento. Stanford y Maui KEY obtienen menor precisión que con los juegos KING pero resultados parecidos a Starbucks. RapidMiner se mantiene con resultados parecidos a los otros clientes. Wandora obtiene mejor precisión que los juegos KING pero menor que para Starbucks, donde era la herramienta de mayor precisión a mayor volumen. Ahora es Maui WIKI quien tiene esa mejor precisión y se comporta de manera similar a los comentarios de Starbucks. Para los comentarios de Youtube y para el mayor volumen de datos existen cuatro herramientas o algoritmos que han conseguido una precisión parecida (Maui WIKI, Wandora, RapidMiner y Maui KEY) y es difícil decir cual es mejor para este caso. En todo caso decir que RapidMiner obtiene unas desviaciones más pequeñas por lo que ha obtenido resultados más parecidos en todos los experimentos (Figura 25).



MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,25 ± 0,14	0,43 ± 0,09	0,2 ± 0,05	0,27 ± 0,19	0,44 ± 0,12
Grupos 2000 comentarios	0,41 ± 0,15	0,45 ± 0,19	0,24 ± 0,055	0,35 ± 0,17	0,45 ± 0,1
Grupos 5000 comentarios	0,38 ± 0,1	0,49 ± 0,12	0,24 ± 0,04	0,33 ± 0,22	0,44 ± 0,09
Grupos 10000 comentarios	0,42 ± 0,12	0,57 ± 0,13	0,25 ± 0,076	0,38 ± 0,2	0,46 ± 0,11
Grupos 15000 comentarios	0,53 ± 0,15	0,58 ± 0,1	0,35 ± 0,086	0,56 ± 0,22	0,55 ± 0,05

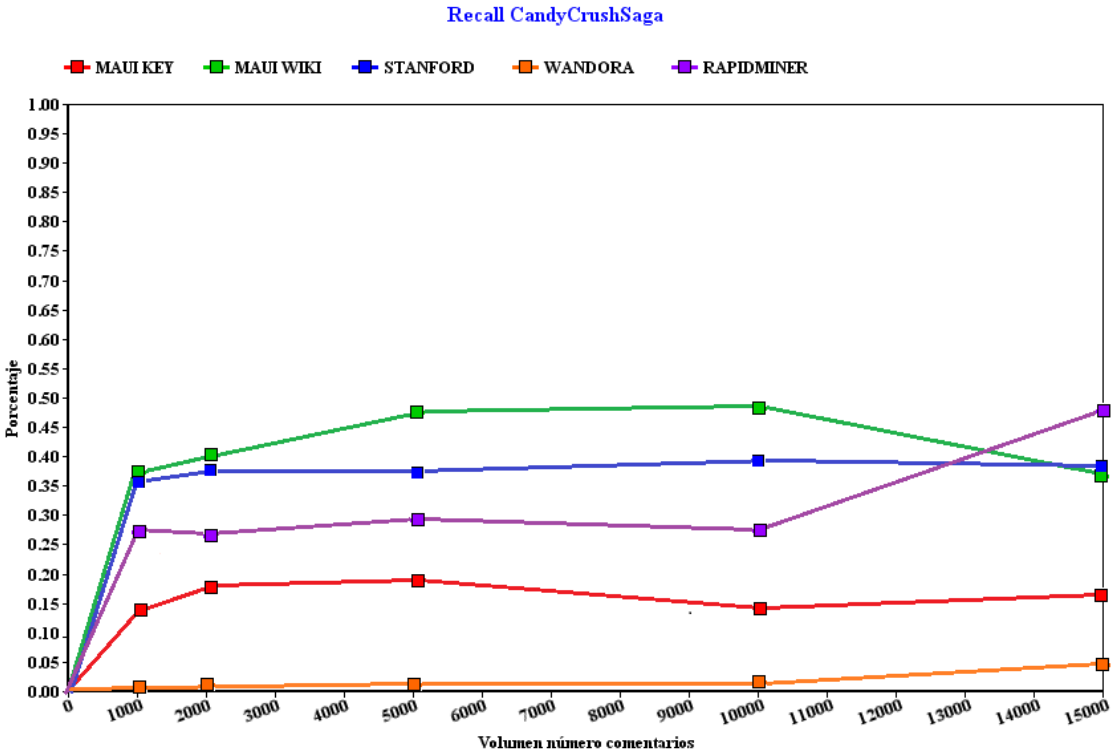
INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,02	x ± 0,04	x ± 0,017	x ± 0,012	x ± 0,052
Grupos 2000 comentarios	x ± 0,019	x ± 0,04	x ± 0,027	x ± 0,008	x ± 0,019
Grupos 5000 comentarios	x ± 0,023	x ± 0,036	x ± 0,024	x ± 0,012	x ± 0,017
Grupos 10000 comentarios	x ± 0,22	x ± 0,22	x ± 0,027	x ± 0,01	x ± 0,027
Grupos 15000 comentarios	x ± 0,019	x ± 0,03	x ± 0,027	x ± 0,006	x ± 0,026

Figura 25: Gráfico evolutivo de precisión para Youtube

La desviación típica para Wandora y Maui WIKI aumenta para Starbucks y Youtube respecto a los juegos. Indica que el resultado de la precisión es alta pero con varianza entre experimentos.
Para Maui KEY y RapidMiner la desviación se mantiene pero para Stanford la desviación es la más baja de todos los clientes indicando que obtiene precisión muy similar para todo el conjunto de 50 experimentos de cada volumen de datos (Figuras 24 y 25).

Una mayor recall indica que la herramienta ha devuelto menos topics diferentes en total, es decir que en la mayoría de experimentos se ha obtenido los mismos topics.
Para el cliente CandyCrushSaga se ha obtenido diferentes comportamientos.
Wandora tiene un recall muy bajo debido a que devuelve muchos topics relevantes diferentes y esto indica que no es muy exhaustivo en el proceso.

Maui KEY es la segunda herramienta con peor recall durante todo el volumen de datos. RapidMiner se encuentra entre medio pero finalmente obtiene un mejor recall a mayor volumen de datos a diferencia de Maui WIKI que tiene mejor recall hasta llegar al mayor volumen de datos donde recibe un bajón importante deduciendo que a mayor volumen de datos se obtiene mayor número de topics relevantes diferentes. Stanford se mantiene estable para cualquier tipo de volumen de datos obteniendo un buen valor de recall respecto al resto de herramientas. La desviación típica para todas las herramientas es baja indicando valores de recall parecidos durante los 50 experimentos de cada bloque de volumen de datos, a excepción de Maui WIKI que ha obtenido las mejores puntuaciones de recall pero sus desviaciones son un poco más altas (Figura 26).

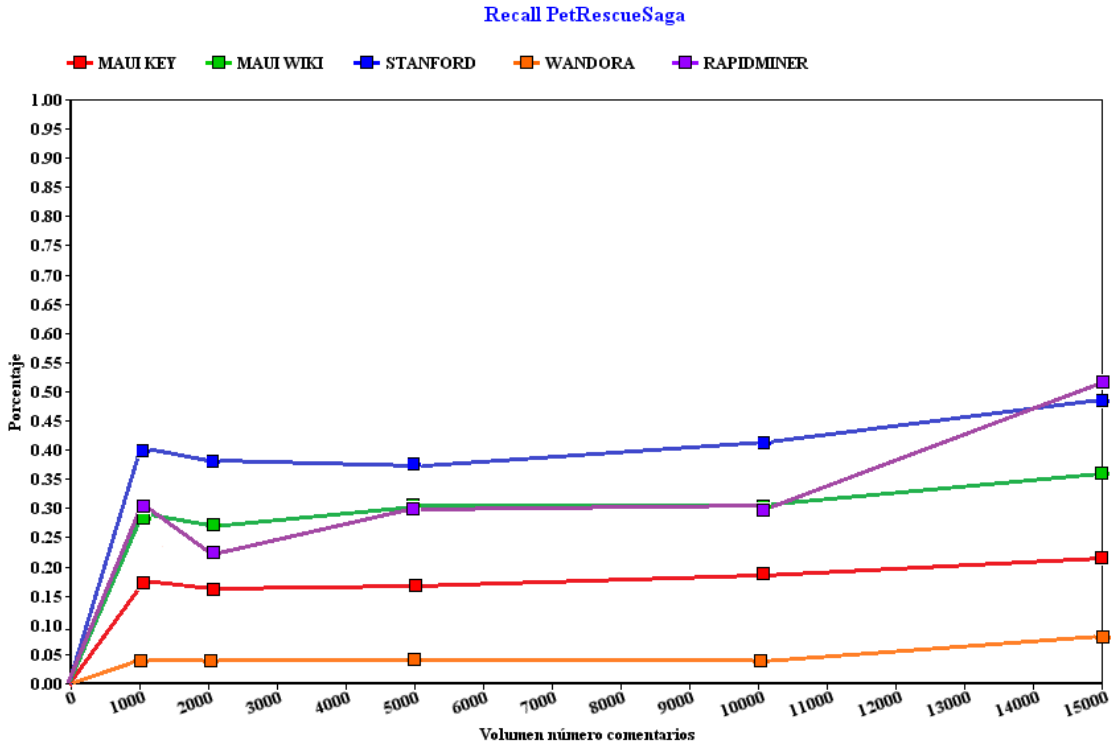


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,14 ± 0,03	0,36 ± 0,14	0,356 ± 0,076	0,01 ± 0,027	0,288 ± 0,021
Grupos 2000 comentarios	0,18 ± 0,08	0,41 ± 0,13	0,37 ± 0,063	0,0248 ± 0,01	0,28 ± 0,06
Grupos 5000 comentarios	0,19 ± 0,054	0,48 ± 0,14	0,37 ± 0,08	0,027 ± 0,015	0,3 ± 0,045
Grupos 10000 comentarios	0,148 ± 0,047	0,498 ± 0,12	0,4 ± 0,072	0,028 ± 0,13	0,26 ± 0,06
Grupos 15000 comentarios	0,16 ± 0,05	0,37 ± 0,07	0,39 ± 0,056	0,031 ± 0,021	0,489 ± 0,08

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,008	x ± 0,04	x ± 0,02	x ± 0,007	x ± 0,019
Grupos 2000 comentarios	x ± 0,022	x ± 0,037	x ± 0,018	x ± 0,004	x ± 0,017
Grupos 5000 comentarios	x ± 0,015	x ± 0,04	x ± 0,023	x ± 0,004	x ± 0,012
Grupos 10000 comentarios	x ± 0,13	x ± 0,034	x ± 0,02	x ± 0,0037	x ± 0,017
Grupos 15000 comentarios	x ± 0,014	x ± 0,02	x ± 0,015	x ± 0,006	x ± 0,023

Figura 26: Gráfico evolutivo de recall para CandyCrushSaga

Para PetRescueSaga todas las herramientas tienen mejor recall a mayor volumen de datos de entrada. Wandora tiene el mismo comportamiento que CandyCrushSaga aunque un recall un poco más elevado.
Maui KEY se comporta de manera similar al igual que RapidMiner aunque tenga un comportamiento un poco extraño, pero obteniendo el mejor recall al igual que con CandyCrushSaga
Es Maui WIKI quien ha bajado su recall indicando mayor número de topics diferentes en los resultados.
Justo al contrario que Stanford que con PetRescueSaga obtiene mejor recall y por tanto ha devuelto los mismos topics durante los experimentos sin topics relevantes diferentes.
Pero a grandes rasgos el comportamiento es parecido entre los juegos indicando que los comentarios son similares por parte de los usuarios siendo de la misma temática (Figura 27).

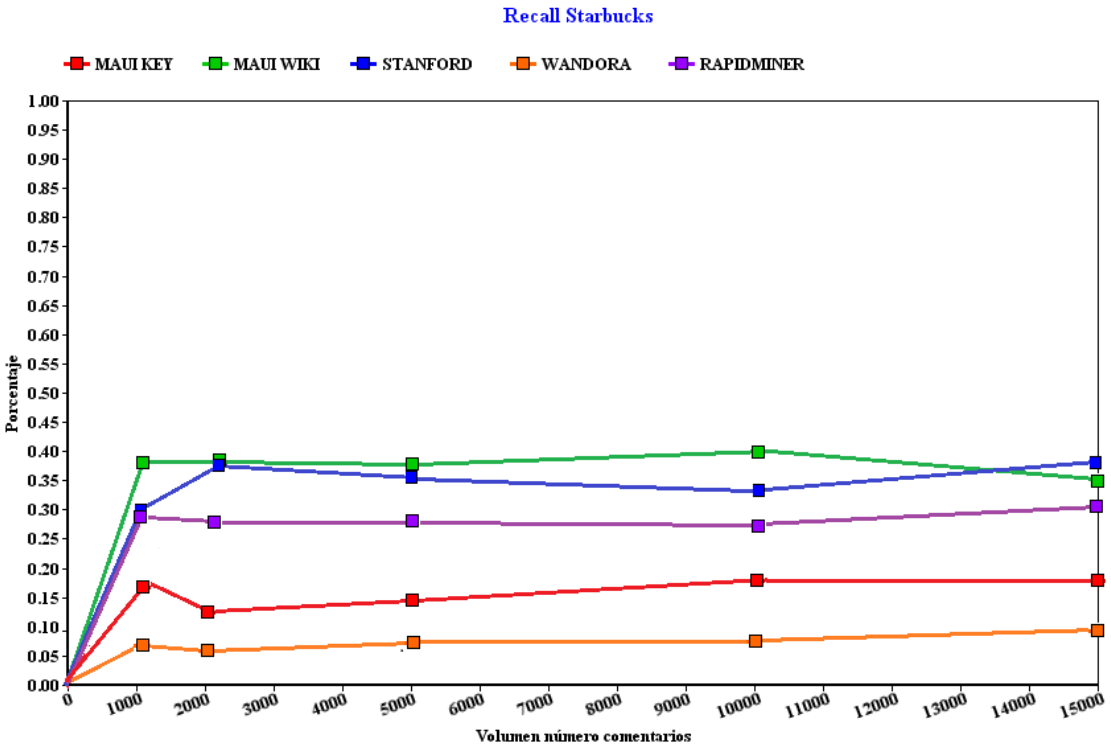


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,17 ± 0,04	0,29 ± 0,08	0,4 ± 0,13	0,04 ± 0,03	0,3 ± 0,044
Grupos 2000 comentarios	0,16 ± 0,057	0,27 ± 0,07	0,37 ± 0,15	0,04 ± 0,03	0,22 ± 0,06
Grupos 5000 comentarios	0,165 ± 0,066	0,3 ± 0,05	0,36 ± 0,05	0,04 ± 0,03	0,29 ± 0,011
Grupos 10000 comentarios	0,18 ± 0,079	0,3 ± 0,09	0,41 ± 0,04	0,04 ± 0,03	0,3 ± 0,06
Grupos 15000 comentarios	0,22 ± 0,06	0,35 ± 0,11	0,49 ± 0,1	0,08 ± 0,026	0,52 ± 0,08

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,01	x ± 0,022	x ± 0,035	x ± 0,009	x ± 0,012
Grupos 2000 comentarios	x ± 0,016	x ± 0,02	x ± 0,04	x ± 0,009	x ± 0,17
Grupos 5000 comentarios	x ± 0,018	x ± 0,015	x ± 0,013	x ± 0,009	x ± 0,003
Grupos 10000 comentarios	x ± 0,022	x ± 0,027	x ± 0,01	x ± 0,009	x ± 0,016
Grupos 15000 comentarios	x ± 0,018	x ± 0,03	x ± 0,027	x ± 0,007	x ± 0,023

Figura 27: Gráfico evolutivo de recall para PetRescueSaga

En comentarios de Starbucks Wandora tiene el peor recall pero mejor que para los juegos KING. Maui KEY tiene similar comportamiento que con los juegos KING.
RapidMiner se mantiene estable en todo el conjunto de volúmenes pero finalmente tiene peor recall que para CandyCrushSaga o PetRescueSaga, por lo que funciona mejor con los juegos que con los comentarios de Starbucks. Son Maui WIKI y Stanford los que han obtenido resultados similares que para los juegos con finalmente Stanford obteniendo el mejor resultado de recall, a mayor volumen de datos menos topics relevantes diferentes y mayor exhaustivo y concreto es el resultado de su algoritmo (Figura 28).

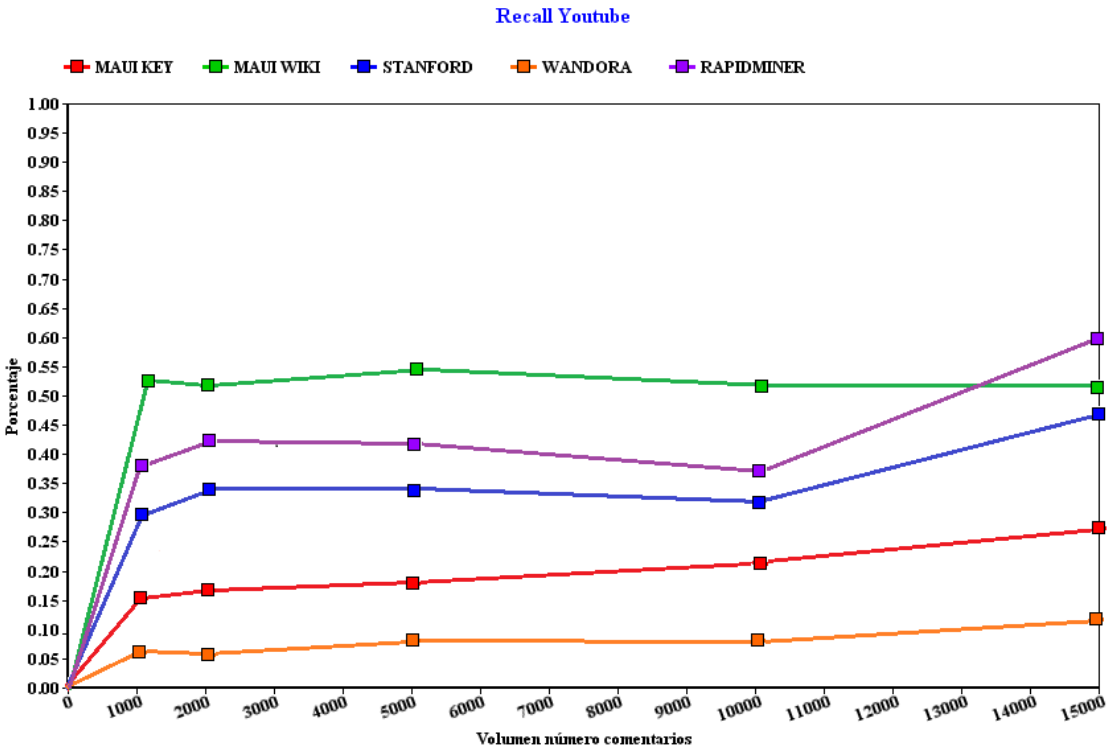


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,17 ± 0,08	0,38 ± 0,07	0,3 ± 0,14	0,07 ± 0,021	0,29 ± 0,08
Grupos 2000 comentarios	0,13 ± 0,09	0,38 ± 0,12	0,37 ± 0,17	0,06 ± 0,05	0,28 ± 0,08
Grupos 5000 comentarios	0,15 ±0,08	0,37 ± 0,12	0,35 ± 0,13	0,07 ± 0,027	0,28 ± 0,07
Grupos 10000 comentarios	0,18 ± 0,095	0,4 ± 0,088	0,33 ± 0,13	0,07 ± 0,022	0,27 ± 0,12
Grupos 15000 comentarios	0,18 ± 0,06	0,34 ± 0,083	0,39 ± 0,16	0,1 ± 0,055	0,3 ± 0,13

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,023	x ± 0,021	x ± 0,039	x ± 0,006	x ± 0,023
Grupos 2000 comentarios	x ± 0,024	x ± 0,035	x ± 0,047	x ± 0,013	x ± 0,023
Grupos 5000 comentarios	x ± 0,023	x ± 0,024	x ± 0,036	x ± 0,007	x ± 0,021
Grupos 10000 comentarios	x ± 0,026	x ± 0,023	x ± 0,036	x ± 0,006	x ± 0,035
Grupos 15000 comentarios	x ± 0,017	x ± 0,023	x ± 0,044	x ± 0,015	x ± 0,036

Figura 28: Gráfico evolutivo de recall para Starbucks

Para los comentarios de Youtube la herramienta Wandora sigue siendo la peor, indica que para cualquier tipo de comentarios se ha obtenido muchos topics relevantes diferentes. Todas las herramientas mejoran el recall a mayor volumen de datos. Maui KEY tiene un comportamiento similar al resto de clientes. Stanford ha obtenido resultados similares a los otros clientes de gráficas anteriores y RapidMiner ha vuelto a situarse como la herramienta con mejor recall. Maui WIKI parece que ha funcionado mejor con los comentarios de Youtube pero ha demostrado que a mayor volumen de datos su recall empeora (Figura 29).

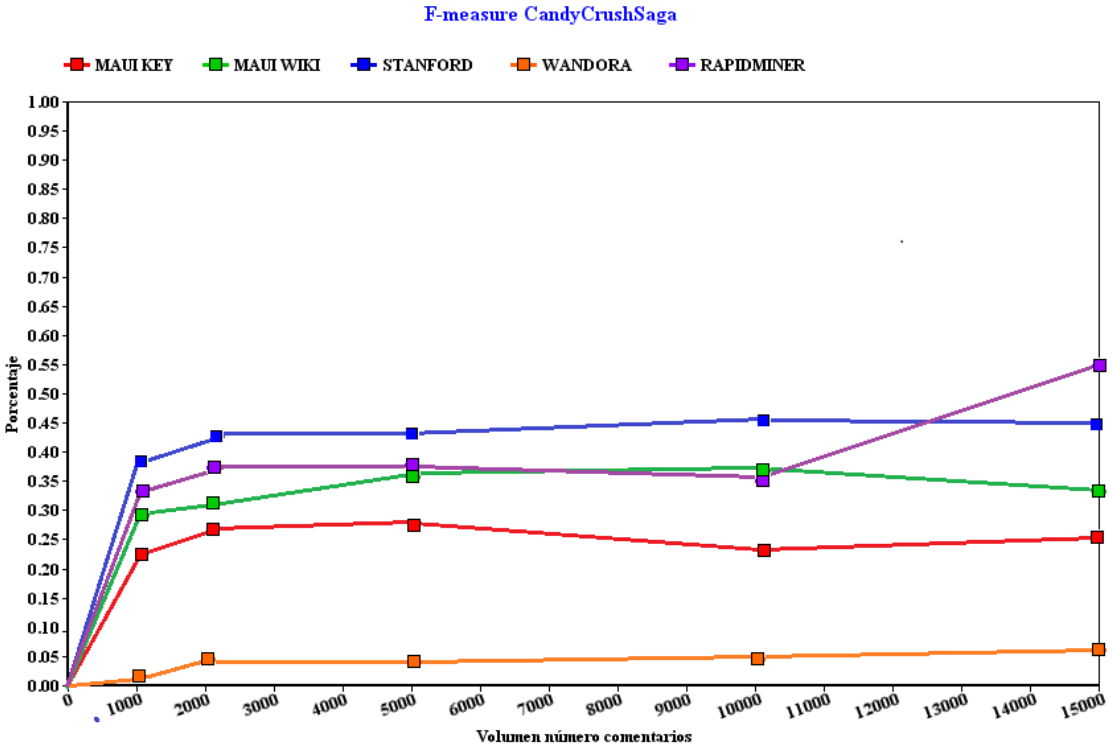


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,15 ± 0,073	0,53 ± 0,15	0,29 ± 0,06	0,06 ± 0,043	0,37 ± 0,09
Grupos 2000 comentarios	0,16 ± 0,07	0,52 ± 0,146	0,34 ± 0,01	0,057 ± 0,03	0,41 ± 0,068
Grupos 5000 comentarios	0,17 ± 0,07	0,54 ± 0,13	0,34 ± 0,087	0,067 ± 0,043	0,4 ± 0,06
Grupos 10000 comentarios	0,19 ± 0,08	0,52 ± 0,08	0,31 ± 0,1	0,06 ± 0,04	0,35 ± 0,1
Grupos 15000 comentarios	0,25 ± 0,069	0,52 ± 0,11	0,44 ± 0,097	0,1 ± 0,02	0,6 ± 0,095

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,02	x ± 0,04	x ± 0,017	x ± 0,012	x ± 0,052
Grupos 2000 comentarios	x ± 0,019	x ± 0,04	x ± 0,027	x ± 0,008	x ± 0,019
Grupos 5000 comentarios	x ± 0,023	x ± 0,036	x ± 0,024	x ± 0,012	x ± 0,017
Grupos 10000 comentarios	x ± 0,22	x ± 0,22	x ± 0,027	x ± 0,01	x ± 0,027
Grupos 15000 comentarios	x ± 0,019	x ± 0,03	x ± 0,027	x ± 0,006	x ± 0,026

Figura 29: Gráfico evolutivo de recall para Youtube

F-measure realiza una medida fusionando la precisión y el recall.
Para CandyCrushSaga se ha obtenido que Wandora es la peor herramienta, bastante lejos del resto de herramientas. Su precisión es de las más bajas y el recall bastante bajo, indicando un cuarto de topic relevantes por experimento y diferentes entre ellos sin exhaustividad
Maui KEY y Maui WIKI han obtenido comportamiento similar siendo Maui WIKI un poco mejor, tiene peor precisión, la mitad de Maui KEY, pero devuelve un número más bajo de topics relevantes diferentes, ya que Maui KEY tiene un recall mucho más bajo.
Stanford y RapidMiner se erigen como las mejores herramientas, altas precisiones del 50% y altas recall, siendo RapidMiner quien obtiene mejor resultado a mayor volumen de datos debido a que su recall es más efectivo en ese caso (Figura 30).

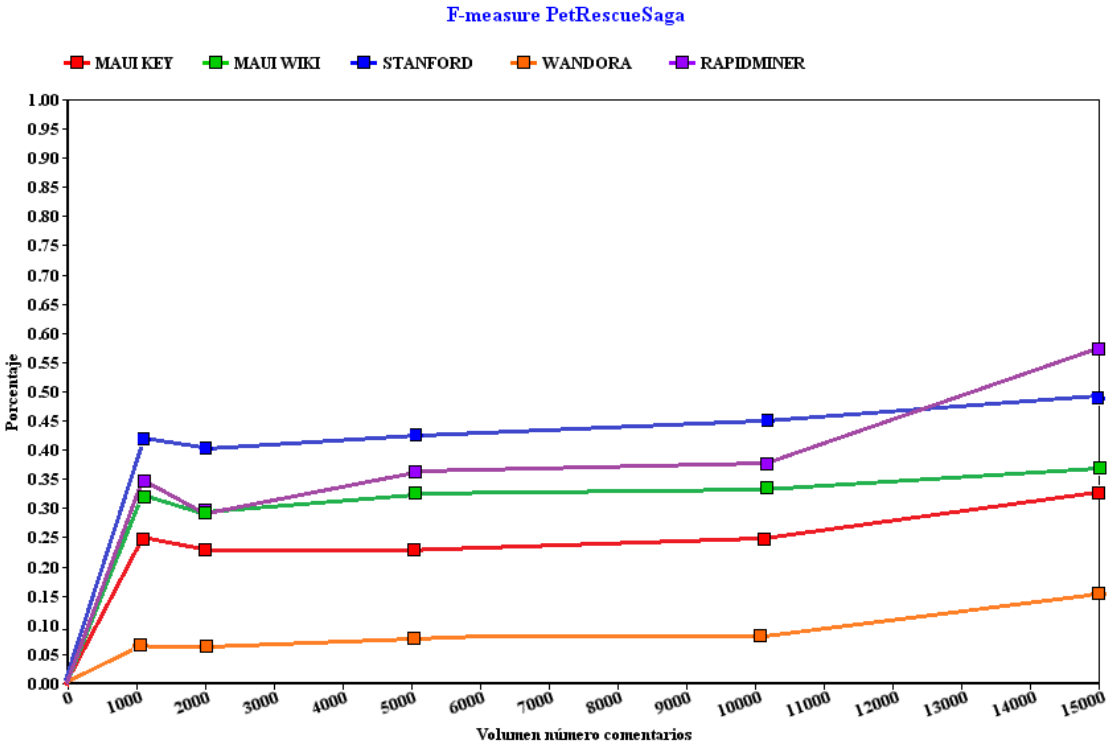


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,22 ± 0,088	0,297 ± 0,04	0,38 ± 0,08	0,019 ± 0,02	0,33 ± 0,11
Grupos 2000 comentarios	0,26 ± 0,116	0,31 ± 0,06	0,43 ± 0,1	0,045 ± 0,03	0,37 ± 011
Grupos 5000 comentarios	0,27 ± 0,01	0,36 ± 0,022	0,43 ± 0,1	0,0489 ±	0,369 ± 0,06
Grupos 10000 comentarios	0,23 ± 0,08	0,374 ± 0,054	0,45 ± 0,087	0,05 ± 0,01	0,346 ± 0,077
Grupos 15000 comentarios	0,25 ± 0,11	0,32 ± 0,067	0,44 ± 0,07	0,057 ± 0,034	0,568 ± 0,14

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,024	x ± 0,12	x ± 0,023	x ± 0,005	x ± 0,029
Grupos 2000 comentarios	x ± 0,32	x ± 0,017	x ± 0,028	x ± 0,009	x ± 0,032
Grupos 5000 comentarios	x ± 0,0027	x ± 0,006	x ± 0,028	x ± 0,003	x ± 0,017
Grupos 10000 comentarios	x ± 0,023	x ± 0,15	x ± 0,024	x ± 0,04	x ± 0,021
Grupos 15000 comentarios	x ± 0,03	x ± 0,018	x ± 0,019	x ± 0,009	x ± 0,039

Figura 30: Gráfico evolutivo de F-measure para CandyCrushSaga

El comportamiento de las herramientas con PetRescueSaga es muy similar a CandyCrushSaga debido a que los comentarios de los usuarios son parecidos y la temática basada en juegos y niveles del juego. Se aplican las mismas conclusiones que con CandyCrushSaga pero haciendo una pequeña diferencia, para el volumen de datos de 2000 comentarios todas las herramientas sufren un bajón para luego aumentar su efectividad, tanto en precisión como en recall, a mayor volumen de datos (Figura 31).



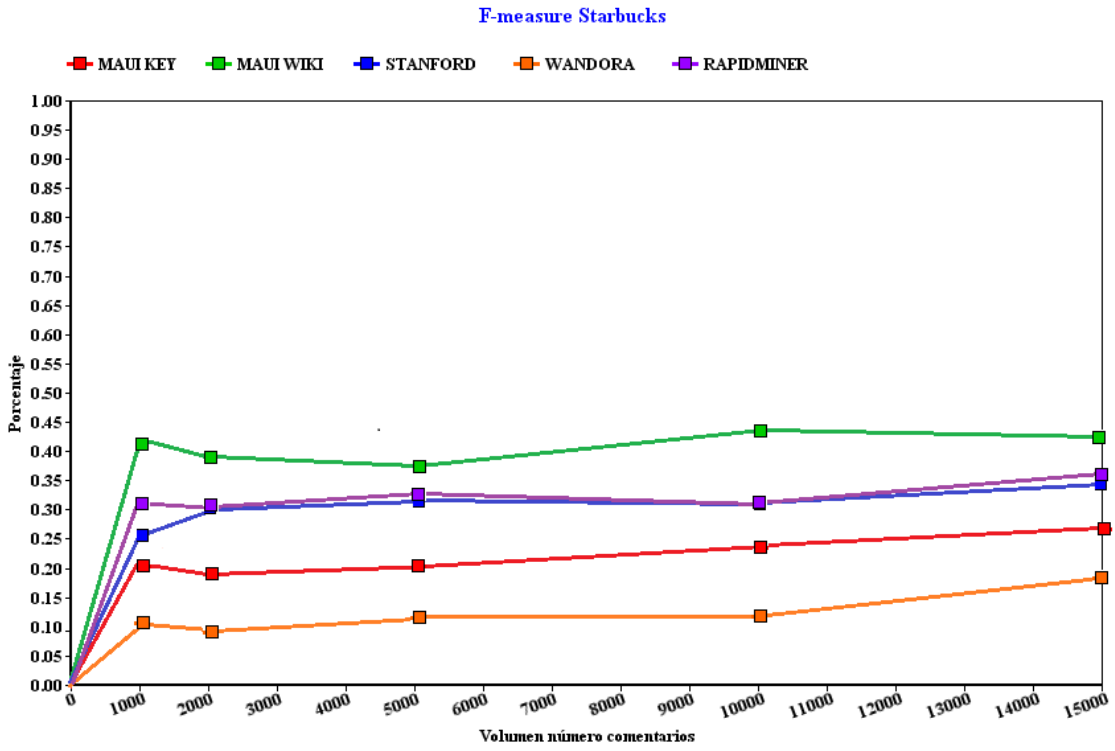
MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,25 ± 0,086	0,32 ± 0,03	0,42 ± 0,09	0,06 ± 0,01	0,35 ± 0,11
Grupos 2000 comentarios	0,23 ± 0,09	0,29 ± 0,07	0,41 ± 0,11	0,06 ± 0,05	0,29 ± 0,12
Grupos 5000 comentarios	0,23 ± 0,084	0,328 ± 0,21	0,42 ± 0,11	0,069 ± 0,028	0,36 ± 0,066
Grupos 10000 comentarios	0,25 ± 0,083	0,33 ± 0,12	0,45 ± 0,09	0,07 ± 0,019	0,386 ± 0,08
Grupos 15000 comentarios	0,33 ± 0,097	0,36 ± 0,09	0,49 ± 0,05	0,14 ± 0,05	0,57 ± 0,075

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,024	x ± 0,009	x ± 0,027	x ± 0,004	x ± 0,03
Grupos 2000 comentarios	x ± 0,025	x ± 0,02	x ± 0,03	x ± 0,015	x ± 0,035
Grupos 5000 comentarios	x ± 0,023	x ± 0,05	x ± 0,03	x ± 0,008	x ± 0,18
Grupos 10000 comentarios	x ± 0,023	x ± 0,033	x ± 0,027	x ± 0,005	x ± 0,023
Grupos 15000 comentarios	x ± 0,027	x ± 0,027	x ± 0,015	x ± 0,015	x ± 0,021

Figura 31: Gráfico evolutivo de F-measure para PetRescueSaga

Se ha comprobado que Stanford y RapidMiner se erigen como la mejor opción para el proceso de comentarios de usuarios basados en los juegos de la empresa KING y mejor a un volumen alto de comentarios en la entrada.

Para comentarios de Starbucks, Wandora es la peor herramienta pero ha obtenido mejores resultados que con los juegos, debido a que su precisión ha sido más alta.
Todas las herramientas mejoran con un mayor volumen de datos. Maui KEY, Stanford y RapidMiner tienen un comportamiento similar como con los comentarios de los juegos.
Pero para este tipo de comentarios Maui WIKI ha sido la mejor herramienta debido a una precisión alta y un recall alto (Figura 32).



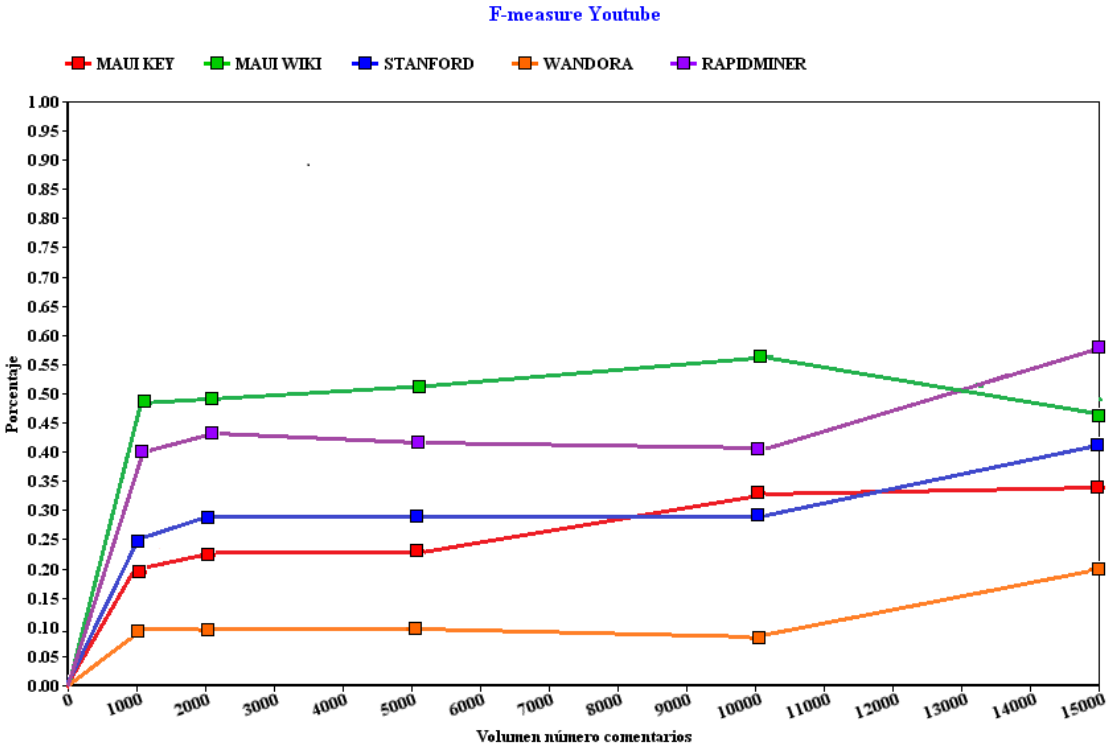
MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,21 ± 0,11	0,41 ± 0,11	0,25 ± 0,1	0,12 ± 0,024	0,31 ± 0,06
Grupos 2000 comentarios	0,18 ± 0,1	0,38 ± 0,12	0,3 ± 0,1	0,1 ± 0,07	0,3 ± 0,1
Grupos 5000 comentarios	0,2 ± 0,07	0,37 ± 0,12	0,31 ± 0,15	0,12 ± 0,024	0,32 ± 0,09
Grupos 10000 comentarios	0,24 ± 0,1	0,44 ± 0,15	0,3 ± 0,1	0,12 ± 0,028	0,3 ± 0,09
Grupos 15000 comentarios	0,26 ± 0,07	0,43 ± 0,07	0,34 ± 0,15	0,17 ± 0,01	0,35 ± 0,14

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,03	x ± 0,03	x ± 0,028	x ± 0,006	x ± 0,017
Grupos 2000 comentarios	x ± 0,028	x ± 0,035	x ± 0,028	x ± 0,019	x ± 0,028
Grupos 5000 comentarios	x ± 0,021	x ± 0,035	x ± 0,042	x ± 0,006	x ± 0,024
Grupos 10000 comentarios	x ± 0,028	x ± 0,042	x ± 0,028	x ± 0,008	x ± 0,035
Grupos 15000 comentarios	x ± 0,021	x ± 0,019	x ± 0,042	x ± 0,027	x ± 0,038

Figura 32: Gráfico evolutivo de F-measure para Starbucks

Stanford y RapidMiner han conseguido grandes resultados, pero para comentarios de Starbucks, es decir, comentarios dedicados a una marca o logo, el algoritmo Maui con indexación a Wikipedia es mejor opción.

Para comentarios de Youtube, Wandora obtiene los peores resultados, pero al igual que con comentarios de Starbucks son mejores debido a una alta precisión. Maui KEY tiene un comportamiento similar a Starbucks por debajo de Maui WIKI quien tiene buenos resultados debidos a una recall y precisión altas. Stanford debido a su recall alta pero baja precisión tiene unos resultados decentes entre medias de las herramientas.
Es RapidMiner quien ha obtenido mejores resultados, sobre todo para volumen de datos mayor, siendo superado por Maui WIKI para volúmenes bajos de datos (Figura 33).



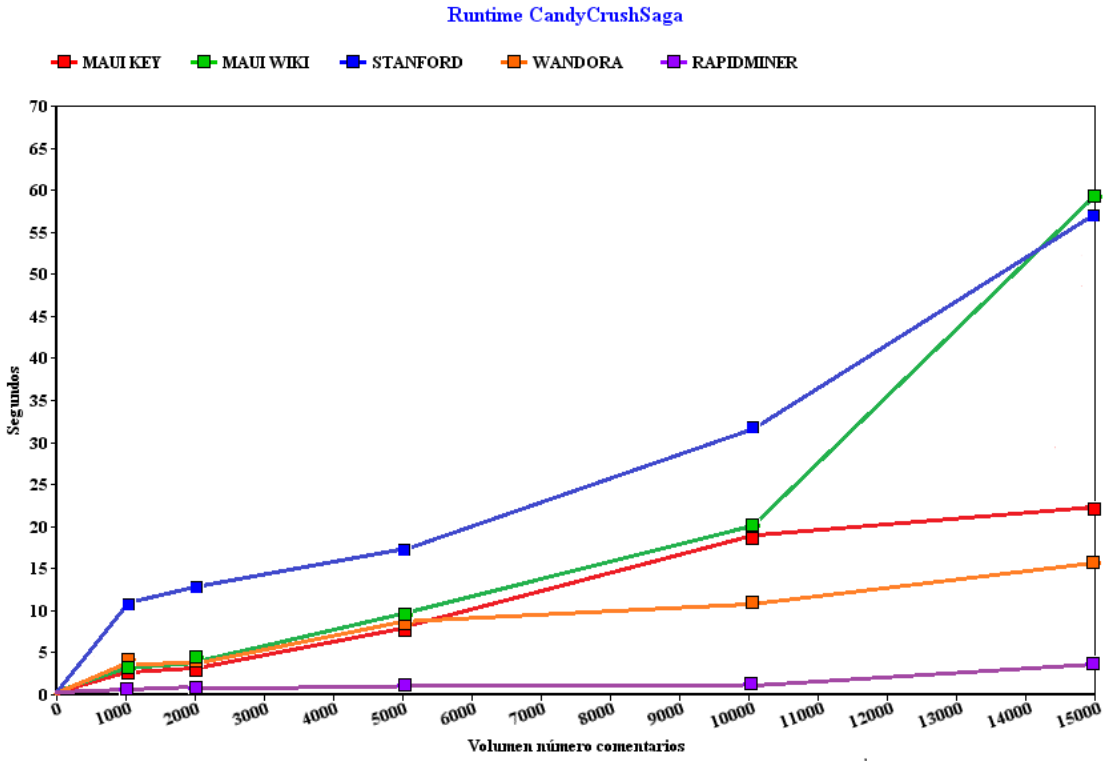
MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	0,19 ± 0,076	0,47 ± 0,12	0,24 ± 0,1	0,1 ± 0,07	0,4 ± 0,12
Grupos 2000 comentarios	0,23 ± 0,1	0,48 ± 0,12	0,28 ± 0,1	0,1 ± 0,07	0,43 ± 0,11
Grupos 5000 comentarios	0,23 ± 0,11	0,51 ± 0,14	0,28 ± 0,09	0,11 ± 0,03	0,42 ± 0,09
Grupos 10000 comentarios	0,326 ± 0,1	0,54 ± 0,135	0,28 ± 0,08	0,09 ± 0,03	0,4 ± 0,12
Grupos 15000 comentarios	0,33 ± 0,12	0,45 ± 0,13	0,39 ± 0,07	0,17 ± 0,03	0,56 ± 0,04

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,021	x ± 0,032	x ± 0,027	x ± 0,02	x ± 0,032
Grupos 2000 comentarios	x ± 0,027	x ± 0,032	x ± 0,027	x ± 0,02	x ± 0,032
Grupos 5000 comentarios	x ± 0,03	x ± 0,038	x ± 0,025	x ± 0,008	x ± 0,025
Grupos 10000 comentarios	x ± 0,027	x ± 0,037	x ± 0,22	x ± 0,008	x ± 0,032
Grupos 15000 comentarios	x ± 0,032	x ± 0,036	x ± 0,22	x ± 0,008	x ± 0,12

Figura 33: Gráfico evolutivo de F-measure para Youtube

A continuación se muestra los gráficos de evolución para el criterio cuantitativo de tiempo de ejecución de los experimentos, y una tabla con el volumen máximo soportado en la entrada de cada herramienta o algoritmo.

Para comentarios de CandyCrushSaga el tiempo de ejecución en todas las herramientas es mayor a mayor volumen, tardan más en terminar de procesar el texto a más volumen de datos en la entrada. El tiempo de ejecución de RapidMiner es muy bajo, rápidamente la herramienta procesa el texto devolviendo los resultados.
Wandora es la siguiente herramienta en procesar más rápidamente el texto, y apenas altera su tiempo a mayor volumen de datos en la entrada. Maui KEY y Maui WIKI tardan similar al ser el mismo algoritmo en procesar el texto, sin embargo Maui WIKI dispara su tiempo de ejecución a sobrepasar los 1000 comentarios por experimento.
Wandora en volumen pequeño de comentarios obtiene un tiempo de ejecución mayor que el algoritmo de Maui para las dos configuraciones.
Stanford es la herramienta que más tarda en procesar el texto, quizás porque su ejecución se divide en scripts para completar todo el desarrollo y porque procesa el texto desde un archivo csv y no desde un archivo plano o desde la entrada directa de la herramienta. Su tiempo de ejecución aumenta de manera casi lineal al volumen de datos (Figura 34).

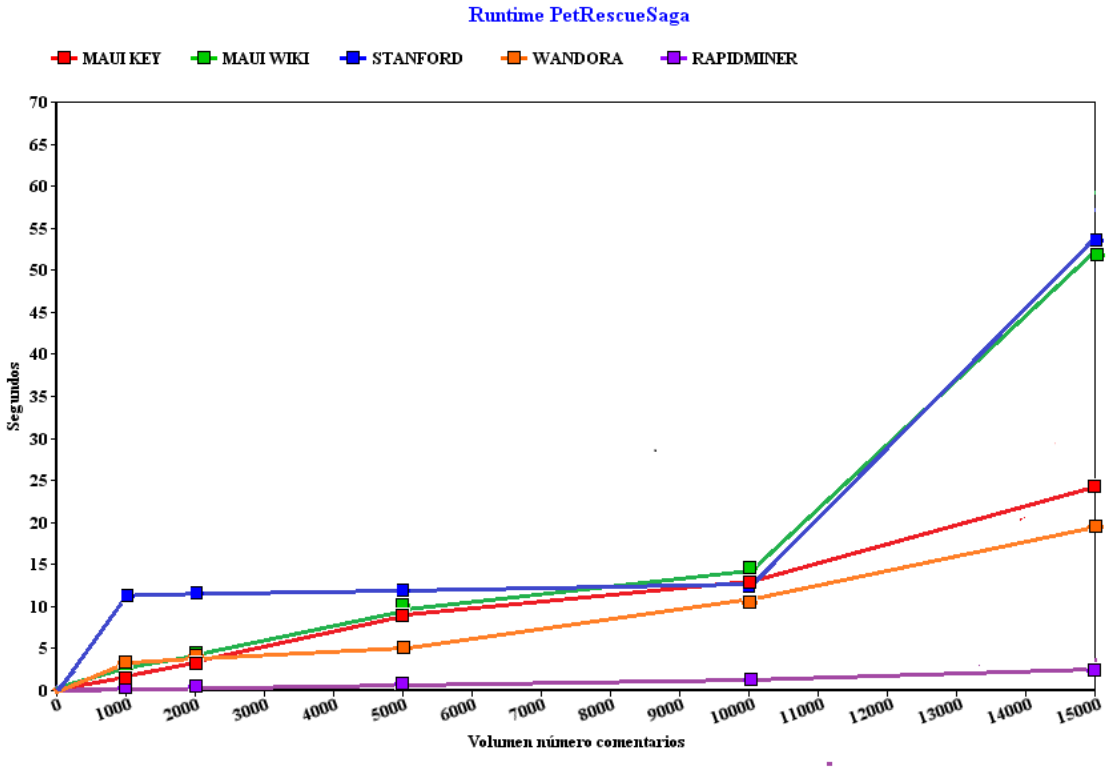


MEDIA + DESVIACION	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	2,252 ± 0,79	2,58 ± 0,83	11,46 ± 1,92	2,66 ± 0,28	0,2 ± 0,001
Grupos 2000 comentarios	3,452 ± 0,8	4,05 ± 1	13,12 ± 1,22	3,55 ± 0,6	0,5 ± 0,001
Grupos 5000 comentarios	8,75 ± 2,1	9,42 ± 2,28	17,12 ± 1,23	6,14 ± 1,06	0,9 ± 0,001
Grupos 10000 comentarios	18,9 ± 2,71	20,31 ± 4,98	32,52 ± 1,25	10,67 ± 1,55	2,06 ± 0,31
Grupos 15000 comentarios	22,98 ± 7,01	58,38 ± 7,51	52,52 ± 1,25	16,02 ± 1,62	4,01 ± 0,48

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,22	x ± 0,23	x ± 0,53	x ± 0,077	x ± 0,00027
Grupos 2000 comentarios	x ± 0,22	x ± 0,27	x ± 0,33	x ± 0,16	x ± 0,00027
Grupos 5000 comentarios	x ± 0,58	x ± 0,63	x ± 0,34	x ± 0,29	x ± 0,00027
Grupos 10000 comentarios	x ± 0,75	x ± 1,38	x ± 0,346	x ± 0,42	x ± 0,086
Grupos 15000 comentarios	x ± 1,94	x ± 2,08	x ± 0,346	x ± 0,45	x ± 0,13

Figura 34: Gráfico evolutivo de Runtime para CandyCrushSaga

Para comentarios de PetRescueSaga las herramientas tienen un comportamiento similar, en principio y a diferencia de los criterios cualitativos antes evaluados, el tiempo de ejecución no debe variar mucho de un cliente a otro. RapidMiner ha obtenido tiempos de ejecución muy bajos.
Wandora es la segunda herramienta con menores tiempos de ejecución y muy similares a CandyCrushSaga, al igual que Maui KEY y Maui WIKI, este último algoritmo sufre la misma exponencialidad a mayor número de comentarios en la entrada.
Es Stanford quien tiene un comportamiento diferente a CandyCrushSaga, ha obtenido unos tiempos de ejecución sin apenas variantes durante gran parte de los experimentos, pero finalmente a mayor volumen también ha obtenido un tiempo de ejecución elevado, pero dando la impresión de exponencialidad al igual que Maui WIKI, quizás debido a que los comentarios de PetRescueSaga contienen menos palabras durante el proceso (Figura 35).

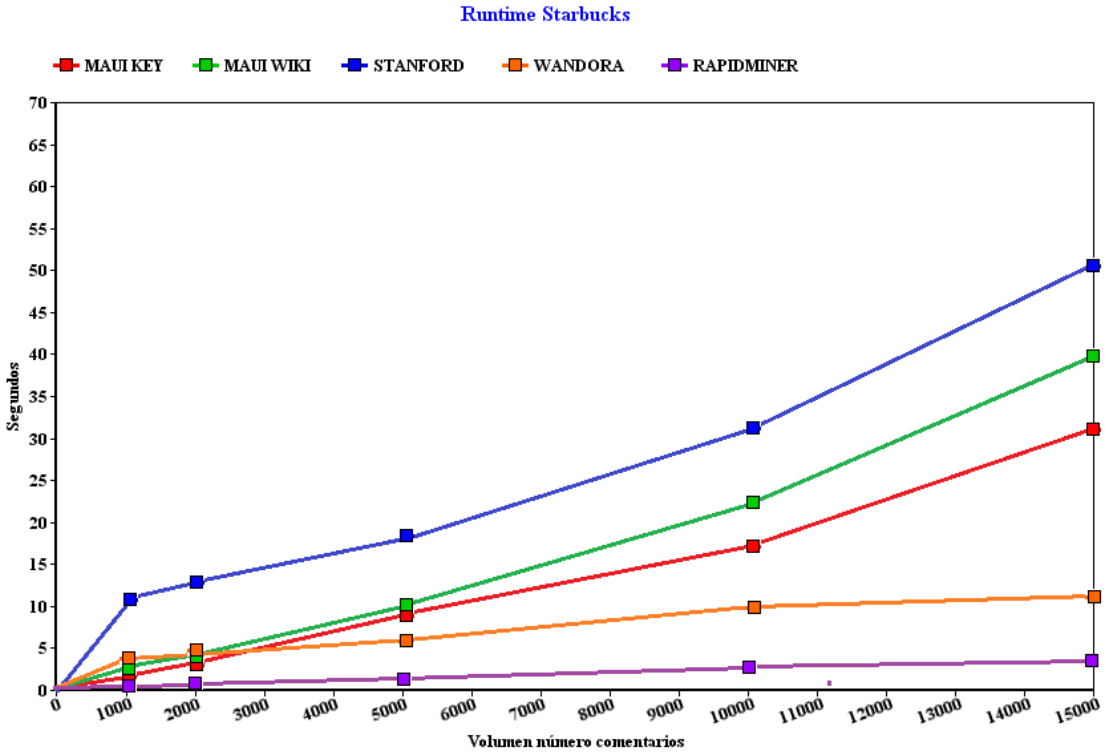


MEDIA + DESVIACION	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	1,56 ± 0,99	2,26 ± 2,14	11,12 ± 0,32	2,85 ± 0,73	0,2 ± 0,001
Grupos 2000 comentarios	3,46 ± 0,87	4,1 ± 1,17	11,36 ± 0,62	3,56 ± 0,68	0,2 ± 0,001
Grupos 5000 comentarios	8,58 ± 1,83	8,64 ± 5,75	11,88 ± 1,36	5,34 ± 1,92	0,6 ± 0,001
Grupos 10000 comentarios	13,39 ± 0,88	14,1 ± 0,894	13 ± 0,89	11,7 ± 1,92	1 ± 0,001
Grupos 15000 comentarios	24,12 ± 6,4	51,52 ± 5,88	52,92 ± 0,62	18,73 ± 1,7	1,952 ± 0,13

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,27	x ± 0,6	x ± 0,088	x ± 0,2	x ± 0,000277
Grupos 2000 comentarios	x ± 0,24	x ± 0,32	x ± 0,17	x ± 0,188	x ± 0,000277
Grupos 5000 comentarios	x ± 0,5	x ± 1,59	x ± 0,37	x ± 0,53	x ± 0,000277
Grupos 10000 comentarios	x ± 0,24	x ± 0,247	x ± 0,24	x ± 0,53	x ± 0,000277
Grupos 15000 comentarios	x ± 1,77	x ± 1,63	x ± 0,17	x ± 0,47	x ± 0,036

Figura 35: Gráfico evolutivo de Runtime para PetRescueSaga

Para comentarios de Starbucks las herramientas RapidMiner y Wandora tienen el mismo comportamiento que para los comentarios de los juegos. Maui WIKI y Maui KEY tienen un espectro parecido, siendo de nuevo Maui WIKI quien tiene mayor tiempo de ejecución. Maui WIKI tiene un comportamiento más lineal, debido a la menor cantidad de palabras disponibles en los comentarios de Starbucks.
En Stanford ocurre parecido, es la herramienta que más ha tardado en devolver los resultados y su comportamiento es lineal aumentando el tiempo de ejecución a mayor volumen de entrada. Parece no influir la cantidad de palabras en comentarios porque tiene resultados similares a CandyCrushSaga y PetRescueSaga (Figura 36).

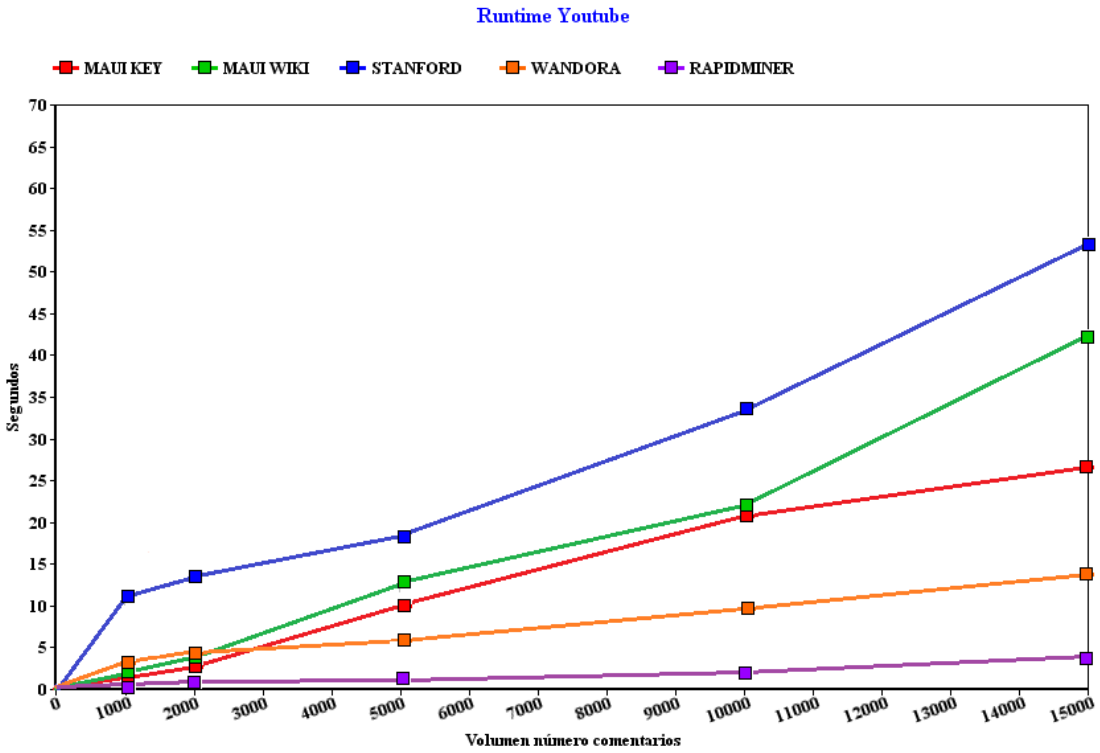


MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	1,88 ± 0,43	2,33 ± 0,28	11,6 ± 1,02	2,55 ± 0,35	0,02 ± 0,0001
Grupos 2000 comentarios	2,91 ± 0,27	3,5 ± 0,4	13,33 ± 0,76	3,54 ± 0,21	0,4 ± 0,0001
Grupos 5000 comentarios	8,42 ± 0,66	9,03 ± 0,43	17,53 ± 0,79	5,76 ± 0,46	1 ± 0,11
Grupos 10000 comentarios	16,48 ± 3,77	24,9 ± 0,16	32,06 ± 1,81	8,37 ± 0,38	3 ± 0,55
Grupos 15000 comentarios	32 ± 3,09	40,7 ± 0,67	51,6 ± 0,51	11,42 ± 0,47	3,5 ± 0,7

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,12	x ± 0,07	x ± 0,28	x ± 0,09	x ± 0,00002
Grupos 2000 comentarios	x ± 0,07	x ± 0,11	x ± 0,21	x ± 0,06	x ± 0,00002
Grupos 5000 comentarios	x ± 0,18	x ± 0,12	x ± 0,22	x ± 0,13	x ± 0,03
Grupos 10000 comentarios	x ± 1,04	x ± 0,16	x ± 0,5	x ± 0,105	x ± 0,15
Grupos 15000 comentarios	x ± 0,85	x ± 0,67	x ± 0,51	x ± 0,13	x ± 0,19

Figura 36: Gráfico evolutivo de Runtime para Starbucks

Para los comentarios de Youtube no hay grandes variaciones a los tiempos de otros clientes. RapidMiner ha seguido con tiempos muy cortos seguido por Wandora. Maui KEY y Maui WIKI con un comportamiento lineal al igual que los comentarios de Starbucks, y con unos tiempos similares, debido a que Starbucks y Youtube tienen un número de palabras similar en el conjunto global de todos los comentarios. Maui WIKI sufre una exponencialidad a mayor número de comentarios en la entrada.
Y Stanford obtiene el mismo tiempo de ejecución que el resto de clientes por lo que no parece afectar el número de palabras por comentario, pero aumenta de manera lineal por el volumen de comentarios a la entrada (Figura 37).



MEDIA + DESVIACIÓN	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	1,63 ± 0,43	1,88 ± 0,38	11,55 ± 1,21	2,61 ± 0,29	0,12 ± 0,02
Grupos 2000 comentarios	2,44 ± 0,38	3,2 ± 0,31	13,61 ± 0,61	3,35 ± 0,39	0,55 ± 0,02
Grupos 5000 comentarios	9,63 ± 0,48	11,43 ± 1,17	17,61 ± 0,62	4,93 ± 0,54	1,51 ± 0,11
Grupos 10000 comentarios	19,69 ± 1,26	21,15 ± 1,61	32,5 ± 1,67	8,46 ± 0,81	1,97 ± 0,16
Grupos 15000 comentarios	25,5 ± 4,08	41,6 ± 3,01	52,5 ± 1,67	11,94 ± 1,17	3,55 ± 0,25

INTERVALO CONFIANZA	MAUI KEY	MAUI WIKI	STANFORD	WANDORA	RAPIDMINER
Grupos 1000 comentarios	x ± 0,12	x ± 0,105	x ± 0,33	x ± 0,08	x ± 0,005
Grupos 2000 comentarios	x ± 0,105	x ± 0,086	x ± 0,17	x ± 0,11	x ± 0,005
Grupos 5000 comentarios	x ± 0,13	x ± 0,32	x ± 0,17	x ± 0,15	x ± 0,03
Grupos 10000 comentarios	x ± 0,35	x ± 0,45	x ± 0,46	x ± 0,22	x ± 0,04
Grupos 15000 comentarios	x ± 1,13	x ± 0,83	x ± 0,46	x ± 0,32	x ± 0,07

Figura 37: Gráfico evolutivo de Runtime para Youtube

La siguiente tabla muestra el valor máximo aproximado de palabras que puede evaluar cada herramienta sin que aparezca un mensaje de error por falta de memoria o directamente la herramienta deje de funcionar. Tener en cuenta que los experimentos y la ejecución de las herramientas se han realizado en un ordenador con las siguientes características:

Procesador Intel Pentium Dual 1.86 GHz
Memoria RAM 2 GB
Sistema Operativo 32 bits Windows 7 Home Premium

Y que posiblemente para ordenadores mucho más potentes las herramientas permitan realizar procesos con mayor volumen de datos, pero nos sirve para realizar posibles cálculos de que herramienta permite más y cuanto podría alcanzar a soporte con un equipo mejor equipado.

Herramienta o Algoritmo	Num max palabras aprox
Maui	185990
Stanford	6262746
Wandora	443793
RapidMiner	1101809

Stanford es la única herramienta que ha sido capaz de procesar todo el conjunto global de CandyCrushSaga, que es el cliente con mayor cantidad de palabras, y devolver un resultado sin error, aunque el tiempo de ejecución fue extremadamente largo.

Maui es el algoritmo más restrictivo a la hora de volumen de datos de entrada, fue el algoritmo que ha condicionado el resto de herramientas y que ha condicionado la manera de agrupar los comentarios y realizar los experimentos por bloques para realizar las evaluaciones.

RapidMiner puede procesar una cantidad muy grande de texto, el tiempo de ejecución es mucho mayor que los pocos segundos que ha tardado en los experimentos durante el proyecto, pero consigue devolver resultados sin error.

Wandora se ha quedado en tercera posición en volumen de datos en la entrada, ha podido procesar el doble que el algoritmo de Maui pero menos de la mitad que RapidMiner, quizás sea debido a que necesita conexión a internet y este condicionado a la velocidad de transmisión de los datos.

6.7. Extra Resultados (empresa KING)

Como último apartado extra de las evaluaciones se han obtenido resultados utilizando las herramientas usando los comentarios de la propia empresa KING distribuidora de los juegos analizados CandyCrushSaga y PetRescueSaga además de otros juegos. Estos comentarios son oficiales de la propia marca diferenciándose de los comentarios analizados hasta ahora de usuarios independientes a la marca. Y se han obtenido unas pequeñas conclusiones.

Marca empresa KING		
Cliente	**Num Palabras**	**Num Comentarios**
BubbleWitchSaga	1813	62
CandyCrushSaga	7841	310
DiamondDiggerSaga	2052	87
FarmHeroesSaga	7268	320
PappaPearSaga	10269	428
PepperPanicSaga	4673	210
PetRescueSaga	9982	384
PyramidSolitaireSaga	2999	118

En los comentarios de la empresa KING el nombre de la marca de la empresa ‘king’ aparece muchas veces como el topic más relevante y la palabra ‘play’ la sigue en importancia además de los nombres de cada uno de los juegos evaluados.

Al ser el volumen de datos de la empresa KING muy pequeño el tiempo de ejecución es muy corto para todas las herramientas por lo que no se puede realizar una valoración efectiva en este punto y obteniendo el resultado de un único experimento ya que el volumen es tan poco, que no es coherente separar en grupos, solamente podemos calcular la precisión porque no tiene sentido el cálculo de recall en un solo experimento.

De esta manera se han obtenido las gráficas que se muestran a continuación en formato de barras para ver la precisión de cada herramienta y hacer una rápida y mejor comparación entre todos los juegos de la marca KING.

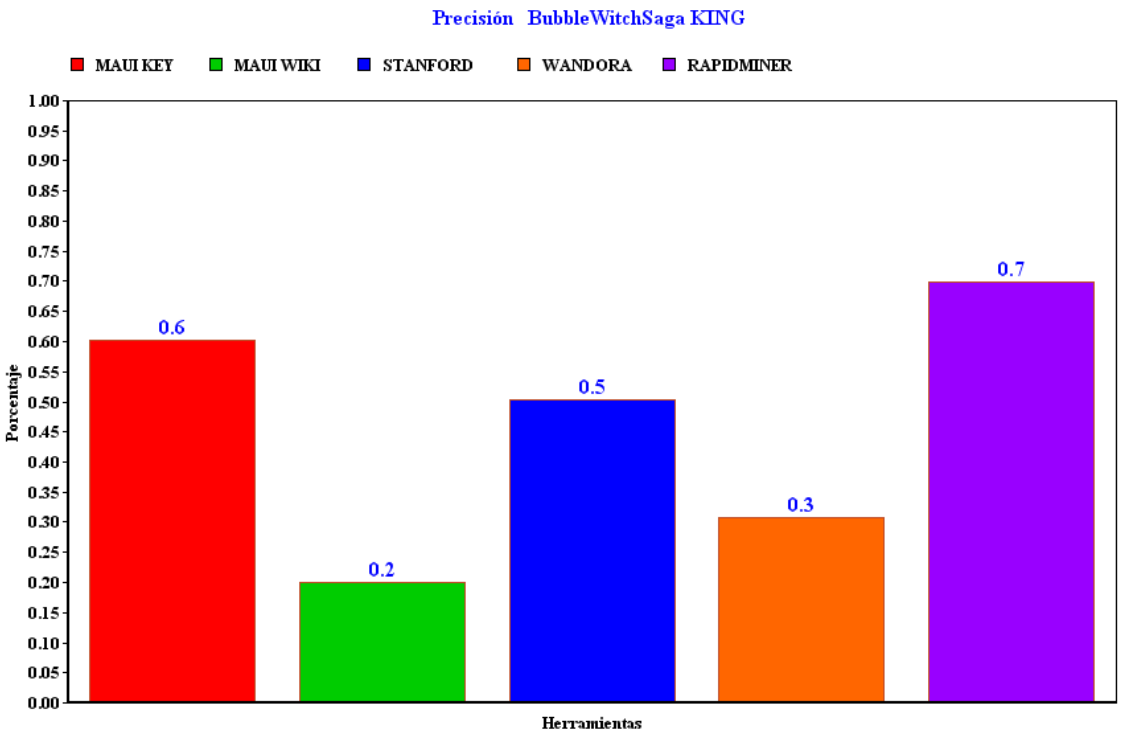


Figura 38: Gráfico de barras precisión para BubbleWitchSaga KING

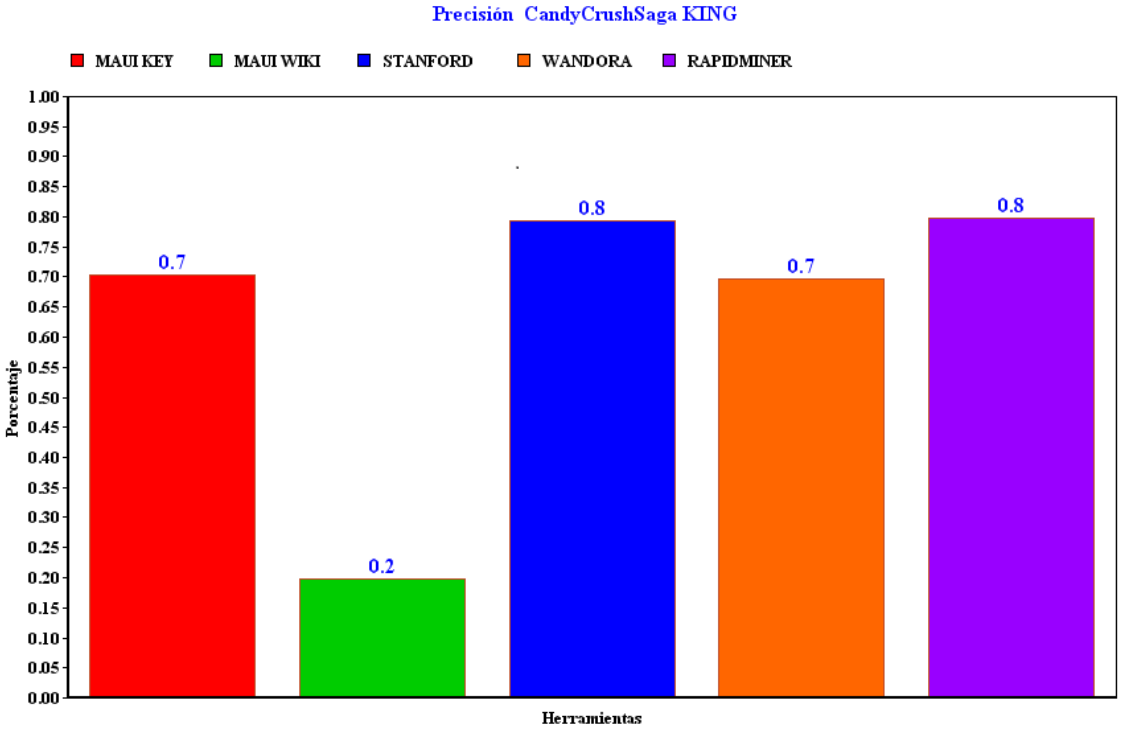


Figura 39: Gráfico de barras precisión para CandyCrushSaga KING

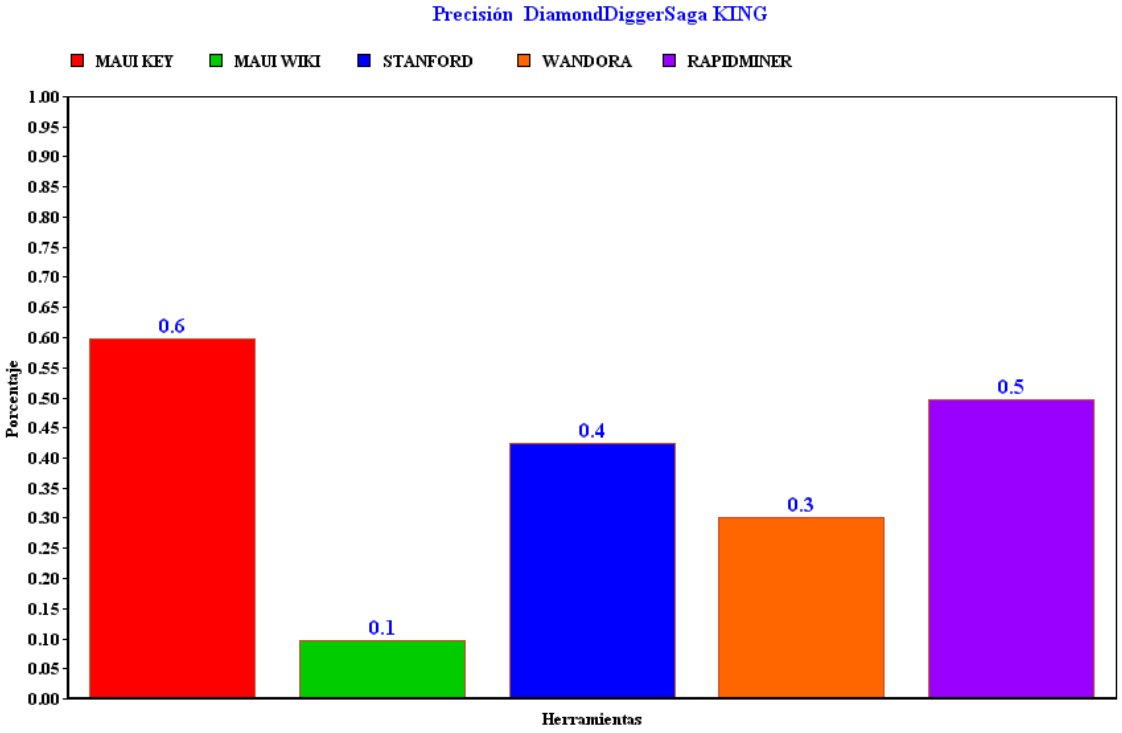


Figura 40: Gráfico de barras precisión para DiamondDiggerSaga KING

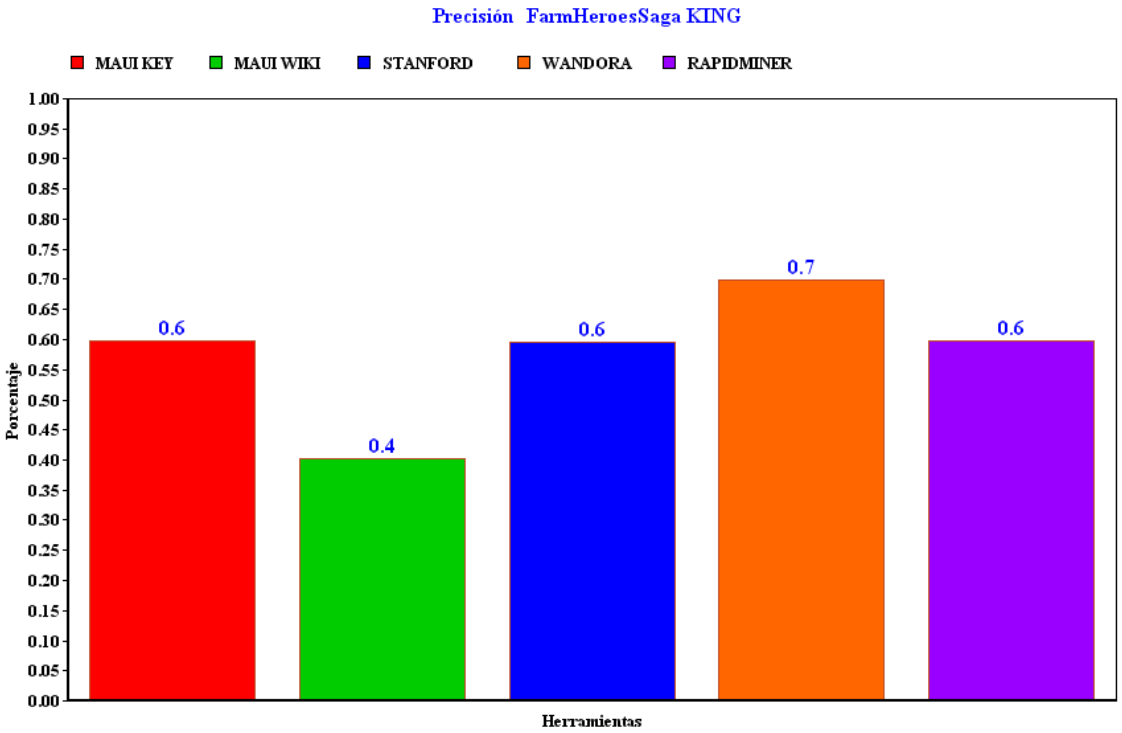


Figura 41: Gráfico de barras precisión para FarmHeroesSaga KING

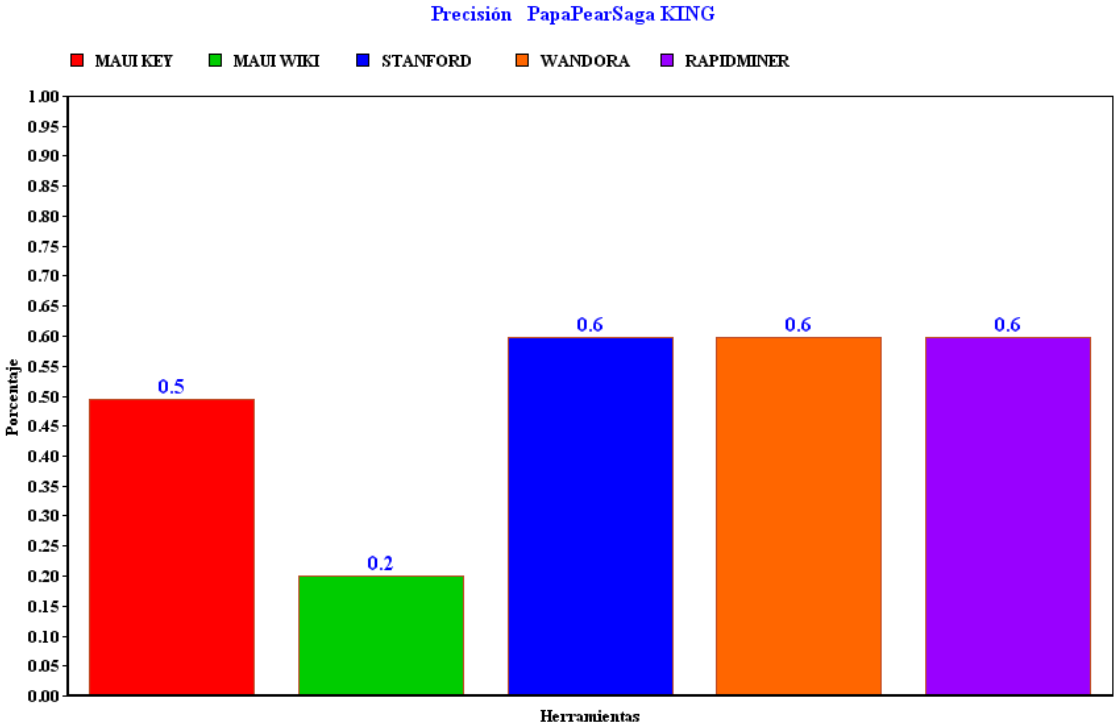


Figura 42: Gráfico de barras precisión para PapaPearSaga KING

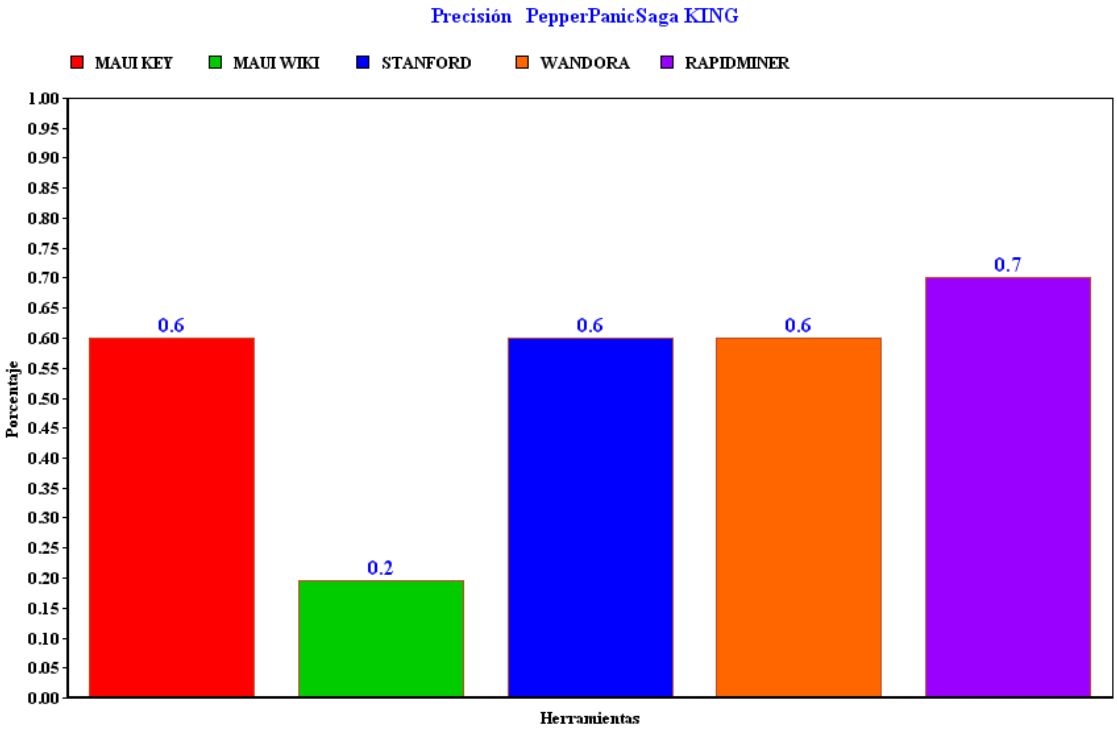


Figura 43: Gráfico de barras precisión para PepperPanicSaga KING

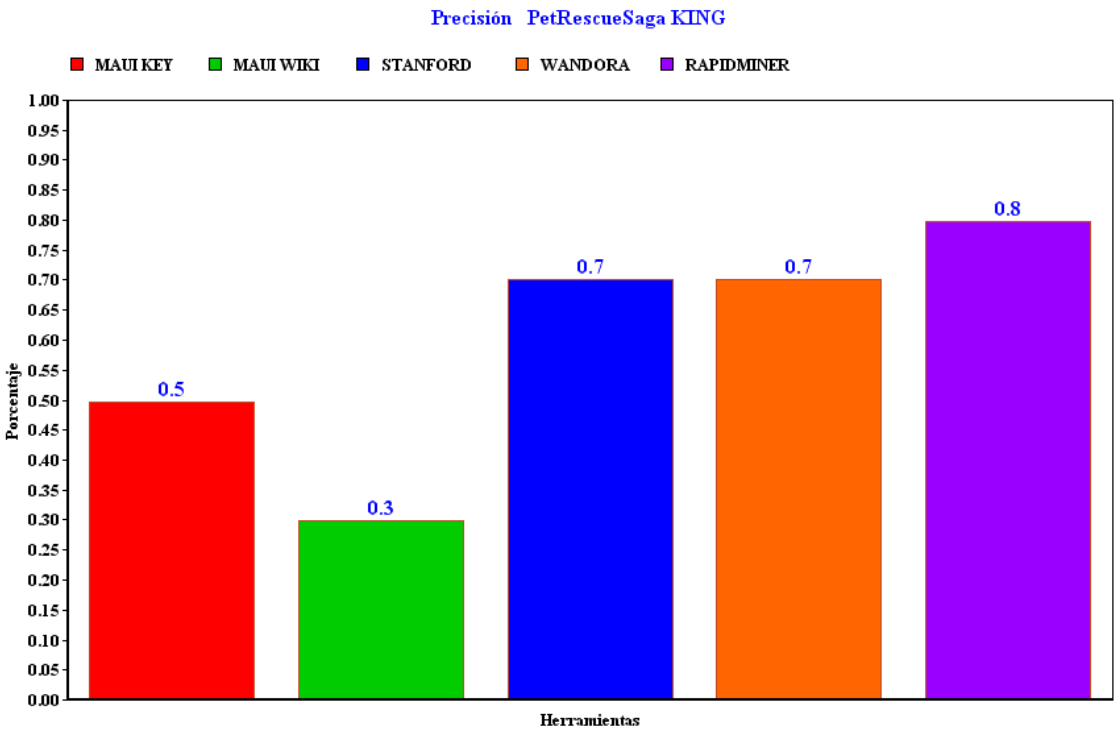


Figura 44: Gráfico de barras precisión para PetRescueSaga KING

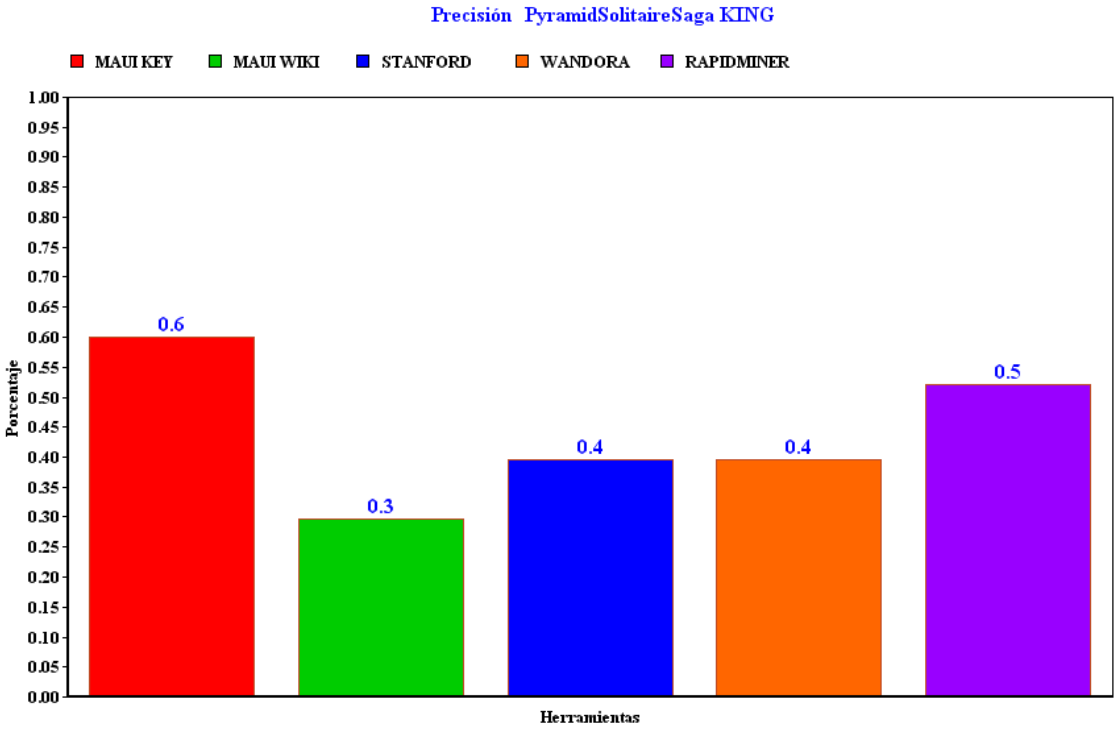


Figura 45: Gráfico de barras precisión para PyramidSolitarieSaga KING

La precisión con Maui KEY se ha encontrado entre el 50% y 70% mientras con Maui WIKI solamente entre 10% y 40%, por lo que esta opción no es recomendable para este tipo de textos que abarcan noticias o novedades por parte de una empresa.

Con Stanford la precisión ha oscilado entre 40% y 80%.
Con Wandora la precisión entre 30% y 70%, alcanzando precisiones más altas que la media general de los comentarios de los usuarios.

RapidMiner ha obtenido precisión entre 50% y 80%, en general ha obtenido mejores precisiones que el resto de herramientas.

Generalmente Stanford y RapidMiner han obtenido las mejores precisiones. Maui KEY y Maui WIKI con precisiones estables. Y por último Wandora que ha obtenido precisiones muy dispares para cada tipo de juego.

Finalmente ha parecido que las herramientas funcionan mejor con comentarios oficiales que por comentarios de usuarios, quizás porque los comentarios de los usuarios son más caóticos y muchas veces mal escritos o con acortaciones en las palabras. Y los comentarios de la propia marca utilizan en sus aportes las mismas palabras clave una y otra vez como parte de la publicidad hacia su marca y servicios.

Capítulo 7

Conclusiones y Líneas futuras

7.1. Conclusión final

La principal conclusión es que todas las herramientas, en más o menos, aumentan su precisión a mayor volumen de datos. Con esta condición Stanford será la mejor opción ya que puede procesar una gran cantidad de volumen con un solo experimento, mientras que el algoritmo de Maui, al contrario, solamente permite un volumen de datos de entrada mucho menor y a veces puede resultar insuficiente para un resultado fiable.

Para los comentarios de usuarios basados en los juegos las mejores herramientas son Stanford y RapidMiner al obtener las mejores precisiones y recall altos, indicando gran precisión y un número bajo de topics diferentes en todos los resultados, indica alto recall. Maui WIKI es también una decente opción con un recall alto, sin embargo el tiempo de ejecución para volúmenes grandes de datos convierte a RapidMiner en la mejor elección para el procesado de este tipo de textos.
Para la marca Starbucks Maui WIKI es la mejor opción, obtiene la mejor precisión, solo superado por Wandora por poco al mayor volumen de datos, y tiene la mejor recall. Solamente que su tiempo de ejecución es bastante alto a grandes volúmenes de datos, por lo que si lo importante es el tiempo Wandora se ha convertido en la mejor herramienta, fijándonos en este punto es la mejor opción aún cuando su recall es el más bajo de todas las herramientas, sino también por su alta precisión. Aunque si queremos algo estable en todos los criterios RapidMiner es la mejor herramienta, con una precisión buena, aunque algo inferior a Maui WIKI o Wandora en grandes volúmenes, pero con un recall decente, aunque superado de nuevo por Maui WIKI, pero finalmente con un tiempo de ejecución mucho más inferior a cualquier herramienta.

Para los comentarios de Youtube las conclusiones son parecidas a los comentarios de Starbucks, pero en este caso RapidMiner acaba superando a Maui WIKI en recall a grandes volúmenes, y la precisión de RapidMiner es ínfimamente inferior a Maui WIKI que es la mejor herramienta en precisión. Por lo que sumado a un tiempo de ejecución muy bajo convierte a RapidMiner en la mejor elección.

Sin embargo Wandora y Maui WIKI son buenas opciones por tener buenas precisiones, y Wandora aunque bajo recall es compensado por tener tiempos de ejecución inferiores por ejemplo a Maui KEY.

Concluyendo finalmente RapidMiner es la mejor herramienta evaluada y mejor compensada entre todos los criterios, muy bajos tiempos de ejecución, entre los mejores resultados en precisión y recall a grandes volúmenes y soporta una gran cantidad de volúmenes de datos de entada.

Y haciendo una pequeña distinción Maui con la configuración de indexación de Wikipedia es una buena opción para comentarios basados en alguna marca o logo, como ha sido el caso de Starbucks o Youtube, cuyos comentarios por parte de los usuarios es hablar de su producto o su propia marca, solamente si ignoramos que su tiempo de ejecución se dispara a mayor volumen aunque como no admite grandes volúmenes se compensan entre ellas estas dos desventajas.

Herramienta	Precisión	Recall	Runtime	Volumen datos
Maui KEY	Buena	Regular	Medio	Bajo (limitado)
Maui WIKI	Alta	Alta	Alto	Bajo (limitado)
Stanford NLP	Buena	Alto	Alto	Ilimitado
Wandora	Buena	Bajo	Bajo	Buena
RapidMiner	Alta	Alta	Muy bajo	Alto

7.2. Líneas futuras

Durante el desarrollo de la memoria se han mencionado posibles líneas de desarrollo para futuros proyectos.

Centrando básicamente en el desarrollo de esta memoria una línea futura es extender las evaluaciones y el análisis de datos en dos importantes vías.

La primera vía sería extender las evaluaciones con las mismas herramientas utilizadas en esta memoria con datos obtenidos de otras fuentes como por ejemplo Twitter, además de otras diversas temáticas.
Realizar evaluaciones para comentarios obtenidos de marcas similares para comparar posibles diferencias en su contenido.

La segunda vía sería con los mismos datos origen de esta memoria realizar experimentos con otras herramientas mencionadas en el proyecto u otras diferentes y extender las comparaciones para encontrar alguna herramienta aún más optima de las empleadas en la memoria.

Stanford permitía procesar los textos por periodos de fecha o autor, lo que nos permite pensar en una línea futura de evaluación de textos por fechas: meses, trimestres, años... y realizar una evaluación el tiempo.

Por último y el más trabajo futuro más sencillo sería calcular con los resultados obtenidos para las herramientas evaluadas en la memoria otro tipo de criterios de evaluación para realizar la comparación de los resultados.

Capítulo 8

Presupuesto del proyecto

8.1. Costes

A continuación se muestra un cálculo de costes asociados al proyecto, desde los costes de recursos del personal asociado al mismo, hasta los gastos de material o equipos.

Para la determinación del coste/mes de personal tendremos en cuenta los siguientes parámetros:

-Se precisa la intervención de un experto programador, que será el encargado de la realización de los experimentos y evaluaciones de las herramientas.

-No será necesaria la intervención de un analista de sistema: el conjunto ha de funcionar de forma correcta.

-No será necesaria la intervención de ningún técnico programador: no se precisa una preinstalación del programa, cualquier instalación en el equipo del usuario la realizará este mismo asistido por el propio programa.

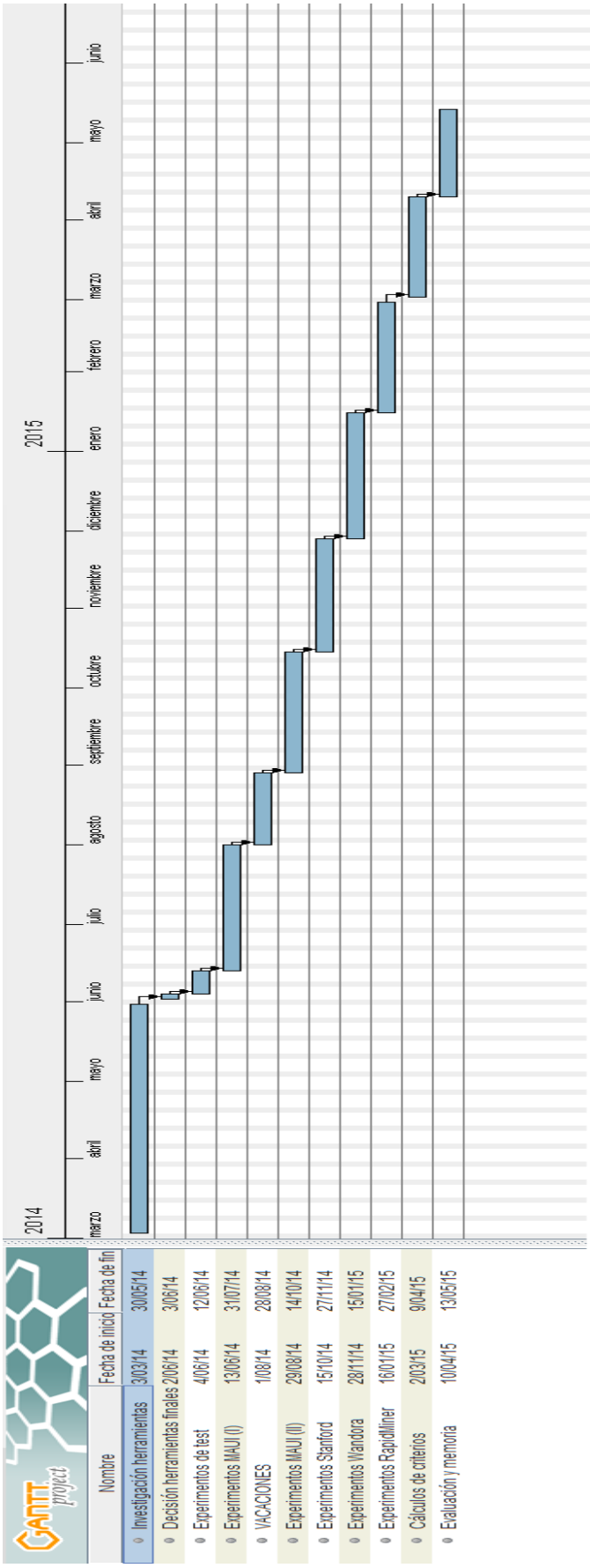
Presupuesto				
Autor	David García Rodríguez			
Departamento	Departamento de Ingeniería de Sistemas y Telemática			
Descripción del proyecto				
Titulo	Estudio de herramientas para Topic Detection and Tracking			
Duración	15 meses			
Desglose presupuestario				
Personal				
Nombre	Categoría	Dedicación	Coste al mes	Coste total
García Rodríguez, David	Ingeniero	15 meses	1900 €*	28500 €
			Total	28500 €
Material				
Descripción		Dedicación	Coste al mes	Coste total
Ordenador 1.86GHz 2 GB RAM		15 meses	0 €	500 €
Software Eclipse		3 meses	0 €	0 €
Software Stanford TMNT		1.5 meses	0 €	0 €
Software Wandora		1.5 meses	0 €	0 €
Software RapidMiner		1.5 meses	0 €	0 €
Conexión Internet		15 meses	40 €	600 €
			Total	1100 €

* La duración del proyecto es de 15 meses, aunque 1 mes fue de vacaciones, días laborales sin contar fines de semana y festivos, la jornada diaria de 8 horas.

Los costes indirectos por ser de determinación apriorística, se calcularán considerando un porcentaje estimado en un 12% de los costes directos.

Resumen de costes	
Presupuesto costes totales	Presupuesto total
Personal	28500 €
Material	1100 €
Costes directos (personal + material)	29600 €
Costes indirectos (12% costes directos)	3552 €
Total costes	33152 €

8.1. Gantt


GANTT project
Nombre
Fecha de inicio
Fecha de fin
Investigación herramientas 3/03/14 30/05/14
Decisión herramientas finales 2/06/14 3/06/14
Experimentos de test 4/06/14 12/06/14
Experimentos MAUI (I) 13/06/14 31/07/14
VACACIONES 1/08/14 28/08/14
Experimentos MAUI (II) 29/08/14 14/10/14
Experimentos Stanford 15/10/14 27/11/14
Experimentos Wandora 28/11/14 15/01/15
Experimentos RapidMiner 16/01/15 27/02/15
Cálculos de criterios 2/03/15 9/04/15
Evaluación y memoria 10/04/15 13/05/15
2014
2015
marzo
abril
mayo
junio
julio
agosto
septiembre
octubre
noviembre
diciembre
enero
febrero
marzo
abril
mayo
junio

Bibliografía

[1] http://definicion.de/facebook/

[2] Daniel T. Larose. Discovering Knowledge in Data. John Wiley and Sons, 2005.

[3] W. Graham. Facebook API Developers Guide. Apresspod Series. Apress, 2008.

[4] http://www.monografias.com/trabajos27/datamining/datamining.shtml

[5] Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Natl Acad Sci USA 1999, 96:2907-2912.

[6] Toronen P, Kolehmainen M, Wong G, Castren E. (1999). Analysis of gene expression data using self-organizing maps. FEBS Lett 1999, 451:142-146.

[7] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier
under zero-one loss. Mach. Learn., 29(2-3):103{130, November 1997.

[8] Pascal Hitzler, Markus Kr□otzsch, and Sebastian Rudolph. Foundations of Semantic Web
Technologies. Chapman & Hall/CRC, 2009.

[9] Nasukawa, T.; Nagano, T. (2001). Text analysis and knowledge mining system. IBM Systems Journal, knowledge management. Vol. 40 (4).

[10] Frawley, W. J. et al. (1991). Knowledge Discovery in Databases: An Overview. MIT Press.

[11] http://recuperacioninf.orgfree.com/trec.html

[12] Shatkay, H. and Feldman, R. (2003). Mining the biomedical literature in the genomic era: An overview. Journal of Computational Biology Vol. 10, 6, pp. 821-856.

[13] http://eprints.rclis.org/8759/1/Investigaci%C3%B3nRI.pdf

[14] Porter M F. (1980) An algorithm for suffix stripping. Program, 14 no. 3, pp 130-137.

[15] TDT Homepage at the National Institute of Standards and Technology, http://www.nist.gov/TDT

[16] Beeferman, D., Berger, A., and Lafferty, J., Text Segmentation Using Exponential

Models, In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pp. 35-46. 1997

[17] Martin, A., Doddington, G., Kamm, T., Ordowski, M., y Przybocki, M .; "La curva DET en la Evaluación de Desempeño de Tareas de detección", Actas de EuroSpeech 97, Volumen 4, páginas 1895 a 1898; Asociación Europea de Comunicación Oral (ESCA).

[18] https://gephi.org

[19] http://www.mathworks.co.uk/products/matlab/index.html

[20] http://www.graphviz.org

[21] https://code.google.com/p/maui-indexer/

[22] http://wikipedia-miner.cms.waikato.ac.nz/

[23] http://jung.sourceforge.net

[24] http://mahout.apache.org

[25] http://hadoop.apache.org/

[26] http://nlp.stanford.edu/software/index.shtml

[27] http://www.gnu.org/software/octave

[28] www.wandora.org/

[29] https://rapidminer.com/

[30] http://www.cs.waikato.ac.nz/ml/weka/

[31] http://www.r-project.org

[32] http://rattle.togaware.com/

[33] http://www.visualdataweb.org/relfinder.php

[34] http://www.gnu.org/software/pspp/

[35] http://www.nzdl.org/Kea

[36] http://es.wikipedia.org/wiki/Latent_Dirichlet_Allocation

[37] David M. Blei, Andrew Y. Ng, Michael I. Jordan, John Lafferty. 2003. Latent Dirichlet Allocation. JMLR.

[38] www.alchemyapi.com

[39] Markus Hofmann, Ralf Klinkenberg, "RapidMiner: Data Mining Use Cases and Business Analytics Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)," CRC Press, October 25, 2013.

[40] http://www.upf.edu/hipertextnet/numero-1/evaluacion_ri.html

[41] http://en.wikipedia.org/wiki/Precision_and_recall

[42] http://es.wikipedia.org/wiki/Desviación_típica

[43] http://www.vitutor.com/estadistica/descriptiva/b_13.html

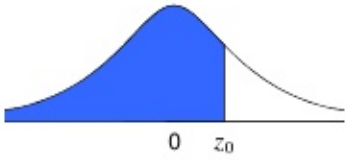
ANEXO Tabla Z

Tabla de la distribución normal N(0,1) para probabilidad acumulada inferior

μ = Media

σ = Desviación típica

$$P(z \le z_0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_0} e^{\frac{-z^2}{2}} dz$$



Tipificación: $z_0 = \frac{x - \mu}{\sigma}$

z_0	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09	z_0
0,0	0,5000	0,5040	0,5080	0,5120	0,5160	0,5199	0,5239	0,5279	0,5319	0,5359	0,0
0,1	0,5398	0,5438	0,5478	0,5517	0,5557	0,5596	0,5636	0,5675	0,5714	0,5753	0,1
0,2	0,5793	0,5832	0,5871	0,5910	0,5948	0,5987	0,6026	0,6064	0,6103	0,6141	0,2
0,3	0,6179	0,6217	0,6255	0,6293	0,6331	0,6368	0,6406	0,6443	0,6480	0,6517	0,3
0,4	0,6554	0,6591	0,6628	0,6664	0,6700	0,6736	0,6772	0,6808	0,6844	0,6879	0,4
0,5	0,6915	0,6950	0,6985	0,7019	0,7054	0,7088	0,7123	0,7157	0,7190	0,7224	0,5
0,6	0,7257	0,7291	0,7324	0,7357	0,7389	0,7422	0,7454	0,7486	0,7517	0,7549	0,6
0,7	0,7580	0,7611	0,7642	0,7673	0,7704	0,7734	0,7764	0,7794	0,7823	0,7852	0,7
0,8	0,7881	0,7910	0,7939	0,7967	0,7995	0,8023	0,8051	0,8078	0,8106	0,8133	0,8
0,9	0,8159	0,8186	0,8212	0,8238	0,8264	0,8289	0,8315	0,8340	0,8365	0,8389	0,9
1,0	0,8413	0,8438	0,8461	0,8485	0,8508	0,8531	0,8554	0,8577	0,8599	0,8621	1,0
1,1	0,8643	0,8665	0,8686	0,8708	0,8729	0,8749	0,8770	0,8790	0,8810	0,8830	1,1
1,2	0,8849	0,8869	0,8888	0,8907	0,8925	0,8944	0,8962	0,8980	0,8997	0,9015	1,2
1,3	0,9032	0,9049	0,9066	0,9082	0,9099	0,9115	0,9131	0,9147	0,9162	0,9177	1,3
1,4	0,9192	0,9207	0,9222	0,9236	0,9251	0,9265	0,9279	0,9292	0,9306	0,9319	1,4
1,5	0,9332	0,9345	0,9357	0,9370	0,9382	0,9394	0,9406	0,9418	0,9429	0,9441	1,5
1,6	0,9452	0,9463	0,9474	0,9484	0,9495	0,9505	0,9515	0,9525	0,9535	0,9545	1,6
1,7	0,9554	0,9564	0,9573	0,9582	0,9591	0,9599	0,9608	0,9616	0,9625	0,9633	1,7
1,8	0,9641	0,9649	0,9656	0,9664	0,9671	0,9678	0,9686	0,9693	0,9699	0,9706	1,8
1,9	0,9713	0,9719	0,9726	0,9732	0,9738	0,9744	0,9750	0,9756	0,9761	0,9767	1,9
2,0	0,9772	0,9778	0,9783	0,9788	0,9793	0,9798	0,9803	0,9808	0,9812	0,9817	2,0
2,1	0,9821	0,9826	0,9830	0,9834	0,9838	0,9842	0,9846	0,9850	0,9854	0,9857	2,1
2,2	0,9861	0,9864	0,9868	0,9871	0,9875	0,9878	0,9881	0,9884	0,9887	0,9890	2,2
2,3	0,9893	0,9896	0,9898	0,9901	0,9904	0,9906	0,9909	0,9911	0,9913	0,9916	2,3
2,4	0,9918	0,9920	0,9922	0,9925	0,9927	0,9929	0,9931	0,9932	0,9934	0,9936	2,4
2,5	0,9938	0,9940	0,9941	0,9943	0,9945	0,9946	0,9948	0,9949	0,9951	0,9952	2,5
2,6	0,9953	0,9955	0,9956	0,9957	0,9959	0,9960	0,9961	0,9962	0,9963	0,9964	2,6
2,7	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9972	0,9973	0,9974	2,7
2,8	0,9974	0,9975	0,9976	0,9977	0,9977	0,9978	0,9979	0,9979	0,9980	0,9981	2,8
2,9	0,9981	0,9982	0,9982	0,9983	0,9984	0,9984	0,9985	0,9985	0,9986	0,9986	2,9
3,0	0,99865	0,99869	0,99874	0,99878	0,99882	0,99886	0,99889	0,99893	0,99896	0,99900	3,0
3,1	0,99903	0,99906	0,99910	0,99913	0,99916	0,99918	0,99921	0,99924	0,99926	0,99929	3,1
3,2	0,99931	0,99934	0,99936	0,99938	0,99940	0,99942	0,99944	0,99946	0,99948	0,99950	3,2
3,3	0,99952	0,99953	0,99955	0,99957	0,99958	0,99960	0,99961	0,99962	0,99964	0,99965	3,3
3,4	0,99966	0,99968	0,99969	0,99970	0,99971	0,99972	0,99973	0,99974	0,99975	0,99976	3,4
3,5	0,99977	0,99978	0,99978	0,99979	0,99980	0,99981	0,99981	0,99982	0,99983	0,99983	3,5
3,6	0,99984	0,99985	0,99985	0,99986	0,99986	0,99987	0,99987	0,99988	0,99988	0,99989	3,6
3,7	0,99989	0,99990	0,99990	0,99990	0,99991	0,99991	0,99992	0,99992	0,99992	0,99992	3,7
3,8	0,99993	0,99993	0,99993	0,99994	0,99994	0,99994	0,99994	0,99995	0,99995	0,99995	3,8
3,9	0,99995	0,99995	0,99996	0,99996	0,99996	0,99996	0,99996	0,99996	0,99997	0,99997	3,9

$1-\alpha$	90%	92%	94%	95%	96%	97%	98%	99%
α	10%	8%	6%	5%	4%	3%	2%	1%
$z_{\alpha/2}$	1,645	1,751	1,881	1,960	2,054	2,170	2,326	2,576
z_α	1,282	1,405	1,555	1,645	1,751	1,881	2,054	2,326

Siendo:

$1-\alpha$ = Nivel de confianza

α = Nivel de significación