CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO
NACIONAL

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN DE COMPUTACIÓN

Sistema de análisis y filtraje de correo masivo no solicitado SPAM

Tesis que presenta

Francisco Javier Alejandro Lagunes

Para obtener el grado de

Maestro en Ciencias

en la Especialidad de

Ingeniería Eléctrica

opción Computación

Director de la tesis:

Dr. Arturo Díaz Pérez

México, D.F. Julio 2005

Resumen

El uso del correo electrónico como medio de comunicación y transmisión de información va en aumento debido a su eficiencia y facilidad de uso. Desafortunadamente, por estas mismas características es utilizado para enviar correos masivos no solicitados (SPAM). Los índices de correo SPAM van en aumento, y por ello son necesarias técnicas y métodos para abatir este problema.

Esta tesis presenta el desarrollo de un filtro inteligente para la detección y eliminación de correo SPAM. Este filtro esta construido por un conjunto de reglas deterministas y heurísticas así como un análisis estadístico.

Las reglas deterministas rechazan correos cuyo emisor ha sido previamente identificado como fuente de correo SPAM. Pero aceptan inmediatamente correos cuya fuente ha sido validada por el usuario. Las reglas heurísticas analizan el tema y cuerpo de un mensaje mediante la identificación de palabras o frases consideradas como características de correo SPAM. Finalmente, el análisis estadístico, basado en el teorema de Bayes, calcula la probabilidad de que las frases de cierto tamaño de un cierto mensaje pertenezcan a un mensaje tipo SPAM. Posteriormente se realiza un cálculo de probabilidad de que el correo sea SPAM.

El filtro propuesto identifica y elimina correctamente la mayoría del correo SPAM. Además es adaptable. La actualización se logra con la información de mensajes SPAM y no SPAM. Los porcentajes de detección mejoraron en un intervalo de 1 % a 4 % respecto a los sistemas actuales que filtran el 95 % de correo SPAM. Todo esto se logra por la combinación de las reglas deterministas, las heurísticas y por el análisis estadístico.

Palabras clave: correo electrónico, SPAM, reglas deterministas y heurísticas, filtro probabilístico.

Abstract

Nowadays the electronic mail (email) is the most common service at the internet. People usually use it for comunication and data transmision. However, its best features (eficiency and facility of use) have increased its used as SPAM mail. The amount of SPAM mails is rising every day, therefore, new SPAM-handling techniques are needed.

This thesis presents the development of an intelligent filter for detection and elimination of SPAM mails. This filter is built using a set of deterministic and heuristic rules, and a statistical analysis.

Deterministic rules reject such mails which senders have been previously identified as spammers, but if the sender have been authenticated by the user, then the email is accepted. Heuristic rules analyze the email's subjet and body to identify SPAM words or phrases. Finally, the statistical analysis calculates the email's spam probability using the Bayes theorem.

The filter proposed identifies and eliminates correctly the most of SPAM mails. Futhermore, it adapts its behavior using the information of valid and SPAM mails.

The proposed system outperforms in 1 to 4 % to some of the best SPAM filters.

Keywords: electronic mail, SPAM mail, deterministic and heuristic rules, probabilistic filter.

Agradecimientos

Agradezco a mis padres Genoveva Lagunes Monzón y Manuel Alejandro Guillen (finado), a mis hermanos y seres queridos por su apoyo incondicional. Esta tesis está dedicada a ustedes.

Agradezco a mi asesor, el Dr. Arturo Díaz Pérez por guiarme y aconsejarme en todo momento, además de compartirme sus conocimientos.

A mis sinodales, los Dres. Luis Gerardo de la Fraga y Guillermo Morales Luna, por su aportación para mejorar este documento de tesis.

Agradezco a Sofia por su amistad y apoyo en los aspectos administrativos durante mi estancia en la maestría.

Agradezco al CINVESTAV por facilitarme las instalaciones en las que curse las materias, así como el desarrollo este trabajo de tesis.

Agradezco a la sección de Computación por permitirme formar parte del programa de Maestría, así mismo agradezco a los Dres. que me enseñaron en sus clases.

Agradezco al CONACyT por la beca otorgada durante mi estancia en el programa de maestría en el CINVESTAV. Este trabajo de tesis se derivó del proyecto CONACyT titulado "Algoritmos y Arquitecturas con Dispositivos Reconfigurables"(Ref. CONACyT 31892-A) cuyo responsable es el Dr. Arturo Díaz Pérez.

Agradezco a mis compañeros y amigos por brindarme su amistad y apoyo, haciendo de mi estancia en la maestría, algo agradable.

Índice general

Índice de figuras

Índice de algoritmos

Capítulo 1

Introducción

En el entorno de la red Internet se cuenta con una variedad de servicios, entre los que destaca el servicio de correo electrónico, el cual permite comunicarnos con rapidez y de forma sencilla con otros usuarios. Entre otras ventajas tenemos el ahorro de recursos, debido a que sustituye el uso del correo ordinario. Elimina muchas de las llamadas telefónicas, sobre todo las de larga distancia y nos permite utilizarlo como fax. Por todas sus aportaciones en la comunicación y el envío de información, es sin duda de gran trascendencia en los medios de comunicación electrónica.

En todo este ambiente del servicio de correo electrónico se ha observado la presencia de correo masivo no solicitado, denominado correo SPAM. El SPAM contiene publicidad, invitaciones de visitas a otros sitios Web, entre otros contenidos. También pueden contener archivos con virus o programas pasivos, estos últimos usados para espiar el contenido de nuestra computadora.

La presencia del correo SPAM causa disminución en el rendimiento de la red Internet y de los sistemas de cómputo. Entre los efectos dañinos tenemos, la saturación de mensajes en los servidores Web y en las cuentas de correo, el incremento en el tráfico en Internet, y la disminución de la productividad por el tiempo dedicado a atender otros mensajes, entre otros efectos. La forma de este fenómeno es que sin ser solicitados llegan correos SPAM al buzón de un usuario.

Se sabe que la cantidad de usuarios de correo electrónico va en aumento. Por otro lado, existen usuarios llamados *spammers* dedicados a capturar cuentas de correo electrónico las cuales usan para distribuir publicidad.

En promedio 71.4 billones de correos por día circularon por Internet en el año 2004 [17], del cual el 70 % - 80 % se consideró como SPAM [18], y se estima que para el 2005 el 96 % sea SPAM [18]. De todos los usuarios, poco más del 78 % ha recibido correo SPAM [19]. Las empresas alrededor del mundo gastaron sólo en el año 2004 poco más de 20 mil millones de dólares para abatir el SPAM [19]. Por otro lado, el borrado del correo SPAM le cuesta a las empresas hasta 22 mil millones de dólares anuales en pérdida de productividad [19].

Para concluir con estas estadísticas veamos lo que se presenta en el Centro de Investigaciones y de Estudios Avanzados del IPN (CINVESTAV-IPN). En el CINVESTAV-IPN existen en promedio 1500 cuentas de correo electrónico, y se ha observado que en promedio se reciben 10 mensajes SPAM diarios por cuenta. También en algún momento se han observado ataques de 8,000 mensajes SPAM por hora. Dado que el fenómeno SPAM ha ido en aumento, es necesario estudiarlo y proporcionar alternativas de solución que eliminen o cuando menos aligeren sus efectos.

Algunas de las alternativas para combatir al fenómeno del SPAM son el uso de la heurística *lista negra*, dicho método consiste en bloquear los mensajes de remitentes que estén incluidos en la lista negra, otra alternativa es la heurística *palabras clave*, las cuales son consideradas como características de mensajes SPAM y a cada palabra se le asocia un valor ponderado de pertenecer a un mensaje SPAM, este método consiste en definir a los mensajes como SPAM a todos aquellos que contienen una cantidad considerable de dichas palabras y que alcanzan el valor del umbral de ser SPAM, otra alternativa es la definición de la heurística de *frases clave*, siendo estas frases significativas y propias de un mensaje SPAM, el mensaje que contenga alguna de esas frases es definido inmediatamente como SPAM. Existen algunas otras alternativas más eficientes como el *método probabilístico*, el cual es un método predictivo basado en datos estadísticos tomados de mensajes SPAM para predecir si un cierto mensaje es SPAM. Existen algunas herramientas que implementan algunos de estos métodos de análisis tal es el caso de la herramienta *SpamAssassin* [27]. En el artículo "A plan for spam" de Paul Graham [3] describe un algoritmo probabilístico bayesiano. De manera general, el algoritmo en su primera etapa obtiene la probabilidad de ciertos tokens de que sean parte de un mensaje SPAM, en la segunda etapa, con dichas probabilidades es posible predecir si un cierto mensaje que contenga algunos de esos tokens sea considerado como SPAM.

1.1. Planteamiento del problema

Debido a la presencia de correos SPAM y de las consecuencias que estos generan surge la necesidad de contar con métodos y técnicas para eliminar o disminuir el fenómeno SPAM, en este caso es necesario un sistema que detecte y filtre el correo masivo no solicitado.

Al momento de definir los métodos y las técnicas, se debe definir claramente las características de los correos válidos y de los que deben ser considerados como SPAM. También se debe evitar que el sistema tenga errores en la clasificación. Es decir, que no obtenga correos *falsos positivos*, este tipo de correos son clasificados como SPAM sin serlo. Por otro lado, que tampoco obtenga correos *falsos negativos*, es decir cuando un correo es clasificado como válido, sin serlo.

Algo que se debe tener presente es una solución que cubra las necesidades de la variedad de usuarios de correo electrónico, es decir, unir criterios y obtener una solución generalizada y adecuada a las necesidades. Tampoco se debe olvidar que todo sistema para que se mantenga vigente, debe tener la característica de ser mantenible, en este caso se debe considerar el surgimiento de correos SPAM con características nunca antes vistas.

Lo importante de esta solución para aumentar eficacia es que se debe integrar para detectar y filtrar el correo SPAM. Funcionando en los tres niveles jerárquicos, a nivel organización, a nivel servidor y a nivel de usuario.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar e implantar un sistema que detecte y elimine el correo SPAM, haciendo uso de reglas deterministas y heurísticas, así como un análisis estadístico.

1.2.2. Objetivos específicos

- Detectar y filtrar el correo SPAM cuyo emisor ha sido previamente identificado como fuente de correo SPAM.
- Detectar y filtrar el correo SPAM haciendo un análisis del tema y cuerpo del mensaje, basado en una lista de palabras clave para buscar características de correo SPAM.

- Detectar y filtrar el correo SPAM haciendo un análisis del tema y cuerpo del mensaje, basado en una lista de frases clave, las cuales definen por si solas a los mensajes como SPAM.

- Detectar y filtrar el correo SPAM haciendo un análisis estadístico, basado en el Teorema de Bayes, para calcular la probabilidad de frases de tamaños variables pertenezcan a un correo SPAM.

- Integrar todo lo anterior, y ponerlo en funcionamiento en los tres niveles jerárquicos: a nivel buzón de usuario y a nivel servidor.

- El sistema de análisis de mensajes se debe integrar en un entorno real de servicio de entrega de correos, para nuestro caso, el sistema debe funcionar en al menos un servidor de correos del CINVESTAV-IPN, apoyado de la herramienta de UNIX *procmail* (procesador de correos local).

1.3. Metodología

Considerando la funcionalidad que ofrece *procmail*, al momento de que un correo llega a la entrada estándar, *procmail* evalúa un archivo de reglas, dichas reglas indican el buzón para depositar el correo, también es posible llamar a un programa externo para que analice el correo y esperar el resultado del análisis. Para nuestro caso se manda llamar el sistema de análisis de correo y con el resultado del análisis determina si el mensaje es válido o SPAM.

En términos generales, la propuesta que se tiene se muestra en la Figura 1.1, en dicha propuesta se usa a *procmail* como base y a través de un archivo de reglas se llama a un programa externo. Con esto se obtiene un sistema que filtra el correo SPAM de manera efectiva.

A continuación se describe como se obtuvo el sistema de análisis y filtraje de correo SPAM, considerando siempre los objetivos de la tesis.

- Se revisó la literatura actual sobre el correo electrónico, estadísticas del servicio de correo, el problema del SPAM y sus características, así como alternativas de solución al fenómeno del SPAM. Esta recopilación de información se hizo durante el proyecto, en particular para el desarrollo de cada módulo del sistema de análisis propuesto.

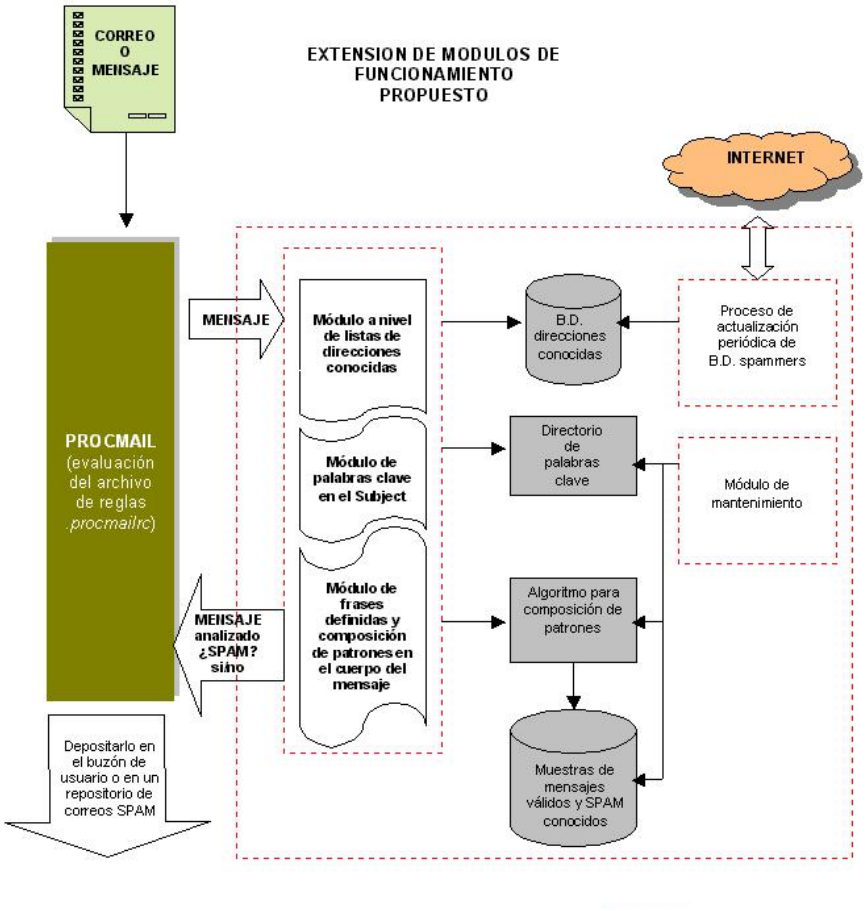

Figura 1.1: Esquema general de solución del sistema antispam.

- Se desarrolló el módulo de detección y filtraje de correo SPAM a nivel de listas de direcciones definidas por el usuario como fuente de correo SPAM, las llamadas *listas negras*. De manera similar, se tienen *listas blancas*, cuya fuente es validada por el usuario.

- Se desarrolló el módulo de análisis basado en una lista de palabras clave. Este análisis se hace al tema y cuerpo del mensaje, obteniendo con ello una ponderación de ser SPAM.

- Se desarrolló el módulo de análisis basado en frases definidas como propias de un mensaje SPAM. Este análisis también aplica para el tema y cuerpo de mensaje, al

encontrarse alguna de dichas frases clave el mensaje es definido como SPAM.

- Se desarrollaron métodos y técnicas más efectivas y dinámicas basadas en el Teorema de Bayes, la forma base es predecir si un determinado mensaje es SPAM, dadas ciertas evidencias, en este caso las evidencias son las palabras y frases contenidas en el mensaje. También se basa en datos estadísticos.

- Para obtener un sistema robusto y efectivo se integraron los módulos anteriores. Además, es posible hacer funcionar el sistema de manera jerárquica, a nivel servidor y a nivel usuario. El análisis a nivel servidor se logra indicando ejecutar el sistema de análisis desde el archivo de reglas */etc/procmail*. Para el análisis a nivel de usuario se indica ejecutar el sistema desde el archivo de reglas *$HOME/.procmailrc*.

- Para validar los resultados, se presentaron dos casos de estudio, en los cuales es posible mostrar los porcentajes de efectividad. Es decir, de los mensajes de prueba enviados se sacaron estadísticas de resultados que permiten mostrar cuantos mensajes fueron filtrados por el sistema desarrollado y cuantos errores de clasificación hubo.

1.4. Organización de la tesis

El capítulo 2 contiene la teoría básica necesaria para la realización de este trabajo de tesis. Conceptos como filtros inteligentes, reglas deterministas, heurísticas, filtros bayesianos, fundamentos de probabilidad, conceptos de correo electrónico, etc.

El capítulo 3 presenta el algoritmo extendido para el análisis y filtraje de correos SPAM, basado en el cálculo de probabilidades de frases de tamaños variables pertenezcan a un correo SPAM. El capítulo 4 explica el diseño e implementación del filtro inteligente para la detección y eliminación de correo SPAM. El capítulo 5 muestra los resultados obtenidos en las pruebas de efectividad y rendimiento, aplicadas al sistema de análisis y filtraje de correo SPAM. El capítulo 6 menciona las conclusiones y logros alcanzados con el desarrollo de este trabajo de tesis, también menciona líneas de investigación y estudio de trabajos futuros para el mejoramiento de lo realizado.

Capítulo 2

Marco teórico

En este capítulo se describen el servicio de correo, los protocolos utilizados para el intercambio de correo electrónico, así como los protocolos utilizados por las aplicaciones de correo y la herramienta de UNIX *procmail*.

Una vez definido el problema del SPAM se describen los tipos de análisis de mensajes que se pueden realizar para, además se presenta un algoritmo de análisis estadístico basado en la información de sus tokens. Por último se presentan dos herramientas para la detección de mensajes SPAM.

2.1. Correo electrónico

El correo electrónico permite enviar cualquier tipo de información entre cualesquiera par de personas que tienen un buzón o dirección electrónica.

El envío de mensajes a través de la red Internet conlleva ventajas como son, *rapidez* en llegar los mensajes a su destino, *económicamente* es barato, ya que es posible enviar un mensaje a cualquier parte del mundo con el costo de una llamada local, es *confiable* debido a que no se pierden mensajes y cuando por alguna razón no pueden ser entregados se envía devuelta un mensaje al remitente.

Para un mejor entendimiento del servicio de correo electrónico, se muestra la figura 2.1, en ella se presenta un esquema global del servicio de correo electrónico, muestra los elementos involucrados en el proceso, así como los protocolos y los agentes de correo que participan en él.

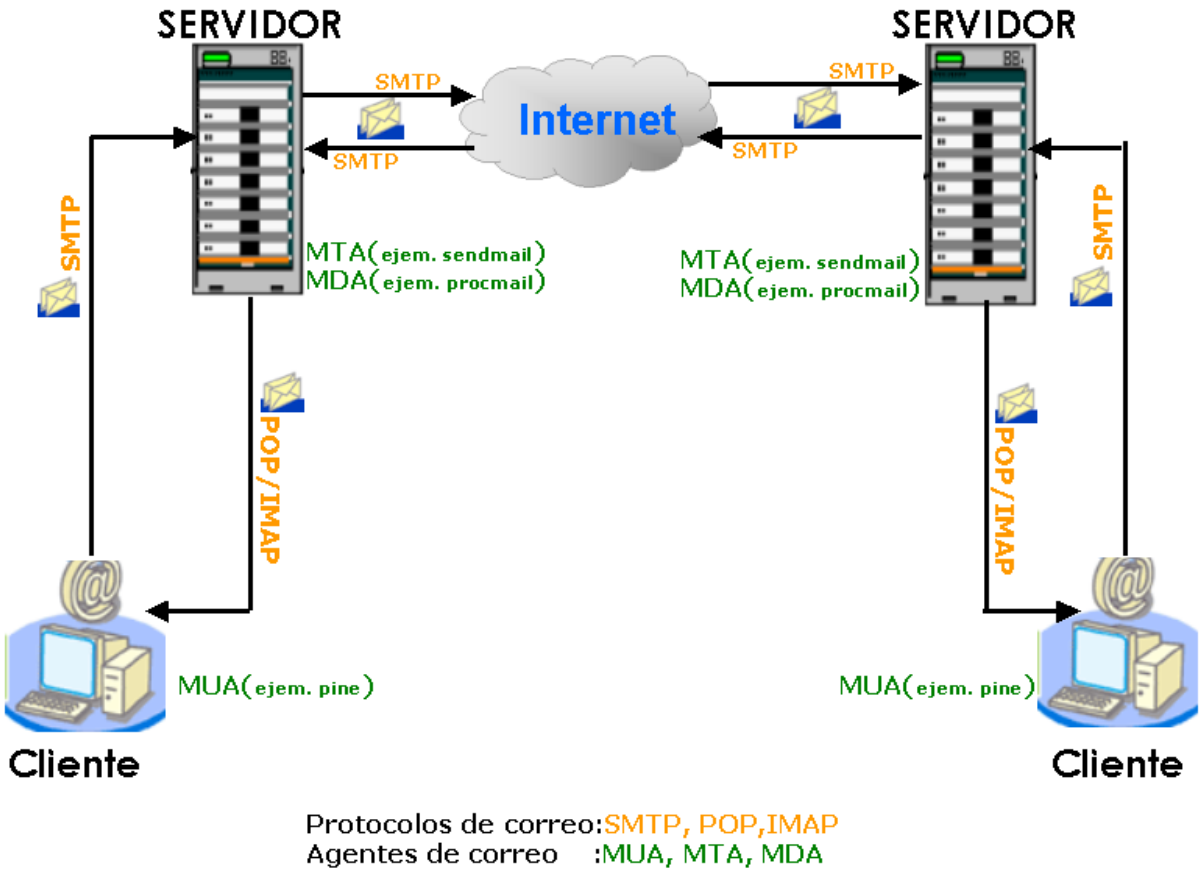

Figura 2.1: Esquema del servicio de correo electrónico.

El servicio de correo electrónico utiliza diversos protocolos para permitir que distintas máquinas con sistemas operativos posiblemente diferentes y con programas de correo electrónico distintos se comuniquen entre sí y transfieran correo electrónico para que llegue a los destinatarios adecuados.

Existen dos tipos de protocolos de correo: *protocolos de transporte de correo* y *protocolos de acceso al correo*. El primero se encarga de entregar correo desde una aplicación cliente a un servidor y desde un servidor origen al servidor destino. El segundo permite a una aplicación cliente recuperar correo desde un servidor de correo.

Protocolo de transporte

SMTP-Simple Mail Transfer Protocol, protocolo simple de transferencia de correo

Este protocolo es el estándar de Internet para el intercambio de correo electrónico. Su objetivo principal es transferir correo entre servidores de correo. SMTP usa el puerto 25 del servidor para comunicarse.

Para poder enviar correo, el cliente envía el mensaje a un servidor de correo saliente, el cual se conecta con el servidor de correo destino para la entrega. Algo importante sobre el protocolo SMTP es que no requiere autenticación, esto permite que cualquier usuario de Internet pueda enviar correo a cualquier otra persona o a grandes grupos de personas, y es el fallo de seguridad que permite la existencia del SPAM.

Protocolos de acceso al correo

Los dos protocolos más importantes usados por las aplicaciones de correo cliente para recuperar correo desde los servidores de correo son: *POP* e *IMAP*. A diferencia de SMTP, estos protocolos requieren autenticación de los clientes usando un nombre de usuario y contraseña. Por defecto, las contraseñas para ambos protocolos son enviadas a través de la red, de forma encriptada.

POP-Post Office Protocol, protocolo de oficina de correos

Cuando se utiliza este protocolo, los correos son descargados a través de las aplicaciones de correo cliente. Por defecto, la mayoría de los clientes de correo POP son configurados para borrar automáticamente el mensaje del servidor de correos después de que éste ha sido transferido exitosamente, sin embargo, esta configuración se puede cambiar.

Para establecer una conexión a un servidor POP, el cliente de correo abre una conexión TCP en el puerto 110 del servidor.

IMAP-Internet Message Access Protocol-protocolo de acceso a mensajes de Internet

Al utilizar este protocolo, los correos se mantienen en el servidor donde los usuarios los pueden leer y borrar. IMAP también permite a las aplicaciones cliente crear, renombrar o borrar directorios en el servidor para organizar y almacenar correos. Además, IMAP es compatible con importantes estándares de mensajes de Internet, como MIME (Multi-

purpose Internet Mail Extensions, extensiones de correo de Internet multipropósito), que permiten recibir archivos adjuntos.

IMAP lo utilizan principalmente los usuarios que acceden a su correo desde varias máquinas. El protocolo es conveniente también para usuarios que estén conectados al servidor de correos a través de una conexión lenta, porque sólo la información de la cabecera del correo es descargada hasta que son abiertos, ahorrando de esta forma ancho de banda. El usuario también tiene la opción de eliminar mensajes sin ver su contenido ni descargarlos.

Por seguridad adicional en los protocolos POP e IMAP es posible utilizar la encriptación *Secure Socket Layer* (SSL) para la autenticación del cliente y las sesiones de transferencias de datos.

Agentes de correo electrónico

Hay tres tipos de programas de correo electrónico, los cuales desempeñan un papel específico en el proceso de trasmitir y administrar mensajes de correo electrónico.

MUA-Mail User Agent- agente de usuario de correo

Un MUA es un programa que permite a un usuario como mínimo leer y escribir mensajes de correo electrónico. A un MUA se le denomina a menudo cliente de correo. Hay programas MUA que ofrecen al usuario más funciones, entre las que se incluyen la recuperación de mensajes mediante los protocolos POP e IMAP. Los programas MUA pueden ser gráficos o en modo texto (ejemplo Pine, Eudora, Outlook, Mozilla mail, etc.).

MTA-Mail Transfer Agent, agente de transferencia de correo

Un agente MTA transfiere el correo electrónico entre máquinas que usan el protocolo SMTP. Un mensaje puede pasar por varios MTA hasta llegar al destino final.

El proceso de envío de mensajes entre las máquinas puede parecer bastante directo, todo el proceso de decidir si un agente MTA concreto puede o debe aceptar un mensaje para entregarlo a un host remoto es bastante complicado. Además, debido a los problemas de correo basura, el uso de un MTA concreto normalmente está limitado por la propia configuración del MTA o el acceso a la red del sistema que lo ejecuta. Ejemplos de un MTA son Sendmail y Postfix.

MDA-Mail Delivery Agent, agente de entrega de correo

Los agentes MTA utilizan programas MDA para entregar el correo electrónico al buzón de un usuario específico. El agente MDA es realmente un LDA (Local Delivery Agent, agente de entrega local), como *procmail*. Un agente MDA se encarga de gestionar mensajes para entregarlos para que sean leídos por un agente MUA. Un agente MDA no transporta mensajes entre sistemas ni actúa como interfaz para el usuario final.

Un *buzón de correo electrónico* es donde se van almacenando los mensajes para su posterior lectura. Cada persona tiene su buzón y solo ella podrá acceder a la información a través de un programa MUA (ejemplo Pine).

Para el envío o recepción de un mensaje es importante distinguir las partes que lo componen:

- To (Destinatario) se refiere a dirección de la persona a la que va dirigido el mensaje.
- Cc (Carbon copy) contiene las direcciones de las personas a las que se quiere enviar al mismo tiempo una copia del mensaje.
- Bcc (Blind carbon copy) permite enviar copias a otros destinatarios sin que su dirección aparezca en las copias de los demás. Está posibilidad solo la ofrecen algunos programas.
- Subject se refiere al asunto del mensaje, el cual debe ser una palabra o una frase corta. Es útil para dar una idea de lo que se trata en el mensaje.
- Attach se refiere a los archivos que se adjuntan al mensaje y que serán enviados al mismo tiempo que éste.
- Cuerpo se refiere al contenido del mensaje.

En la figura 2.2 se muestra el ejemplo de un mensaje ilustrando las partes que lo integran.

2.1.1. Procmail

Hay varios programas que permiten gestionar el correo electrónico. Una vez que se dispone de correo es necesario clasificarlo. *Procmail* permite separar el correo en diferentes

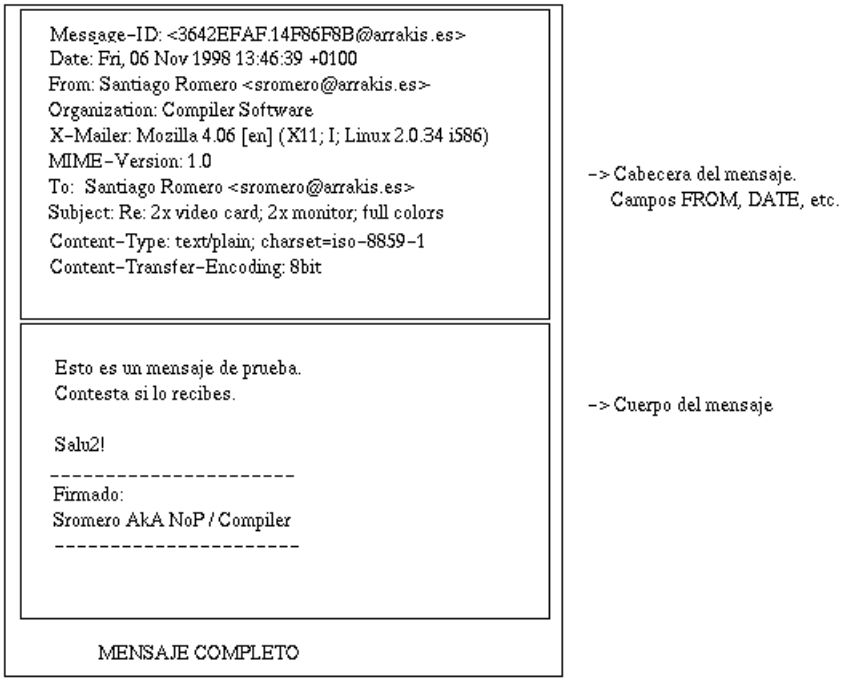

Figura 2.2: Componentes de un correo electrónico.

carpetas, leer con más comodidad las listas de correo, borrar mensajes no deseados, tener autorespuestas, entre otras acciones.

Procmail es un procesador de correo autónomo que se ejecutará en un servidor al producirse la llegada de un nuevo mensaje. *Procmail* es llamado automáticamente gracias al archivo *.forward* o a través de *sendmail*, de tal modo que leerá el correo de la entrada estándar y abrirá el archivo */etc/procmailrc* o el archivo *$HOME/.procmailrc* en ellos se indica qué debe hacer con ese correo, basándose en una serie de reglas que han sido definidas.

En un funcionamiento normal sin *procmail* se agrupan uno tras otro los diferentes correos que se reciben, concatenándose todos en el mismo archivo */var/spool/mail* con el nombre de cada usuario hasta que el usuario recoge el correo y el archivo es vaciado. Mediante *procmail* se puede alterar ese proceso y eliminar mensajes o dejarlos en otro archivo diferente, basado en un conjunto de reglas.

Las reglas de procmail se refieren al análisis del contenido del mensaje en sí (tanto

la cabecera como el cuerpo del mismo), permiten eliminar correo SPAM (al menos un porcentaje), así como repartir los mensajes entre los diferentes usuarios de correo.

Procmail permite filtrar el correo a medida que lo recibe de un servidor de correo remoto y lo deposita en el archivo *spool* de un servidor de correo local o remoto. Comúnmente *procmail* es denominado LDA (Local Delivery Agent - Agente de entrega local), desempeñando una pequeña función en la entrega de correo para su lectura por un agente MUA (ejemplo Pine).

Funcionamiento Básico de *Procmail*

1. Se lee de la entrada estándar un mensaje.

2. Se consultan los archivos */etc/procmail* y *$HOME/.procmailrc*. Estos archivos indican qué hacer con el mensaje que ha leído, de acuerdo a las reglas definidas por el usuario.

3. Se podrá indicar que compruebe la cabecera del mensaje y busque ciertas cadenas en ella para decidir guardar el mensaje, descartarlo, o contestar automáticamente.

4. *Procmail* permitirá tratar el correo que llegue, o almacenarlo en un archivo de manera automática.

Una de las características más importantes de *procmail* es la posibilidad de ejecutar un programa de usuario por medio de parámetros definidos en el archivo de reglas. También es posible detener el flujo de la evaluación del archivo de reglas mientras no termine la ejecución del programa.

A continuación se muestra un ejemplo de una regla de procmail, dicha regla es definida en el archivo */etc/procmail* o *$HOME/.procmailrc*.

: 0 :	← indica inicio de una regla
$*\hat{}From$: *spammer@domain*	← es la condición, si coincide el origen
/home/user/mail/spam	← es la acción, aquí colocará el mensaje

2.2. El fenómeno del correo SPAM

En el contexto del servicio de correo electrónico se presentan los mensajes *SPAM*, definiéndose así a los mensajes que se envían de manera masiva y automatizada, además que el receptor no solicita ni permite expresamente de forma verificable el envío del mensaje [20]. Cuando la cantidad de SPAM que circulan en Internet es elevada se considera un problema, también definido como fenómeno SPAM. De esto se tienen algunas estadísticas.

La cantidad de correo electrónico que circuló por Internet en el año 2004 fue de 71.4 billones de correos por día [17], del cual el 70 % - 80 % se consideró como SPAM [18], y se estima que para el 2005 el 96 % de mensajes generados sea SPAM [18], esto se debe a que las empresas que lo generan, principalmente agencias de mercadotecnia, han sofisticado sus sistemas al grado de engañar a los programas antispam. La cantidad de usuarios que recibe correo SPAM regularmente es de poco más de 78 % y tardan un promedio de cinco minutos en borrarlos [19]. Por otro lado, las empresas alrededor del mundo gastaron poco más de 20 mil millones de dólares en el año 2004 para contrarrestar el SPAM, además de la pérdida de productividad [19].

Debido a los efectos del fenómeno del SPAM es necesario plantear alternativas para eliminarlo.

2.2.1. Alternativas para eliminar el correo SPAM

Para poder eliminar el correo SPAM, es necesario llevar a cabo un proceso de análisis y detección, para lo cual se han propuesto algunos mecanismos antispam, los cuales pueden ser de dos tipos: *estáticos* y *dinámicos*.

Filtros estáticos, se basan en un conjunto limitado de palabras, frases y reglas estáticas, analizan tanto la cabecera como el cuerpo de los mensajes. Las palabras, frases, y reglas son muy sencillas. Su efectividad es limitada.

Filtros dinámicos, realizan el análisis a partir de un conjunto de palabras y reglas, son mucho más completas. Estos modelos son capaces de cambiar su respuesta según obtienen nuevos datos de los mensajes considerados como SPAM y no SPAM. Algo similar a la forma en que los humanos aprendemos a identificar el correo SPAM.

Los mecanismos estáticos incluyen el uso de las listas negras, listas blancas, palabras y frases clave en el tema y cuerpo del mensaje. El modelo dinámico desarrolla un modelo basado en la estadística y la probabilidad. A continuación se describen los mecanismos básicos de detección de mensajes SPAM.

Listas negras, se basan en la idea de localizar las direcciones origen de correo basura y bloquear todos los mensajes que lleguen de ellas.

Palabras clave en el tema, se encarga de buscar una cantidad considerable de palabras características de mensajes SPAM, y se define un valor de umbral de ser SPAM.

Frases clave en el tema, estas frases son significativas propias de un mensaje SPAM, el mensaje que contenga alguna de estas es definido inmediatamente como SPAM.

Análisis de contenido, los mecanismos de búsqueda de palabras clave y de frases clave en el tema se aplican de manera similar para el cuerpo del mensaje.

Estos mecanismos de análisis ayudan a eliminar el correo SPAM. Sin embargo, no son tan efectivos en el análisis y pueden presentar errores en la detección de SPAM, los cuales son:

Falsos positivos, este error existe al definir un mensaje válido como SPAM.

Falsos negativos, caso contrario, este error se presenta cuando un mensaje SPAM es definido como válido.

Debido a los errores presentados en los mecanismos básicos se han propuesto nuevas alternativas, como es el caso de los filtros inteligentes.

2.3. Filtros inteligentes

Los filtros inteligentes intentan detectar la mayor cantidad de mensajes SPAM con un mínimo de falsos positivos. Los filtros basados en el análisis del contenido son la mejor manera de contrarrestar el envío de SPAM, ya que los SPAM pueden saltar todas las barreras que les pongamos menos las que se basen en el mismo contenido del mensaje.

Así lo muestra Paul Graham en sus artículos *"A plan for spam"* [3] y *"Better bayesian filtering"* [4], en donde hace uso de la *lógica bayesiana*.

La lógica bayesiana creada por el matemático inglés Thomas Bayes, se basa en la *estadística* y la *probabilidad*, en el caso de los mensajes de correo electrónico se utiliza para predecir si un mensaje es SPAM o no. Con lo anterior, es posible escribir un algoritmo que filtre los mensajes que contengan palabras características de un mensaje SPAM y mejor aún, que se adapte con el tiempo a la variedad de los novedosos mensajes SPAM.

Hay dos tipos fundamentales de filtros inteligentes: los *basados en heurísticas* y los *adaptativos o bayesianos*.

2.3.1. Filtros con heurísticas

Este tipo de filtro se basa en un conjunto de reglas y patrones que se aplican al texto del mensaje. Los patrones empleados, las reglas y el valor ponderando de ser SPAM, todo es definido por el usuario y pueden ajustarse para mejorar la eficiencia del análisis.

La introducción de heurísticas es un factor importante para la detección de correo SPAM. El tipo de heurísticas que se utilizan son: *heurísticas en la cabecera* y *heurísticas en el cuerpo*. Las cuales se refieren a la definición de características localizadas en la cabecera (asunto) y cuerpo del mensaje. Para este análisis se definen palabras y frases, todas ellas características propias de un correo SPAM, como por ejemplo:

PALABRAS	FRASES
sex	penis enlargement
free	Get a job with enhanced income
promotion	no desea seguir

2.3.2. Filtros adaptativos o bayesianos

La *inferencia bayesiana* se basa en el conocimiento a *priori* de cierta información, lo que permite conocer la probabilidad de que ocurra un hecho en función de la probabilidad de que ocurra otro. Estos filtros predicen si un mensaje es SPAM o no SPAM, calculando la probabilidad asociada a cada evidencia (token) del mensaje.

De esta manera los *filtros adaptativos* calculan, utilizando el *teorema de Bayes*, la probabilidad de que un correo sea SPAM a partir de la información que se les proporciona

en los mensajes seleccionados como SPAM y no SPAM. Estos mensajes deben ser numerosos y representativos, siendo ésta la clave para obtener buenos resultados en el análisis. Su mayor ventaja es la capacidad de adaptación a las nuevas características de mensajes SPAM.

2.4. Teorema de Bayes

El razonamiento bayesiano proporciona un enfoque probabilístico a la inferencia. Está basado en la suposición de que las cantidades de interés son gobernadas por distribuciones de probabilidad y que se pueden tomar decisiones óptimas razonando sobre estas probabilidades junto con los datos obtenidos.

Asumamos que se tiene un conjunto de hipótesis posibles, h_i $(1 \leq i \leq n)$, y un conjunto de datos observados, D. Ahora bien, el teorema de Bayes representa la probabilidad a posteriori que una hipótesis particular h_i tiene, dados los datos observados D. En otras palabras, muestra qué tan probables son los diferentes valores posibles del conjunto de hipótesis, una vez obtenidos los datos D.

Si deseamos determinar la hipótesis h_i más probable, dados los datos observados D más un conocimiento inicial sobre las probabilidades a priori de h_i, entonces de acuerdo a [21][22][23] [24] con el teorema de Bayes se pueden calcular estas probabilidades. Sean: h un conjunto de hipótesis posibles, D el conjunto de evidencias observadas, $P(h_i)$ el conocimiento inicial que se tiene sobre que la hipótesis h_i sea la correcta (se le suele llamar probabilidad a priori de h_i), $P(D)$ se define como el conocimiento inicial que se tiene sobre que los datos D existan, $P(D|h_i)$ denota la probabilidad de observar los datos D dado que tenemos la hipótesis h_i (se le suele denominar verosimilitud y es la información que aportan los datos observados D a nuestro caso).

Con lo anterior se puede actualizar el valor a priori $P(h_i)$, con base en los datos obtenidos D y calcular $P(h_i|D)$, la probabilidad a posteriori utilizando el teorema de Bayes.

$$P(h_i|D) = \frac{P(h_i \cap D)}{P(D)} = \frac{P(D|h_i) * P(h_i)}{\sum_{i=1}^{n} P(D|h_i) * P(h_i)} \tag{2.1}$$

En la fórmula 2.1, el numerador se define como la regla de la multiplicación y el denominador como la regla de la probabilidad total.

Utilizando el teorema de Bayes para el análisis de mensajes, es posible calcular la probabilidad de que un mensaje m que contiene t tokens tomados como evidencias sea considerado de tipo SPAM. Al establecer una relación con la fórmula 2.1 para el análisis de mensajes, se asume que se tienen dos hipótesis posibles, la hipótesis $\mathrm{M_{SPAM}}$, que representa la categoría de un mensaje m de ser SPAM, y la hipótesis $\mathrm{M_{NOSPAM}}$, que representa la categoría de que el mensaje m no sea SPAM.

Si se desea determinar la probabilidad de la hipótesis $\mathrm{M_{SPAM}}$, se define a $\mathrm{P(M_{SPAM})}$ como el conocimiento inicial que se tiene acerca de que un mensaje m aleatorio sea SPAM, $\mathrm{P}(t)$, se define como el conocimiento inicial que se tiene acerca de que los tokens t existan, $\mathrm{P}(t|\mathrm{M_{SPAM}})$ representa la probabilidad de que existan los tokens t en un mensaje m de tipo SPAM.

Considerando lo anterior, es posible determinar la probabilidad de que un mensaje m que contiene los tokens t, sea de tipo SPAM, la cual es considerada como probabilidad a posteriori, y está definida como $\mathrm{P(M_{SPAM|}}_t)$. Dicho cálculo se obtiene con la ecuación siguiente, basada en el teorema de Bayes:

$$\begin{aligned} P(\mathrm{M_{SPAM}}|t) &= \frac{(\mathrm{M_{SPAM}} \cap t)}{P(t)} = \\ &= \frac{P(t|\mathrm{M_{SPAM}}) * P(\mathrm{M_{SPAM}})}{P(t|\mathrm{M_{SPAM}}) * P(\mathrm{M_{SPAM}}) + P(t|\mathrm{M_{NOSPAM}}) * P(\mathrm{M_{NOSPAM}})} \end{aligned} \tag{2.2}$$

La fórmula 2.2 involucra dos hipótesis, una es que un mensaje m sea de tipo SPAM-$(\mathrm{M_{SPAM}})$ y otra de que un mensaje m sea de tipo no SPAM$(\mathrm{M_{NOSPAM}})$. Donde el numerador representa la probabilidad estimada de que los tokens t se encuentren en los mensajes m de tipo SPAM y el denominador representa la probabilidad estimada de que los tokens t se encuentren en ambas categorías, es decir tanto en los mensajes SPAM como en los no SPAM, por lo tanto, el denominador representa la suma de las probabilidades de que los tokens t existan en un mensaje de cualquier tipo, $\mathrm{M_{SPAM}}$ ó $\mathrm{M_{NOSPAM}}$.

Con lo anterior se permite calcular probabilidades con base en determinados antecedentes. El uso del teorema de Bayes en la clasificación de información, solo se puede realizar una vez completado un proceso anterior, que es recolección de información y confección de datos estadísticos; sin estos datos no hay nada que calcular. Los datos estadísticos se obtienen de acuerdo a las palabras (tokens, siendo toda secuencia de caracteres ASCII im-

primibles, lo demás se considera separador de tokens) tomadas del historial de mensajes SPAM y no SPAM.

2.5. Algoritmo basado en tokens

El algoritmo probabilístico para el análisis de mensajes está basado en el teorema de Bayes. El objetivo principal del algoritmo es analizar y clasificar cualquier mensaje que llegue al servidor, definiéndolo como SPAM ó no SPAM, basado en el hecho de que contiene ciertos tokens. El análisis se realiza de acuerdo a la ecuación (2.2).

El algoritmo de análisis para la clasificación de mensajes, trabaja en dos fases. *La primera fase es definida con el proceso de entrenamiento*, el cual consiste en obtener las probabilidades a priori con base en tokens representativos de mensajes de tipo SPAM y no SPAM, la información obtenida es almacenada en los repositorios de información de tokens para su uso posterior. Los mensajes utilizados en esta fase de entrenamiento han sido previamente seleccionados y definidos de manera manual como de tipo SPAM ó no SPAM. *La segunda fase se refiere a la clasificación de mensajes nuevos* que llegan al servidor en SPAM y no SPAM. Para realizar la clasificación de mensajes se aplica el teorema de Bayes con base a los tokens más representativos del mensaje, tomando su información de los repositorios.

La figura 2.3 muestra el repositorio que contiene la información de tokens, dicha información se obtuvo en el proceso de entrenamiento. También muestra el módulo de análisis para la clasificación de los mensajes nuevos que llegan al servidor.

2.5.1. Proceso de entrenamiento

El proceso de entrenamiento consiste en obtener los valores de probabilidad a priori de los tokens de pertenecer a mensajes de tipo SPAM y no SPAM, los valores obtenidos se almacenan en el repositorio de información de tokens. Dicho proceso brinda un grado de confianza al proceso de análisis para la clasificación de mensajes. El proceso de entrenamiento implica preparar, seleccionar y almacenar los datos estadísticos de tokens, para ello se considera que se tiene una muestra de mensajes SPAM y no SPAM clasificados de manera manual.

Con los mensajes no SPAM se forman tokens t_i $(1 \leq i \leq n)$ y se obtiene el valor de ocurrencia de cada token en ellos g_{t_i}, también se obtiene la cantidad de mensajes no SPAM

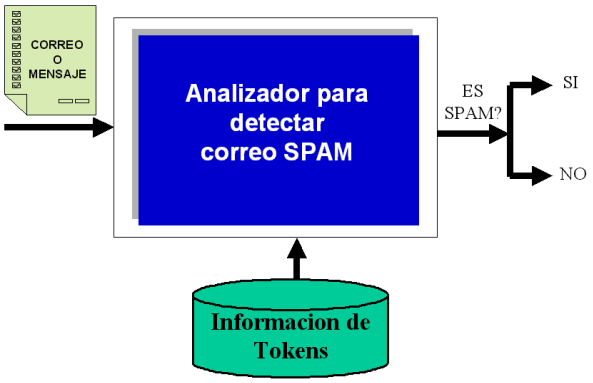

Figura 2.3: Esquema básico de clasificación de mensajes.

utilizados G. De manera similar con los mensajes SPAM se forman tokens t_i $(1 \leq i \leq n)$ y se obtiene el valor de ocurrencia de cada token en ellos b_{t_i}, también se obtiene la cantidad de mensajes SPAM utilizados B.

Considerando los valores de probabilidad obtenidos se hace una selección de los tokens t_i más significativos, definiéndose así aquellos con valor de probabilidad cercano a cero ó cercano a uno, $P_{\mathrm{M_{SPAM}}}(t_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(t_i) > 0.9$. Los tokens seleccionados, así como su información relacionada t_i, b_{t_i}, g_{t_i}, B, G y $P_{\mathrm{M_{SPAM}}}(t_i)$, todo es almacenado en la base de información de tokens. El proceso de entrenamiento (actualización) de la base de información de los tokens se muestra en el algoritmo 1.

Del algoritmo 1 es necesario describir a detalle como se obtiene el valor de probabilidad estimada a priori de los tokens $P_{\mathrm{M_{SPAM}}}(t_i)$ (paso 21). Así mismo, analizar los criterios y el proceso de selección de los tokens t_i más significativos que serán almacenados en la base de información de tokens (paso 24).

Cálculo de la probabilidad a priori de tokens (paso 21)

La probabilidad a priori se refiere al conocimiento inicial que se tiene de los datos en este caso de los tokens, de que sean parte de los mensajes de tipo SPAM y no SPAM.

Se asume que se tiene un conjunto de mensajes conocidos m_i $(0 \leq i \leq n)$ y pueden ser de dos tipos, SPAM ó no SPAM, donde cada categoría representa una hipótesis. La hipótesis $\mathrm{M_{SPAM}}$ representa la categoría de los mensajes m_i que son SPAM y la hipótesis $\mathrm{M_{NOSPAM}}$

Algoritmo 1 Proceso de entrenamiento

Require: Mensajes SPAM y no SPAM, clasificados manualmente
Ensure: Información de los tokens más significativos

1: Inicia **obtención de información de mensajes no SPAM**
2: **for** cada mensaje m_i de tipo no SPAM **do**
3: Formar tokens t_i con el contenido de cada mensaje m_i
4: Por cada token, t_i, obtener:
5: $g_{t_i} \leftarrow$ Ocurrencia del token t_i en el mensaje m_i no SPAM
6: *Guardar el valor de* g_{t_i} de cada token, en la base de datos (B.D.) de tokens
7: **end for**
8: $G \leftarrow$ Número de mensajes m_i de tipo no SPAM
9: Inicia **obtención de información de mensajes SPAM**
10: **for** cada mensaje m_i de tipo SPAM **do**
11: Formar tokens t_i con el contenido de cada mensaje m_i
12: Por cada token, t_i, obtener:
13: $b_{t_i} \leftarrow$ Ocurrencia del token t_i en el mensaje m_i SPAM
14: *Guardar el valor de* b_{t_i} de cada token, en la B.D. de tokens
15: **end for**
16: $B \leftarrow$ Número de mensajes m_i de tipo SPAM
17: Inicia **cálculo de probabilidad a priori de los tokens** t_i **de pertenecer a mensajes** m_i **de tipo SPAM**
18: **for** cada token t_i en la B.D. de información de tokens **do**
19: Buscar y tomar la información de los tokens: t_i, g_{t_i}, b_{t_i}, G, B.
20: Para cada token:
21: *Calcular la probabilidad a priori del token*, $P_{\mathrm{M_{SPAM}}}(t_i)$, de pertenecer a un mensaje m_i de tipo SPAM de acuerdo a la fórmula 2.3.
22: **end for**
23: Inicia **selección y actualización de la información de los tokens** t_i
24: *Seleccionar los tokens* con valor de probabilidad cercana a cero ó cercana a uno, $P_{\mathrm{M_{SPAM}}}(t_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(t_i) > 0.9$.
25: *Actualizar o descartar el token* t_i y toda su información, t_i, b_{t_i}, g_{t_i}, B, G y $P_{\mathrm{M_{SPAM}}}(t_i)$, en la B.D. de tokens

representa la categoría de los mensajes m_i que son no SPAM.

Para obtener la probabilidad de los tokens t_i de ser parte de mensajes de tipo SPAM $P_{\mathrm{M_{SPAM}}}(t_i)$ se utiliza el teorema de Bayes. Para lo cual se define $\mathrm{M_{SPAM}}$ como la categoría de los mensajes m_i de tipo SPAM, b_{t_i} define la frecuencia del token t_i en mensajes de tipo SPAM, g_{t_i} define la frecuencia del token t_i en mensajes de tipo no SPAM, B representa el número total de mensajes SPAM utilizados para el entrenamiento, G representa el número

total de mensajes no SPAM utilizados para el entrenamiento.

Considerando lo anterior, es posible obtener el valor de probabilidad a priori de los tokens t_i de que pertenezcan a un mensaje m_i de tipo SPAM, la cual se representa como $P_{\mathrm{M_{SPAM}}}(t_i)$. Dicho valor se obtiene con la ecuación siguiente:

$$P_{\mathrm{M_{SPAM}}}(t_i) = \frac{P(\mathrm{M_{SPAM}} \cap t_i)}{P(t_i)} = \frac{\frac{b_{t_i}}{B}}{\frac{b_{t_i}}{B} + \frac{g_{t_i}}{G}} \tag{2.3}$$

De la fórmula 2.3, el numerador representa la proporción de la frecuencia de aparición del tokens t_i en mensajes de tipo SPAM con base en todos los mensajes SPAM utilizados para este proceso de entrenamiento. El denominador representa la suma de las proporciones de las frecuencias de aparición del token t_i en los mensajes SPAM y no SPAM con base en todos los mensajes utilizados para el entrenamiento.

De manera similar, si se desea obtener el valor de probabilidad a priori de los tokens t_i de pertenecer a un mensaje m_i de tipo no SPAM $P_{\mathrm{M_{NOSPAM}}}(t_i)$ se utiliza la ecuación siguiente:

$$P_{\mathrm{M_{NOSPAM}}}(t_i) = \frac{P(\mathrm{M_{NOSPAM}} \cap t_i)}{P(t_i)} = \frac{\frac{g_{t_i}}{G}}{\frac{b_{t_i}}{B} + \frac{g_{t_i}}{G}} \tag{2.4}$$

En la fórmula 2.4, el numerador representa la proporción de la frecuencia de aparición del tokens t_i en mensajes de tipo no SPAM con base en todos los mensajes no SPAM utilizados para este proceso de entrenamiento. El denominador representa la suma de las proporciones de las frecuencias de aparición del token t_i en los mensajes SPAM y no SPAM con base en todos los mensajes utilizados para el entrenamiento.

Selección de tokens significativos (paso 24)

La selección de los tokens t_i $(1 \leq i \leq n)$ como parte del proceso de entrenamiento consiste en determinar cuales serán almacenados en el repositorio de información de tokens. Los tokens significativos son aquellos que representan de forma más definida ser parte de mensajes SPAM ó no SPAM.

La selección de tokens t_i que serán actualizados en la base de información de tokens se hace con base en el valor de probabilidad a priori de los tokens t_i, $P_{\mathrm{M_{SPAM}}}(t_i)$, de que

pertenezcan a un mensaje m_i de tipo SPAM. Así mismo, para la selección de los tokens t_i considera su valor de ocurrencia en todos los mensajes.

Los tokens t_i seleccionados son aquellos que tienen un valor de probabilidad cercana a cero ó cercana a uno, $P_{\mathrm{M_{SPAM}}}(t_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(t_i) > 0.9$, de acuerdo a la fórmula 2.3. Dicho criterio de selección se basa en que los tokens t_i con valor de probabilidad cercano a la media ($P_{\mathrm{M_{SPAM}}}(t_i) \approx 0.5$) provocan ambigüedad, es decir hay incertidumbre al clasificar los mensajes como SPAM ó no SPAM. Otra condición que deben cumplir los tokens t_i para ser seleccionados es que deben tener un valor de ocurrencia en todos los mensajes mayor ó igual a tres ($g_{t_i} + b_{t_i} \geq 3$).

El objetivo de este proceso es tener evidencias más precisas que ayuden a obtener mejores resultados en la clasificación de mensajes.

Una ventaja que se tiene al seleccionar los tokens t_i, es que ayuda a prevenir el problema del espacio de variables de las características ya que sería muy grande (del orden de miles de tokens). Otra ventaja es que se atenúa el efecto de considerar la independencia entre los tokens.

En la tabla 2.1 se muestra la lista de tokens t_i más significativos de un mensaje m_i de ejemplo junto con sus probabilidades asociadas de que los tokens t_i aparezcan en un mensaje tipo SPAM $P_{\mathrm{M_{SPAM}}}(t_i)$, y la probabilidad que aparezcan en un mensaje de tipo no SPAM $P_{\mathrm{M_{NOSPAM}}}(t_i)$.

2.5.2. Clasificación de mensajes

Clasificar mensajes como parte del análisis de mensajes, consiste en definir con exactitud cuales mensajes nuevos son SPAM y cuales son no SPAM. Así mismo, el proceso de clasificación de mensajes nuevos tiene como objetivo evitar errores de clasificación (falsos positivos ó falsos negativos).

El proceso de análisis para la clasificación de mensajes nuevos, se realiza con base en los tokens(palabras) de dicho mensaje. Los tokens t_i son considerados como elemento de análisis.

Para poder realizar la clasificación de mensajes m_i nuevos, se asume que se cuenta con un repositorio de información relacionada a tokens, dicha información se obtiene en el proceso de entrenamiento e incluye los siguientes datos: t_i son los tokens más representativos

Tabla 2.1: Tabla de tokens más significativos

Lista de tokens $t_i(\sim 15)$ más significativos		
t_i	$P_{\mathrm{M_{SPAM}}}(t_i)$	$P_{\mathrm{M_{NOSPAM}}}(t_i)$
madam	0.99	0.01
promotion	0.99	0.01
republic	0.99	0.01
shortest	0.047225013	0.952774987
mandatory	0.047225013	0.952774987
standardization	0.07347802	0.92652198
sorry	0.08221981	0.91778019
supported	0.09019070	0.9098093
people's	0.09019070	0.9098093
enter	0.9075001	0.0924999
quality	0.8921298	0.1078702
organization	0.12454646	0.87545354
investment	0.8568143	0.1431857
very	0.14758544	0.85241456
valuable	0.82347786	0.17652214

de los mensajes de entrenamiento (mensajes de tipo SPAM y no SPAM), b_{t_i} es el valor de ocurrencia del token t_i en los mensajes de tipo SPAM, g_{t_i} es el valor de ocurrencia del token t_i en los mensajes de tipo no SPAM, B es la cantidad de mensajes SPAM utilizados para el entrenamiento, G es la cantidad de mensajes no SPAM utilizados para el entrenamiento, $P_{M_{SPAM}}(t_i)$ se refiere a probabilidad estimada del token t_i de pertenecer a un mensaje de tipo SPAM.

El proceso de análisis para clasificar mensajes nuevos está integrado de tres etapas: *preprocesamiento del mensaje m_i nuevo*, *obtención de información de los tokens t_i del mensaje m_i del repositorio*, *cálculo de la probabilidad combinada total de las probabilidades de los tokens t_i*. Con estas tres etapas el proceso de análisis determina si el mensaje es SPAM ó no SPAM.

El proceso de análisis para clasificar mensajes nuevos se describe de la siguiente manera, al llegar un nuevo mensaje m_i al servidor, se extraen de éste algunos tokens t_i. Cada token t_i del mensaje m_i se busca en el repositorio de información de tokens, si el token t_i es encontrado se obtiene toda la información relacionada a él, t_i, b_{t_i}, g_{t_i}, B, G, $P_{\mathrm{M_{SPAM}}}(t_i)$.

De los tokens t_i localizados en el repositorio de información de tokens, se seleccionan

solo los n más significativos. La selección de los tokens se hace con base en el valor de probabilidad estimada del token t_i de que sea parte de un mensaje m_i de tipo SPAM, $P_{\mathrm{M_{SPAM}}}(t_i)$. Los tokens t_i seleccionados son aquellos con valor de probabilidad más cercana a cero ó más cercana a uno, $P_{\mathrm{M_{SPAM}}}(t_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(t_i) > 0.9$.

Con los n tokens t_i más significativos y su información relacionada se realiza el cálculo de la probabilidad combinada total, de acuerdo a la fórmula 2.5. El valor obtenido representa la probabilidad estimada de que el mensaje m_i que contiene los n tokens t_i más significativos sea de tipo SPAM ó no SPAM.

Si el valor de probabilidad combinada total es menor a 0.1, el mensaje m_i es considerado como no SPAM, por otro lado, si el valor de probabilidad es mayor a 0.9, el mensaje m_i, es definido como SPAM.

Algoritmo 2 Algoritmo para clasificar mensajes nuevos

Require: Mensaje nuevo m_i, repositorio con información de tokens
Ensure: Valor de probabilidad combinada total

1: Agrupar el mensaje m_i en tokens t_i
2: **Inicia proceso de recuperación de información de tokens del repositorio**
3: **for** cada token t_i del mensaje m_i **do**
4: Buscar el tokens t_i en el repositorio de información de tokens.
5: Recuperar del repositorio de tokens, toda la información del token t_i encontrado, t_i, b_{t_i}, g_{t_i}, B, G y $P_{\mathrm{M_{SPAM}}}(t_i)$.
6: **end for**
7: **Inicia proceso de selección de los n tokens más significativos**
8: **for** cada tokens t_i recuperado del repositorio de tokens **do**
9: **if** $P_{\mathrm{M_{SPAM}}}(t_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(t_i) > 0.9$ **then**
10: Seleccionar el token t_i
11: **else**
12: Descartar el token t_i
13: **end if**
14: **end for**
15: Ordenar los tokens t_i seleccionados y colocarlos en un lista
16: Considerando que se tienen los tokens t_i ordenados en una lista, tomar los $\frac{n}{2}$ tokens t_i, de cada extremo de la lista
17: **Inicia cálculo de probabilidad combinada total**
18: Calcular la probabilidad combinada total con los n valores de $P_{\mathrm{M_{SPAM}}}(t_i)$ de los tokens t_i más significativos del mensaje m_i, $P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_n)$, de acuerdo a la fórmula 2.5.

En el algoritmo 2 se describe el proceso para clasificar los mensajes nuevos que llegan

al servidor, la clasificación puede ser en SPAM ó no SPAM. La clasificación de mensajes se realiza con base en el valor de probabilidad combinada total (paso 18) de las probabilidades de los n tokens t_i de pertenecer a mensajes de tipo SPAM, dicho valor se calcula como se explica a continuación.

Cálculo de la probabilidad combinada total

A partir de los n tokens t_i más significativos se obtiene el valor de probabilidad, el cual determina la probabilidad estimada del mensaje m_i que contiene los n tokens t_i sea definido como SPAM ó no SPAM. El cálculo se realiza aplicando la fórmula de probabilidad combinada total [5][25]. Sea, $\mathrm{M_{SPAM}}$ la categoría de ser mensajes de tipo SPAM, y sea $P_{\mathrm{M_{SPAM}}}(t_i)$ la probabilidad estimada de que el token t_i sea visto en un mensaje de tipo SPAM. Con lo anterior, si se desea obtener el valor de probabilidad estimada de que el mensaje m_i que contiene n tokens t_i sea de tipo SPAM $P_{M_{SPAM}}(t_1, t_2, ..., t_n)$, se utiliza la siguiente ecuación:

$$P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_n) = \frac{\prod_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(t_i)}{\prod_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(t_i) + \prod_{i=1}^{n} (1 - P_{\mathrm{M_{SPAM}}}(t_i))} \tag{2.5}$$

La fórmula 2.5 representa una combinación de las probabilidades de los n tokens t_i, donde el numerador representa la probabilidad total de los tokens t_i de estar en los mensajes de tipo SPAM y el denominador representa la suma de las probabilidades de los tokens t_i de estar en ambos tipos de mensajes, SPAM y no SPAM.

Si se asume que la exactitud de las predicciones no tienen correlación entre las probabilidades estimadas de los tokens t_i entonces se deduce que la fórmula 2.5 determina la respuesta de que el mensaje es SPAM ó no SPAM.

En la tabla 2.1 se muestran los tokens t_i más significativos de un cierto mensaje m_i de ejemplo junto con sus valores de probabilidad de aparecer en mensajes de tipo SPAM $P_{\mathrm{M_{SPAM}}}(t_i)$ y en mensajes de tipo no SPAM $P_{\mathrm{M_{NOSPAM}}}(t_i)$.

Si se desea clasificar el mensaje m_i del ejemplo en SPAM o en no SPAM, se realiza el análisis con base en la información de los n tokens t_i que éste contiene. Se asume que se tienen los datos de la tabla 2.1 y de acuerdo a la fórmula 2.5 se calcula la probabilidad combinada de los valores de probabilidad de los n tokens t_i. A continuación se describe el proceso de clasificación del mensaje m_i de ejemplo considerando que $n = 4$, es decir, se

clasifica el mensaje m_i de ejemplo con base en los valores de probabilidad de aparecer en un mensaje SPAM de los 4 tokens más significativos, los cuales son aquellos con un valor de $P_{\mathrm{M_{SPAM}}} < 0.1$ ó $P_{\mathrm{M_{SPAM}}} > 0.9$. Los 4 tokens más significativos, tomados de la tabla 2.1, se muestran en la tabla 2.2.

Tabla 2.2: Tabla con los 4 tokens más significativos

i	t_i	$P_{\mathrm{M_{SPAM}}}(t_i)$	$P_{\mathrm{M_{NOSPAM}}}(t_i)$
1	madam	0.99	0.01
2	promotion	0.99	0.01
3	shortest	0.047225013	0.952774987
4	mandatory	0.047225013	0.952774987

Con la información de los 4 tokens t_i seleccionados mostrados en la tabla 2.2, se calcula el valor de probabilidad combinada total de los valores de $P_{\mathrm{M_{SPAM}}}(t_i)$ y $P_{\mathrm{M_{NOSPAM}}}(t_i)$ definido como $P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_n)$, el cálculo se realiza de acuerdo a la fórmula 2.5 y dicho valor determina si el mensaje es SPAM ó no SPAM. Sustituyendo los valores de la tabla 2.2 en la fórmula 2.5, y evaluándola con dichos valores se obtiene lo siguiente,

$$
\begin{aligned}
P_{\mathrm{M_{SPAM}}}(t_1,t_2,t_3,t_4) &= \frac{\prod_{i=1}^{4} P_{\mathrm{M_{SPAM}}}(t_i)}{\prod_{i=1}^{4} P_{\mathrm{M_{SPAM}}}(t_i) + \prod_{i=1}^{4} P_{\mathrm{M_{NOSPAM}}}(t_i)} = \\
&= \frac{0.99{*}0.99{*}0.047225013{*}0.047225013}{0.99{*}0.99{*}0.047225013{*}0.047225013 + 0.01{*}0.01{*}0.952774987{*}0.952774987} = \\
&= \frac{0.0021858208359784506369}{0.0021858208359784506369 + 0.00009077801758528501 69} = \\
&= \frac{0.0021858208359784506369}{0.0022765988535637356538} = \\
&= 0.960125598129339655390066496008383
\end{aligned}
$$

El valor de $P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_4)$ que se obtiene utilizando los 4 tokens de la tabla 2.2, es igual a 0.96012559 y con base en el criterio de clasificación de mensajes, como dicho valor es mayor a 0.9 entonces se concluye que el mensaje m_i del ejemplo es de tipo SPAM. Si se deseara clasificar el mismo mensaje m_i de ejemplo, pero ahora utilizando la información de los 15 tokens t_i de la tabla 2.1, el resultado sería $P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_{15}) = 0.9027$. Con el mismo criterio de clasificación de mensajes y con base en el valor obtenido de

$P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_{15})$, se concluye que el mensajes es SPAM.

Como vemos, la fórmula 2.5 obtiene el valor de probabilidad de que un cierto mensaje m que contiene n tokens t_i sea de tipo SPAM, con base en la existencia de un conjunto de n tokens t_i $(1 \leq i \leq n)$, esto es válido, pero no considera la relación entre los tokens, es decir, no considera la dependencia de existencia de los tokens.

La desventaja que presenta la fórmula 2.5 al evaluar un mensaje sin considerar la dependencia de tokens, es que limita darle un significado al contenido del mensaje. Esto se debe a que el significado de los tokens de manera aislados puede provocar ambigüedad al momento de definirlos como parte de un mensaje SPAM o de un mensaje no SPAM.

2.5.3. Observaciones al algoritmo de análisis

Si se desea analizar un cierto mensaje formado por un conjunto de tokens t_i $(1 \leq i \leq n)$ para ser clasificado como SPAM o no SPAM aplicando la fórmula 2.5, el resultado dependerá del contenido del mensaje y de los tokens t_i seleccionados, específicamente del valor de probabilidad de los tokens t_i de pertenecer a mensajes SPAM $P_{\mathrm{M_{SPAM}}}(t_i)$ y a mensajes no SPAM, $P_{\mathrm{M_{NOSPAM}}}(t_i)$.

Existen mensajes que al ser analizados con base en los valores de probabilidad de los tokens t_i, $P_{\mathrm{M_{SPAM}}}(t_i)$ y $P_{\mathrm{M_{NOSPAM}}}(t_i)$, y al ser combinados dichos valores de acuerdo a la fórmula 2.5 resultan valores cercanos a cero ó cercanos a uno. Es posible obtener valores considerados ideales para clasificar y definir con claridad y certeza al mensaje como SPAM ó no SPAM, dicho valores son aquellos menores a 0.35 ó mayores a 0.65, $P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_n) < 0.35$ ó $P_{\mathrm{M_{SPAM}}}(t_1, t_2, ..., t_n) > 0.65$.

En la figura 2.4 se muestra un rango de porcentajes que va desde 0.35 a 0.65 que no definen con claridad y certeza al mensaje como SPAM ó no SPAM. Cuando el valor de probabilidad combinada total obtenido se ubica en este rango de porcentajes se presentan errores de clasificación del mensaje, es decir, cuando mensajes válidos son definidos como SPAM ó cuando mensajes SPAM son definidos como válidos. Como vemos en la gráfica de la figura 2.4, el rango de incertidumbre para clasificar un mensaje va desde los valores de probabilidad de 0.35 hasta 0.65, considerándose un índice elevado de errores.

La existencia del rango de incertidumbre para clasificar un mensaje con base en su valor de probabilidad combinada total, se presume que se debe a que la fórmula 2.5 no incluye

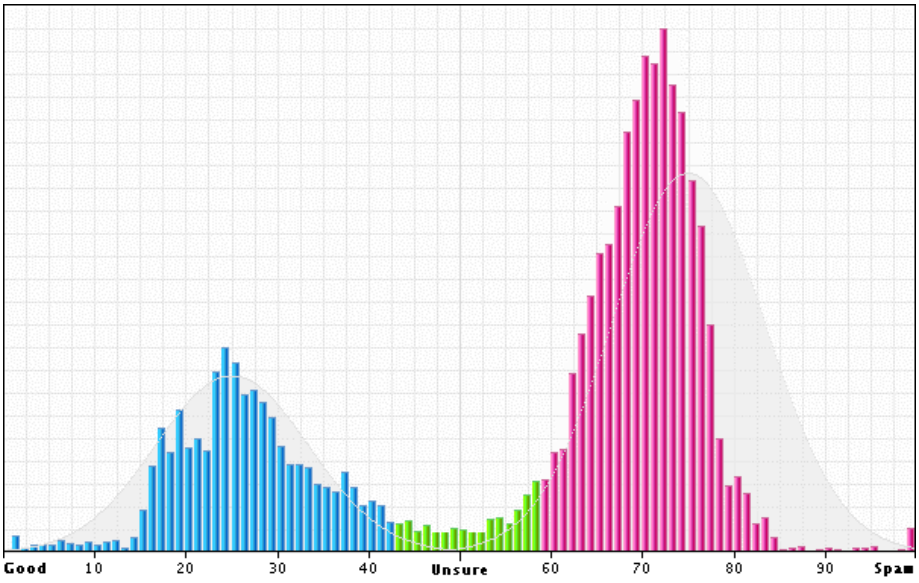

Figura 2.4: Valores de probabilidad combinada total de mensaje.

en su definición, la relación y dependencia de los tokens t_i en el mensaje analizado, por lo tanto, al combinar los valores de probabilidad de los tokens resulta un valor que pertenece a este rango. Por consiguiente, dicha clasificación del mensaje es errónea, afectando con ello, el índice de efectividad de las herramientas de análisis de mensajes que implementan este algoritmo.

2.6. Herramientas de detección de SPAM

Las herramientas existentes más conocidas que permiten hacer un análisis de mensajes para clasificarlos en SPAM ó no SPAM son: SpamProbe y SpamAssassin. A continuación se muestran las características y desventajas de cada una.

SpamProbe

Es una herramienta desarrollada para el análisis y detección de mensajes SPAM, la cual está desarrollada en lenguaje C++, y tiene un 90 % de efectividad. Como unidad de análisis asocia dos tokens y los considera como uno solo. Al realizar el análisis del contenido de los mensajes ignora el código HTML con la finalidad de evitar falsos positivos. Las secciones que considera para el análisis son: Received, Subject, To, From, y Cc. Spamprobe puede

adaptarse y trabajar con Procmail y Maildrop como una herramienta complementaria.

SpamProbe no implementa reglas deterministas (listas negras y blancas). El par de tokens asociados los considera en un solo orden. Utiliza más tiempo para el análisis debido a sus características de análisis. Por todo lo anterior, spamprobe ofrece una efectividad limitada.

SpamAssassin

Es una herramienta que permite analizar y clasificar los mensajes, en SPAM y no SPAM, dicha herramienta está desarrollada en el lenguaje perl, tiene un 95 % de efectividad. En su implementación incluye una combinación de reglas estáticas y dinámicas. Implementa la heurística de listas negras, realiza un análisis de encabezado y cuerpo con heurísticas definidas, también implementa un análisis probabilístico, éste último incluye código HTML para tener mayor variedad de palabras(tokens) para analizar. SpamAssassin permite adaptarse como una herramienta complementaria con Procmail, Qmail, Sendmail, Postfix, entre otras.

Cuando SpamAssassin realiza el análisis presenta una sobrecarga de procesamiento, debido a los módulos de Perl que requiere y que sube a memoria en tiempo de ejecución. Algunas secciones de análisis hacen consultas vía Web (uso de listas negras publicadas en Internet), eso hace lento el análisis y dependiente de una conexión permanente a internet. La base de datos con información de tokens crece rápidamente al entrenar al sistema. El análisis estadístico que implementa se realiza con base en los tokens y a su información, eso limita la efectividad de la herramienta.

SpamProbe y SpamAssassin son dos herramientas que sirven para el análisis de mensajes con la finalidad de clasificarlos en SPAM ó no SPAM. Cada una de ellas presenta limitaciones y desventajas algunas de las más destacadas son: en el caso de SpamProbe no implementa reglas deterministas, es decir, no utiliza listas negras ni listas blancas para la clasificación de mensajes. SpamAssassin está desarrollado en el lenguaje perl, representando esto una desventaja, debido a que al momento de efectuarse el análisis requiere subir a memoria algunos módulos de perl.

En ambas herramientas se implementa un análisis probabilístico de mensajes. Dicho análisis se realiza con base en la información relacionada a tokens. Como característica adicional por parte de SpamProbe, es que asocia dos tokens pero los trata como uno solo y en ese único orden, esta consideración se tiene sin modificar la unidad de análisis token.

En ambas herramientas la clasificación se realiza de acuerdo a la fórmula 2.5, la cual no considera la dependencia de existencia entre los tokens, limitando con ello el significado del contenido del mensaje, por consiguiente, resulta insegura la definición del mensaje, al momento de clasificarlo en SPAM ó no SPAM.

Con base en lo anterior, se tiene la necesidad de una herramienta que implemente un análisis probabilístico para clasificar mensajes en SPAM ó no SPAM, con base en información relacionada a frases (asociación de dos ó más tokens), incluyendo con ello la relación de existencia y dependencia de tokens en un cierto mensaje ofreciendo un mayor significado del contenido del mensaje reflejándose en una clasificación más exacta, evitando errores.

Además de lo anterior, es necesario e importante integrar a la herramienta las características y ventajas que ofrecen las herramientas existentes (spamprobe y spamassassin). Específicamente, que incluya un análisis basado en reglas deterministas, es decir, que haga un análisis del encabezado, específicamente que analice el origen del mensaje con base en listas negras y blancas. Así mismo, que implemente en su análisis heurísticas de palabras y frases clave, logrando con ello analizar el tema (subject) y cuerpo del mensaje.

Capítulo 3

Un algoritmo basado en frases para la clasificación de mensajes

A pesar de que los métodos y las herramientas existentes son buenas, el problema de los mensajes SPAM va en aumento y con características variadas. Esto da margen a mejorar los métodos, o a buscar alternativas antispam para maximizar la efectividad.

En la sección 2.5 se presentó un algoritmo de análisis probabilístico, el cual permite predecir si un mensaje es SPAM, con base en el análisis de sus tokens. El análisis basado en tokens puede verse limitado debido a que cada token se considera independiente de los demás. Como una heurística, se plantea que cada token de mensajes válidos se cuente por dos a fin de evitar los falsos positivos. Sin embargo, esta heurística carece de fundamento estadístico.

Debido a las limitaciones que se tienen al realizar el análisis basado en tokens, se propone una alternativa, tomando como elemento de análisis *frases* del mensaje. Se define como un *token* al agrupamiento de caracteres imprimibles totalmente integrados, donde los tokens se separan usando el caracter de espacio.

Una *frase* se define como la asociación, relación y dependencia de n tokens.

En este capítulo se describe un algoritmo de análisis basado en frases como una extensión y mejora al presentado en la sección 2.5. Este algoritmo consiste en definir a un mensaje como SPAM ó no SPAM, con base en el análisis de sus frases, para lo cual se describen el método de agrupación de los tokens para formar frases, la manera de seleccionarlas y el criterio para definirlas como significativas, así como la fase de entrenamiento y

la fase de clasificación de mensajes, como parte del algoritmo.

Una ventaja que se tiene al realizar la clasificación basada en frases es que se reduce el problema del espacio de variables consideradas para cada token. Otra ventaja es que se atenúa el efecto de considerar la independencia entre los tokens.

3.1. Motivación del análisis basado en frases

Para mostrar la ventaja del análisis de mensajes considerando sus frases se muestra un ejemplo de análisis basado en tokens, así como un ejemplo basado en frases.

El análisis basado en tokens se basa en los datos de la tabla 3.1, con dichos datos se realiza una combinación de las probabilidades de cada token, de acuerdo a la ecuación 2.5, obteniendo un valor de probabilidad combinada de $P_{\mathrm{M_{SPAM}}}(t_1, t_2, t_3, t_4) = 0.8961$. El valor obtenido representa la probabilidad de que el mensaje sea no SPAM.

Tabla 3.1: Tabla con 4 tokens de un mensaje

i	$P_{\mathrm{M_{SPAM}}}(t_i)$	$P_{\mathrm{M_{NOSPAM}}}(t_i)$
1	0.9473	0.0527
2	0.8885	0.1115
3	0.2634	0.7366
4	0.2487	0.7513

Para el análisis basado en frases se utiliza la misma información de la tabla 3.1. A cada token de la tabla se le aplica un ajuste a su valor de probabilidad $P_{\mathrm{M_{SPAM}}}(t_i)$, de acuerdo a la siguiente función.

$$P_{\mathrm{M_{SPAM}}}(t_i)' = a + b * atan(c * P_{\mathrm{M_{SPAM}}}(t_i) + d) \tag{3.1}$$

cuyo comportamiento se muestra en la figura 3.1. Los valores de las constantes dadas son: $a = 0.500011$, $b = 0.401128$, $c = 6.54935$, $d = -3.27474$.

Los valores de probabilidad obtenidos después de aplicar el ajuste se muestran en la tabla 3.2.

El ajuste se realiza para obtener valores más cercanos a cero o más cercanos a uno. Los nuevos valores cercanos a cero se agrupan formando frases, lo mismo se hace con los

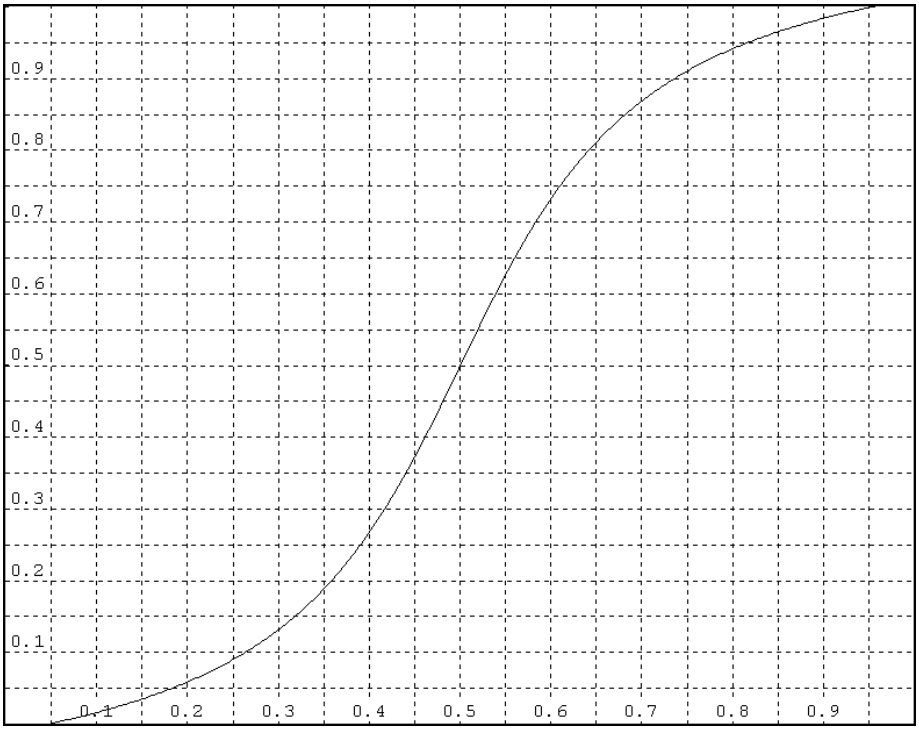

Figura 3.1: Comportamiento de la función 3.1.

Tabla 3.2: Tabla de tokens con probabilidades ajustadas

i	$P_{\mathrm{M_{SPAM}}}(t_i)$	$P_{\mathrm{M_{SPAM}}}(t_i)'$	$P_{\mathrm{M_{NOSPAM}}}(t_i)'$
1	0.9473	0.99815	0.00185
2	0.8885	0.97987	0.02013
3	0.2634	0.09979	0.90021
4	0.2487	0.08895	0.91105

valores cercanos a uno.

Para formar la frase f_1 se agrupan los tokens t_1 y t_2 y al combinar sus valores de probabilidad de acuerdo a la fórmula 3.4 se obtiene una probabilidad estimada de $P_{\mathrm{M_{SPAM}}}(f_1) = 0.98901$. De la misma manera para formar la frase f_2 se agrupan los tokens t_3 y t_4 y al combinar sus valores de probabilidad de acuerdo a la fórmula 3.4 se obtiene una probabilidad estimada de $P_{\mathrm{M_{SPAM}}}(f_2) = 0.09437$.

Con los valores de probabilidad de las nuevas frases $P_{\mathrm{M_{SPAM}}}(f_1)$ y $P_{\mathrm{M_{SPAM}}}(f_2)$ se obtiene el valor de probabilidad combinada total de las frases $P_{\mathrm{M_{SPAM}}}(f_1, f_2)$, de acuerdo a la

fórmula 2.5 como se muestra a continuación,

$$P_{\mathrm{M_{SPAM}}}(f_1, f_2) = \frac{\prod_{i=1}^{2} P_{\mathrm{M_{SPAM}}}(f_i)}{\prod_{i=1}^{2} P_{\mathrm{M_{SPAM}}}(f_i) + \prod_{i=1}^{2} P_{\mathrm{M_{NOSPAM}}}(f_i)} =$$
$$= \frac{0.98901 * 0.09437}{0.98901 * 0.09437 + 0.01099 * 0.90563} =$$
$$= \frac{0.0933328737}{0.0933328737 + 0.0099528737} = \frac{0.0933328737}{0.1032857474} = 0.90363749$$

Con los resultados obtenidos en el análisis basado en tokens, así como en el basado en frases, observamos que en éste último se obtiene un mejor resultado para tomar una decisión de clasificación del mensaje, evitando que la clasificación sea ambigua. Por lo tanto, es posible que el análisis de mensajes basado en frases ayude a mejorar la eficiencia de clasificación de mensajes. Se plantea el uso de este algoritmo como complemento al que se basa en tokens independientes descrito en el capítulo 2.

3.2. Descripción del algoritmo basado en frases

El algoritmo extendido propuesto realiza el análisis y clasificación de mensajes haciendo uso del teorema de Bayes.

De forma similar al algoritmo del capítulo 2, el algoritmo está integrado por dos fases. *La primera fase se refiere al proceso de entrenamiento*, el cual consiste en obtener del historial de mensajes información relacionada a frases, donde dichos mensajes han sido previamente seleccionados de manera manual. Así mismo, se obtiene el valor de probabilidad estimada a priori de las frases formadas, de acuerdo a la ecuación 3.2. Todo lo obtenido se almacena en el repositorio de información de frases.

La segunda fase consiste en clasificar mensajes nuevos en SPAM o no SPAM, de acuerdo a la ecuación 3.8 de la página 43 con base en la información de sus frases más significativas, dicha información se obtuvo en la primera fase.

3.2.1. Formación de frases

El elemento de análisis utilizado en este algoritmo son las frases. A continuación se describe el proceso de formación de frases con los tokens de un mensaje.

Los tokens más significativos de un mensaje son aquellos con valor de probabilidad de pertenecer a un mensaje SPAM cercano a cero ó cercano a uno, específicamente, $P_{\mathrm{M_{SPAM}}}(t_i) < 0.35$ ó $P_{\mathrm{M_{SPAM}}}(t_i) > 0.65$.

Considerando los tokens más significativos se forman dos conjuntos, uno integrado de aquellos con $P_{\mathrm{M_{SPAM}}}(t_i) < 0.35$ y el otro de aquellos con $P_{\mathrm{M_{SPAM}}}(t_i) > 0.65$.

Para obtener frases de tamaño m tokens se realiza una combinación con los tokens de cada conjunto. El proceso de formación de frases es el mismo para ambos conjuntos. Entre las frases resultantes no existen tokens repetidos en una misma frase, tampoco se repiten frases con los mismos tokens en distinto orden.

Para describir el proceso de formación de frases se define lo siguiente,

Sea $T = \{t_1, t_2, ..., t_n\}$ un conjunto de tokens, se forma una frase para cada subconjunto U de T, diferente del vació $U \subset T$, $U \neq \emptyset$. Finalmente las frases se obtienen con todas las combinaciones posibles de k tokens sin repeticiones tomados de T.

$$\bigcup_{k=2}^{n-1} C\binom{n}{k}$$

Por ejemplo, si se considera el conjunto T=$\{a, b, c, d\}$, formando frases de $k = 2$ elementos se obtienen las siguientes: $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$, $\{c, d\}$. Son seis los subconjuntos resultantes.

Al realizar la formación de frases pueden resultar frases muy grandes, cuando $k = n-1$, es por ello que en nuestro caso se recomienda usar frases de tamaño $k = 2$ ó $k = 3$.

3.2.2. Proceso de entrenamiento

El proceso de entrenamiento (primera fase) tiene como objetivo mantener actualizada la información relacionada a frases en el repositorio. La importancia de este proceso se debe a que la clasificación de mensajes se hace con base en dicha información.

La información de frases que se obtiene en este proceso consiste de el número de ocurrencias de la frase en mensajes SPAM, el número de ocurrencias en no SPAM, así como la frase misma y su probabilidad a priori de que pertenezca a mensajes SPAM. Como información general se obtiene el número de correos SPAM y no SPAM utilizados en este proceso.

Para lograr el proceso de entrenamiento, se asume que se tiene un conjunto de mensajes m_i $(1 \leq i \leq n)$ de ambos tipos, SPAM y no SPAM seleccionados de manera manual.

El proceso de entrenamiento se describe de la siguiente manera, los mensajes de tipo SPAM se agrupan en tokens, los tokens se asocian y se combinan formando frases, de cada frase se obtiene su valor de ocurrencia en los mensajes SPAM. Algo similar se realiza con los mensajes no SPAM.

Con la información obtenida de los mensajes y aplicando el teorema de Bayes se obtiene el valor de probabilidad estimada a priori de cada frase de pertenecer a un mensaje de tipo SPAM, definida como $P_{\mathrm{M_{SPAM}}}(f_i)$, dicho cálculo se realiza de acuerdo a la fórmula 3.2.

Para almacenar las frases y toda su información es necesario seleccionar aquellas consideradas como más significativas. El criterio de selección de frases, es de acuerdo a su valor de probabilidad, $P_{\mathrm{M_{SPAM}}}(f_i)$. El pseudocódigo del proceso de entrenamiento se muestra en el algoritmo 3.

Selección de frases de los mensajes

La selección de las frases de un mensaje inicia en el momento en que únicamente se consideran los tokens más significativos para formar las frases.

Las frases de un mensaje son aquellas que resultan de la combinación de los tokens con probabilidad $P_{\mathrm{M_{SPAM}}}(t_i) < 0.35$ ó $P_{\mathrm{M_{SPAM}}}(t_i) > 0.65$.

Las frases formadas de un mensaje pueden ser demasiadas, es por ello que se hace una selección de frases de acuerdo a su probabilidad a priori, el cálculo de dicha probabilidad se describe a continuación.

Cálculo de la probabilidad a priori de las frases

La probabilidad a priori de frases se refiere a la información inicial que se tiene de que sean parte de los mensajes SPAM y no SPAM.

Algoritmo 3 Proceso de entrenamiento

Require: Mensajes SPAM y no SPAM clasificados manualmente
Ensure: Información relacionada a las frases más significativas

Inicia **obtención de información de mensajes SPAM**
for cada mensaje m_i de tipo SPAM **do**
  Formar tokens t_i con el contenido de cada mensaje m_i
  Agrupar tokens t_i para formar frases f_i
  Combinar los tokens t_i de las frases f_i
  Por cada frase f_i formada obtener:
  $b_{f_i} \leftarrow$ Ocurrencia de la frase f_i en el mensaje m_i SPAM
  Guardar el valor de b_{f_i} de cada frase f_i en la B.D. de frases
end for
$B \leftarrow$ Número de mensajes m_i de tipo SPAM
Inicia **obtención de información de mensajes no SPAM**
for cada mensaje m_i de tipo no SPAM **do**
  Formar tokens t_i con el contenido de cada mensaje m_i
  Agrupar tokens t_i para formar frases f_i
  Combinar los tokens t_i de las frases f_i
  Por cada frase f_i formada obtener:
  $g_{f_i} \leftarrow$ Ocurrencia de la frase f_i en el mensaje m_i no SPAM
  Guardar el valor de g_{f_i} de cada frase f_i en la B.D. de frases
end for
$G \leftarrow$ Número de mensajes m_i de tipo no SPAM
Inicia el **cálculo de la probabilidad a priori de las frases** f_i de pertenecer a mensajes m_i de tipo SPAM
for cada frase f_i del mensaje m_i **do**
  Buscar y recuperar la información de cada frase f_i, g_{f_i}, b_{f_i}, G, B
  Para cada frase f_i:
  Calcular la probabilidad a priori de la frase, $P_{M_{SPAM}}(f_i)$, de pertenecer a mensajes de tipo SPAM de acuerdo a la fórmula 3.2
end for
Inicia **selección y actualización de la información de las frases** f_i
Seleccionar las frases f_i con valor de probabilidad, $P_{\mathrm{M_{SPAM}}}(f_i)$, cercano a cero ó cercano a uno, $P_{\mathrm{M_{SPAM}}}(f_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(f_i) > 0.9$.
Actualizar o descartar la frase f_i y toda su información, f_i, g_{f_i}, b_{f_i}, G, B, $P_{\mathrm{M_{SPAM}}}(f_i)$, en el repositorio de frases

Para obtener el valor de probabilidad a priori de las frases f_i formadas se utiliza como base el teorema de Bayes, para lo cual se define lo siguiente: Sean $\mathrm{M_{SPAM}}$, $\mathrm{M_{NOSPAM}}$, B,

G, de manera similar que lo descrito en el capítulo 2 sección 2.5, y sean b_{f_i} la frecuencia de la frase f_i en mensajes SPAM, g_{f_i} la frecuencia de la frase f_i en mensajes no SPAM. También se define a t_i como los tokens de los mensajes, y sea f_i las frases formadas con los tokens t_i. Con lo anterior se obtiene el valor de probabilidad estimada a priori de que la frase f_i sea parte de mensajes SPAM $P_{\mathrm{M_{SPAM}}}(f_i)$ con la siguiente ecuación.

$$P_{\mathrm{M_{SPAM}}}(f_i) = \frac{P(\mathrm{M_{SPAM}} \cap f_i)}{P(f_i)} = \\ = \frac{P(f_i|\mathrm{M_{SPAM}}) * P(\mathrm{M_{SPAM}})}{P(f_i|\mathrm{M_{SPAM}}) * P(\mathrm{M_{SPAM}}) + P(f_i|\mathrm{M_{NOSPAM}}) * P(\mathrm{M_{NOSPAM}})} \tag{3.2}$$

donde, $P(f_i|\mathrm{M_{SPAM}})$ se refiere a la probabilidad estimada de que la frase f_i sea parte de un SPAM, $P(\mathrm{M_{SPAM}})$ representa la probabilidad estimada de que un cierto mensaje aleatorio sea SPAM (ver fórmula 3.7), $P(f_i|\mathrm{M_{NOSPAM}})$ se refiere a la probabilidad estimada de que la frase f_i sea parte de un no SPAM, $P(\mathrm{M_{NOSPAM}})$ representa la probabilidad estimada de que un cierto mensaje aleatorio sea no SPAM(ver fórmula 3.6).

Para obtener el valor $P(f_i|\mathrm{M_{SPAM}})$ de la fórmula 3.2 se consideran las definiciones de t_i y f_i, descritas anteriormente.

$$P(f_i|\mathrm{M_{SPAM}}) = \frac{\frac{\sum_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(t_i)'}{n}}{\frac{\sum_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(t_i)'}{n} + \frac{\sum_{i=1}^{n} P_{\mathrm{M_{NOSPAM}}}(t_i)'}{n}} \tag{3.3}$$

donde el numerador es una combinación de las probabilidades de los n tokens de la frase y representa la probabilidad de que la frase sea parte de un SPAM, el denominador es la suma de dos combinaciones de las probabilidades de los n tokens de la frase, una combinación representa la probabilidad de que la frase sea parte de mensajes SPAM y la otra combinación representa de que sea parte de mensajes no SPAM.

La combinación de las probabilidades de n tokens que integran la frase se logra obteniendo su promedio con la siguiente ecuación.

$$Promedio_{f_i} = \frac{\sum_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(t_i)'}{n} \tag{3.4}$$

Para poder realizar la combinación es necesario que a cada token se le aplique un

ajuste a su valor de probabilidad con la finalidad de obtener valores más cercanos a cero o más cercanos a uno. El ajuste consiste en evaluar cada valor $P_{\mathrm{M_{SPAM}}}(t_i)$, con la siguiente función.

$$P_{\mathrm{M_{SPAM}}}(t_i)' = a + b * atan(c * P_{\mathrm{M_{SPAM}}}(t_i) + d) \tag{3.5}$$

cuyo comportamiento se muestra en la figura 3.1. Los valores de las constantes usadas son: $a = 0.500011$, $b = 0.401128$, $c = 6.54935$, -3.27474.

Los valores de $P_{\mathrm{M_{SPAM}}}(t_i)$ y $P_{\mathrm{M_{NOSPAM}}}(t_i)$ de la fórmula 3.3 representan las probabilidades a priori de los tokens t_i de pertenecer a mensajes SPAM y a no SPAM, respectivamente. Ambos valores han sido obtenidos en el proceso de entrenamiento del algoritmo basado en tokens de la sección 2.5.

Para obtener los valores de $P(\mathrm{M_{SPAM}})$ y $P(\mathrm{M_{NOSPAM}})$ de la fórmula 3.2 se define lo siguiente, sea G el número total de mensajes no SPAM y sea B el número total de mensajes SPAM, ambos utilizados en el proceso de entrenamiento. Los valores de $P(\mathrm{M_{SPAM}})$ y $P(\mathrm{M_{NOSPAM}})$ se obtienen con las ecuaciones siguientes.

$$P(\mathrm{M_{NOSPAM}}) = \frac{G}{B+G} \tag{3.6}$$

$$P(\mathrm{M_{SPAM}}) = \frac{B}{B+G} \tag{3.7}$$

Para obtener el valor de probabilidad estimada a priori de que la frase f_i formada por los tokens t_i pertenezca a mensajes de tipo SPAM, se evalúan las ecuaciones 3.3, 3.6, 3.7 y 2.3, 2.4, estas últimas se refieren al cálculo de probabilidad de que un token pertenezca a mensajes de tipo SPAM y a mensajes de tipo no SPAM, respectivamente, y todas se sustituyen en la ecuación 3.2.

Selección de frases significativas

La selección de frases significativas consiste en determinar cuales evidencias permiten clasificar los mensajes con mayor certeza en SPAM o no SPAM, dichas frases serán almacenadas.

Las frases significativas representan características más propias de mensajes SPAM ó mensajes no SPAM de acuerdo a su valor de probabilidad a priori de que pertenezcan a mensajes de tipo SPAM $P_{\mathrm{M_{SPAM}}}(f_i)$, dichas frases se utilizan en el proceso de clasificación de mensajes.

Una *frase significativa* es aquella con un valor de probabilidad cercano a cero ó cercano a uno, $P_{\mathrm{M_{SPAM}}}(f_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(f_i) > 0.9$.

Dicho criterio es válido para la selección de frases debido a que las frases f_i con valor de probabilidad cercano a la media ($P_{\mathrm{M_{SPAM}}}(f_i) \approx 0.5$) provocan ambigüedad al efectuarse la clasificación de mensajes.

3.2.3. Clasificación de mensajes

Clasificar mensajes nuevos que lleguen al servidor (segunda fase) consiste en definir con certeza los que son SPAM y no SPAM. La clasificación debe resultar libre de errores.

La clasificación se realiza con base en información relacionada a las frases que integran los mensajes, dicha información fue almacenada en el proceso de entrenamiento (primera fase).

Considerando la información de frases, se realiza la clasificación de mensajes nuevos en SPAM ó no SPAM. El proceso de clasificación está integrado de tres etapas: *preprocesamiento del mensaje* m_i, *obtención de información de las frases* f_i *del mensaje* m_i *del repositorio* y *cálculo de probabilidad combinada de las frases* f_i *más significativas*, ésta última determina si el mensaje es SPAM ó no SPAM.

El proceso de clasificación de mensajes nuevos se describe de la siguiente manera, al llegar un nuevo mensaje al servidor se forman tokens t_i, los tokens t_i se agrupan y se combinan para formar frases f_i. Cada frase f_i formada se busca en el repositorio de información de frases, si la frase f_i es encontrada, se obtiene toda su información: f_i, b_{f_i}, g_{f_i}, B, G, $P_{\mathrm{M_{SPAM}}}(f_i)$.

De las frases f_i localizadas en el repositorio de información de frases, se seleccionan solo las n más significativas, de acuerdo a como se describió en la sección 3.2.2. La selección se realiza con base en su valor de probabilidad estimada a priori de las frases de que sean parte de mensajes de tipo SPAM $P_{\mathrm{M_{SPAM}}}(f_i)$.

Con la información de las n frases seleccionadas se realiza el cálculo de probabilidad

combinada de todas las frases, de acuerdo a la fórmula 3.8, la cual determina la probabilidad estimada de que el mensaje que contiene las n frases f_i más significativas sea de tipo SPAM ó no SPAM.

$$P_{\mathrm{M_{SPAM}}}(f_1, f_2, ..., f_n) = \frac{\prod_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(f_i)}{\prod_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(f_i) + \prod_{i=1}^{n}(1 - P_{\mathrm{M_{SPAM}}}(f_i))} \tag{3.8}$$

En la fórmula 3.8, el numerador representa la probabilidad total de las frases de pertenecer a mensajes SPAM y el denominador representa la probabilidad total de las frases f_i pertenecer a mensajes de ambos tipos, SPAM y no SPAM.

El valor de $P_{\mathrm{M_{SPAM}}}(f_1, f_2, ..., f_n)$ determina la probabilidad estimada de que el mensaje que contiene las n frases más significativas sea de tipo SPAM ó no SPAM. De manera similar que en la selección de frases, ahora un valor de $P_{\mathrm{M_{SPAM}}}(f_1, f_2, ..., f_n) < 0.1$ se interpretará como un mensaje no SPAM y un valor de $P_{\mathrm{M_{SPAM}}}(f_1, f_2, ..., f_n) > 0.9$ se interpretará como un mensaje SPAM. En el algoritmo 4 se describe el proceso para clasificar los mensajes nuevos con base en su valor de probabilidad combinada total (paso 20) de las n frases.

Un método alternativo para calcular la probabilidad combinada es mediante el meta-análisis conocido como *método de Fisher*[28]. Este método utiliza la función inversa Chi-Cuadrada para obtener la probabilidad combinada de todas las frases.

Asumiendo que se tiene un conjunto de probabilidades de las n frases f_i $P_{\mathrm{M_{SPAM}}}(f_i)$ $(1 \leq i \leq n)$, se calcula el doble del logaritmo del producto de las mismas, se considera una distribución Chi cuadrada con $2n$ grados de libertad y se calcula la probabilidad combinada total. Finalmente, se define lo que se denomina "hipótesis nula"para poder trabajar con la estadística aplicada al análisis de mensajes para detectar aquellos que son SPAM. El objetivo es rechazar la hipótesis nula para obtener la probabilidad de que un mensaje sea no SPAM. Lo anterior se puede calcular con la siguiente ecuación.

$$S = P_{\mathrm{M_{SPAM}}}(f_1, f_2, .., f_n) = C^{-1}(-2ln\prod_{i=1}^{n} P_{\mathrm{M_{SPAM}}}(f_i), 2n) \tag{3.9}$$

donde: C^{-1} es la función inversa Chi cuadrada con $2n$ grados de libertad.

Así también, se puede utilizar para calcular la probabilidad de que un mensaje sea no SPAM como se muestra en la ecuación 3.10.

Algoritmo 4 Proceso de clasificación de mensajes nuevos

Require: Mensaje nuevo m_i, repositorio de información de frases
Ensure: Valor de probabilidad combinada total del mensaje m_i

Inicia el preprocesamiento del mensaje m_i
Agrupar el mensaje m_i en tokens t_i
Agrupar y combinar los tokens t_i para formar frases f_i
Inicia la recuperación de información de frases, del repositorio
for cada frase f_i del mensaje m_i **do**
    Buscar la frase f_i en el repositorio de información de frases
    Si la frase f_i fue encontrada, entonces se recupera toda su información, f_i, b_{f_i}, g_{f_i}, B, G, $P_{\mathrm{M_{SPAM}}}(f_i)$
end for
Inicia la selección de las n frases f_i más significativas
for cada frase f_i del mensaje m_i **do**
    if $P_{\mathrm{M_{SPAM}}}(f_i) < 0.1$ ó $P_{\mathrm{M_{SPAM}}}(f_i) > 0.9$ **then**
        Seleccionar la frase f_i
    else
        Descartar la frase f_i
    end if
end for
Ordenar las frases f_i seleccionadas y colocarlas en una lista
Considerando que se tienen las frases ordenadas en una lista, tomar las $\frac{n}{2}$ frases f_i de cada extremo de la lista
Inicia cálculo de probabilidad combinada total
Calcular el valor de probabilidad combinada total con los n valores de $P_{\mathrm{M_{SPAM}}}(f_i)$ de las n frases f_i seleccionadas del mensaje m_i, $P_{\mathrm{M_{SPAM}}}(f_1, f_2, ..., f_n)$ de acuerdo a la fórmula 3.8.

$$G = P_{\mathrm{M_{NOSPAM}}}(f_1, f_2, .., f_n) = C^{-1}(-2ln\prod_{i=1}^{n}(1 - P_{\mathrm{M_{SPAM}}}(f_i)), 2n) \tag{3.10}$$

Una vez obtenidos ambas probabilidades, si se desea obtener la probabilidad de que un mensaje sea SPAM ó no SPAM es necesario obtener un valor indicador para ello se define lo siguiente, sea I el valor indicador a favor de ser SPAM ó no SPAM, sea $P_{\mathrm{M_{NOSPAM}}}(f_1, f_2, .., f_n)$ la probabilidad de que el mensaje sea no SPAM y sea $P_{\mathrm{M_{SPAM}}}(f_1$,-$f_2, .., f_n)$ la probabilidad de que el mensaje sea SPAM. El valor indicador se obtiene con la siguiente ecuación.

$$I = \frac{1 + S - G}{2} \tag{3.11}$$

Con el valor indicador I es posible definir a los mensajes en SPAM ó no SPAM, si el valor indicador I es cercano a cero entonces las evidencias apuntan a la conclusión de que el mensaje m_i es no SPAM, por otro lado, si el valor indicador I es cercano a uno entonces las evidencias apuntan a la conclusión de que el mensaje m_i es SPAM.

Capítulo 4

Sistema de análisis y filtraje de SPAM

Ante el problema de la presencia de correos SPAM en los servidores, es necesario disminuirlo y abatirlo. En el capítulo 3 se mostró una alternativa de solución para mejorar la eficiencia del algoritmo de clasificación basado en el análisis estadístico y probabilístico.

En el presente capítulo se describe el desarrollo de la propuesta del capítulo 3. Dicha solución se muestra como una extensión al funcionamiento básico del procesador de correos del sistema operativo Linux (procmail). También se muestra el contexto organizacional del sistema, así como los elementos que interactúan en el ambiente, entre otros aspectos se muestra la perspectiva de los datos que manejan estos elementos y el aspecto funcional del sistema desarrollado.

La funcionamiento básico del sistema desarrollado es el siguiente, al arribar un mensaje se le extraen sus características, tales características son tomadas y analizadas para detectar mensajes SPAM, al término del análisis se decide si el mensaje es SPAM o no. La figura 4 muestra al sistema de análisis en su forma básica.

El entorno del sistema desarrollado se origina de su integración al procesador de correos local (procmail) de Linux, de esta manera se tiene que al momento de leerse un correo por la entrada estándar se toma alguno de los archivos de reglas, */etc/procmailrc* o *$HOME/.procmailrc*, en los cuales se indica qué debe hacerse con el correo, pudiéndose alterar el proceso de entrega en función de una serie de reglas. Dichas reglas se refieren al análisis del contenido del mensaje en sí.

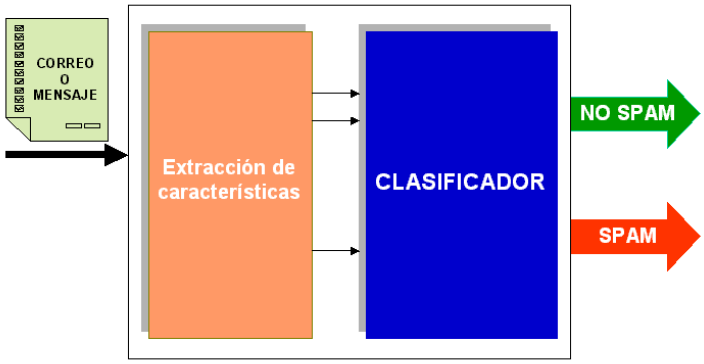

Figura 4.1: Esquema general de un sistema clasificador de mensajes SPAM.

Escenario básico para el sistema desarrollado

1. Se lee un mensaje de la entrada estándar.
2. Se consulta uno de los archivos */etc/procmailrc* ó *$HOME/.procmailrc*.
3. Por medio de parámetros definidos en el archivo de reglas, se activa el sistema propuesto para analizar dicho mensaje.
4. Con base en el análisis, procmail permite tratar el correo que nos llegue o almacenarlo en algún buzón específico.

4.1. Arquitectura del sistema antispam

Para nuestro propósito, una de las características más importantes de procmail es la posibilidad de ejecutar un programa de usuario. El sistema está desarrollado como una extensión a la funcionalidad de *procmail*, la figura 4.1 muestra la extensión de la funcionalidad al momento de que un mensaje arriba al servidor.

El sistema de análisis y clasificación de mensajes tiene la funcionalidad que a continuación se describe. Al momento que un mensaje llega al servidor es atendido por el procesador de correos local (procmail). A través del archivo de reglas indica la ejecución del sistema de análisis desarrollado, enviándole el mensaje original. El mensaje es recibido

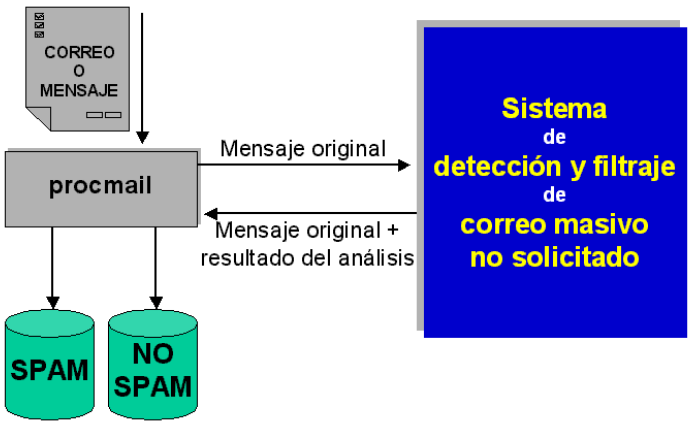

Figura 4.2: Esquema del sistema y su entorno.

por el sistema, el cual realiza el análisis en busca de la conclusión de que sea SPAM o no SPAM. Al término del análisis se regresa el mensaje original más el resultado del análisis. Con base en el resultado procmail decide en dónde depositar el mensaje, si en el buzón de mensajes SPAM o en el buzón normal del usuario.

Veamos la arquitectura del sistema de análisis y clasificación de correo SPAM, desde la perspectiva de cómo esta integrado, qué datos maneja y su funcionamiento.

El sistema tiene una organización por módulos, donde cada módulo se encarga de un nivel de análisis. Los módulos del sistema son: *módulo a nivel de listas de direcciones conocidas, módulo de palabras clave en el tema y cuerpo del mensaje, módulo de frases clave en el tema y cuerpo del mensaje, módulo de análisis estadístico y probabilístico del mensaje*. La arquitectura general se muestra en la figura 4.3.

La arquitectura del sistema integra tres grupos de elementos, dichos grupos son: *elementos del entorno del sistema*, *elementos referente a los repositorios de información* y *elementos que integran al sistema mismo*.

Los elementos que de manera directa interactuan con el sistema son:

Mensaje es el correo que se lee de la entrada estándar para ser analizado.

Procmail es el procesador de correos de Linux y por medio del archivo de reglas se manda llamar al sistema de análisis de mensajes.

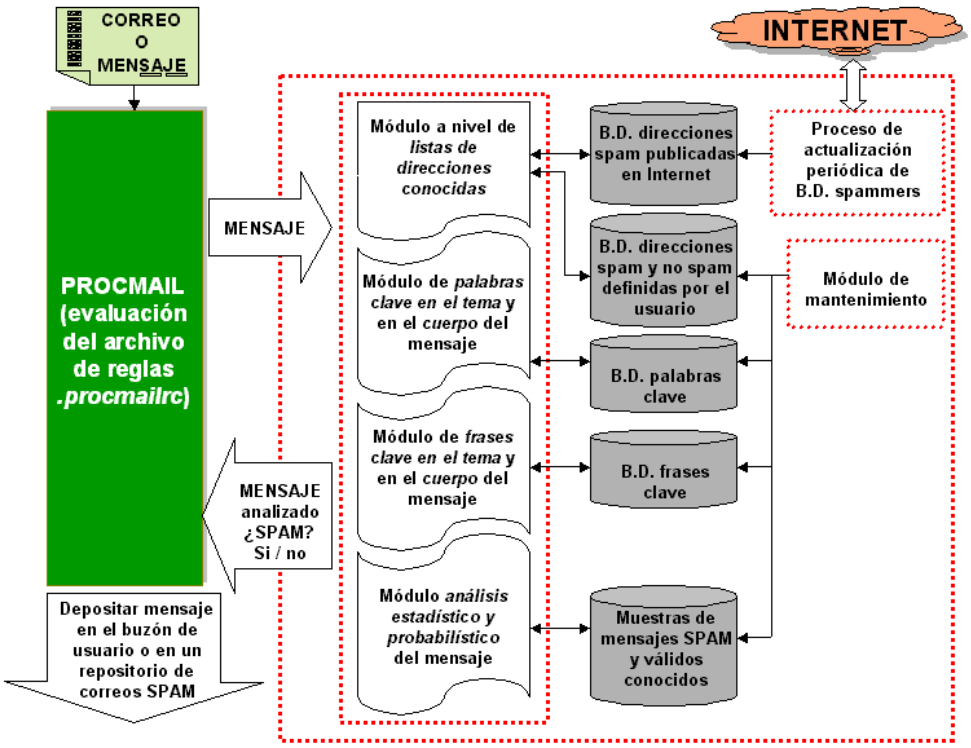

Figura 4.3: Arquitectura general del sistema.

Servicio de Internet a través de éste se realiza el proceso de actualización periódica de la lista de spammers, dicho proceso se conecta directamente a Internet para actualizar la información.

Los elementos que representan los repositorios de información son:

Lista de direcciones SPAM publicadas en Internet contiene direcciones SPAM conocidas, dichas direcciones son tomadas de sitios públicos en Internet.

Lista de direcciones SPAM se refiere a las direcciones que el usuario define como emisores de correo SPAM.

Lista de direcciones no SPAM incluye direcciones que a criterio del usuario son seguras como emisores de correo válido para el usuario.

Lista de palabras clave almacena las palabras consideradas por el usuario como características de correo SPAM, así como el valor asociado a la palabra valorándola como parte de un mensaje SPAM.

Lista de frases clave contiene las frases que a criterio del usuario son definidas como propias de un mensaje SPAM.

Historial de mensajes SPAM y no SPAM se refiere a los mensajes que el usuario ha seleccionado y clasificado como mensajes SPAM o no SPAM, de acuerdo a su criterio, a su experiencia y observación.

Los módulos que integran el sistema principal de análisis son los siguientes:

Módulo de análisis a nivel de listas de direcciones conocidas evalúa si un remitente está previamente identificado como emisor de SPAM o como un usuario válido, para colocarlo en el buzón de mensajes SPAM o en el buzón normal del usuario, respectivamente.

Módulo de análisis de palabras clave en el tema y cuerpo del mensaje analiza el asunto y cuerpo del mensaje en busca de palabras definidas como SPAM.

Módulo de análisis de frases clave en el tema y cuerpo del mensaje busca frases propias de mensajes SPAM, en el tema y cuerpo del mensaje, al encontrar alguna de ellas el mensaje se define como SPAM.

Módulo de análisis estadístico y probabilístico del mensaje realiza el análisis con base en el teorema de Bayes para determinar si un mensaje es SPAM o no SPAM.

4.1.1. Estructura de directorios y archivos usada por el sistema

Los elementos y módulos que integran el sistema son: *elementos de procmail*, *elementos del sistema de análisis* y *los buzones de correo*.

Los elementos del sistema de análisis, así como los buzones son los elementos más importantes. A continuación se muestra como están organizados, así como su ubicación.

Estructura de directorios y archivos

Subdirectorio	Archivos
Elementos del Sistema	
$*HOME/antispam/*	
	antiSPAM.exe
	listablanca
	listanegrausr
	listanegraweb
	palabrasclave
	frasesclave
	html2text.exe
	updatelistanegra.exe
Buzones de correo del sistema	
$*HOME/mail/*	
	LISTAblancausr
	LISTAnegrausr
	LISTAnegraweb
	PALABRASclave
	FRASESclave
	probable-spam
	spam

Elementos de Procmail

Procmail es el procesador de correos local que se ejecuta al momento que llega un correo al servidor de correos, el procesamiento del correo se hace de acuerdo a un archivo de reglas, en el cual se define que hacer con el mensaje.

Ubicación de los archivos de reglas

Archivo de reglas	Descripción
/etc/procmailrc	Las reglas colocadas en este archivo aplican para todos los usuarios registrados en este servidor.
$HOME/.procmailrc	Las reglas de este archivo solo aplican para el usuario que tenga este archivo en su home.

Elementos del sistema propuesto

El sistema de análisis esta integrado de cinco módulos, donde cada uno requiere de ciertos elementos para efectuar el análisis.

Los elementos que requieren los módulos del sistema son: *lista blanca definida por el usuario*, *lista negra definida por el usuario*, *lista negra tomada de Internet*, *lista de palabras clave*, *lista de frases clave*.

A continuación se describen los elementos necesarios para los módulos del sistema de análisis.

Elementos del sistema ubicados en $\$HOME/antispam/$

Elemento del sistema	Descripción
antiSPAM.exe	Se refiere al programa que realiza el proceso de análisis.
listablanca	Contiene las direcciones de usuarios válidos.
listanegrausr	Incluye las direcciones definidas por el usuario como emisores de SPAM.
listanegraweb	Incluye las direcciones tomadas desde Internet, consideradas como emisores de SPAM.
palabrasclave	Incluye las palabras que el usuario define como características de correos SPAM.
frasesclave	Incluye las frases que el usuario define como características propias de correos SPAM.
detectaFrases.exe	Realiza el análisis basado en frases.
html2text.exe	Se refiere a la herramienta que permite quitar el código html a los mensajes.
updatelistanegra.exe	A través de este programa, se actualiza la lista *listanegraweb*.

Los buzones de correo

Una vez que se ha realizado el análisis del mensaje de acuerdo a los módulos que incluye el sistema, es necesario definir en que buzón serán colocados los mensajes con base en el resultado de su análisis.

El buzón de correo por default para los usuarios es $var/spool/mail/\$USER$. Para el sistema se han definido buzones de correo específicos, los cuales se muestran a continuación.

Buzones de correo del sistema ubicados en $\$HOME/mail/$

Buzón	Descripción
LISTAblancausr	Almacena los correos que han sido filtrados por el módulo de lista blanca.
LISTAnegrausr	Almacena los correos filtrados por el módulo de lista negra definida por el usuario.
LISTAnegraweb	Almacena los correos filtrados por el módulo de lista negra tomada desde Internet.
PALABRASclave	Almacena los correos filtrados por el módulo de análisis de palabras clave.
FRASESclave	Almacena los correos filtrados por el módulo de análisis de frases clave.
probable-spam	Almacena los correos filtrados por el SpammAssassin.
spam	Almacena los correos filtrados por el análisis estadístico basado en frases.
nospam	Almacena los correos que no fueron filtrados por ningún módulo.

4.1.2. Análisis de mensajes basado en listas de usuarios conocidos

La heurística basada en listas de usuarios conocidos consiste en definir a los usuarios válidos y a los usuarios que previamente han sido detectados como emisores de SPAM, por tanto son indeseados para el usuario. La lista de usuarios válidos también se conoce como *lista blanca*, la lista de usuarios indeseables también conocida como *lista negra*.

El análisis de mensajes con base en la heurística de las listas de usuarios conocidos, considera el emisor (origen) de los mensajes. Dicho análisis consiste en tomar el dato origen (from) de un cierto mensaje y verifica si el emisor pertenece a alguna de las listas de usuarios.

Si se concluye que el emisor existe en la lista blanca, entonces se acepta el mensaje y se coloca en el buzón normal del usuario. En el algoritmo 5 se tiene el pseudocódigo del proceso de análisis de mensajes con base en la *lista blanca*.

Por otro lado, si se concluye con la existencia del emisor en la lista negra inmediatamente se define el mensaje como tipo SPAM y se coloca en un buzón de mensajes SPAM. El algoritmo 6 muestra el pseudocódigo del proceso de análisis con base en la *lista negra*.

Algoritmo 5 Algoritmo de análisis basado en la heurística, *lista blanca*

Require: Mensaje m, Lista de usuarios válidos
Ensure: Definición del mensaje m de tipo válido

```
1: Al arribar un mensaje m
2: if mensaje tiene encabezado then
3:    Extraer el dato From: del mensaje m
4:    Buscar el dato From: en la lista de usuarios válidos (lista blanca)
5:    if el emisor del mensaje m, pertenece a la lista blanca then
6:       Aceptar el mensaje m
7:       Colocar el mensaje m en el buzón normal del usuario
8:    else
9:       Continuar el análisis del mensaje m con otro módulo
10:   end if
11: end if
```

Algoritmo 6 Algoritmo de análisis basado en la heurística, *lista negra*

Require: Mensaje m, Lista de usuarios inválidos
Ensure: Definición del mensaje m de tipo inválido

```
1: Al arribar un mensaje m
2: if mensaje m tiene encabezado then
3:    Extraer el dato From: del mensaje m
4:    Buscar el dato From: en la lista de usuarios inválidos (lista negra)
5:    if el emisor del mensaje m pertenece a la lista negra then
6:       Rechazar el mensaje m
7:       Colocar el mensaje m en el buzón de mensajes SPAM
8:    else
9:       Continuar el análisis del mensaje m con otro módulo
10:   end if
11: end if
```

Al utilizar las listas de usuarios conocidos, cada usuario debe definir los remitentes válidos y los indeseados, es importante tener responsabilidad al definirlos.

El análisis de mensajes con base en la heurística lista negra incluye una **lista negra publicada en Internet**. Para el análisis basado en esta lista se realiza una conexión remota a un sitio público de Internet y se descarga la lista negra para su uso en el análisis de mensajes (a través del servicio *FTP*). La descarga de la lista negra se realiza periódicamente (con el servicio *crontab* de Linux) con la finalidad de realizar el análisis con base en una lista actualizada.

La efectividad al utilizar las listas de usuarios conocidos para el análisis de mensajes, dependerá de qué tan actualizadas estén dichas listas. Una ventaja al utilizar este método, es la rapidez de análisis, debido a que no realiza muchas operaciones para determinar si el emisor pertenece a una lista o a otra, así también, los datos analizados son pocos (solo analiza el origen).

4.1.3. Análisis de mensajes basado en palabras clave

Se definen como *palabras clave* aquellas palabras que representan características de correos SPAM. Mediante palabras clave, en algunos casos es posible clasificar a un cierto mensaje como SPAM después de analizar su contenido y concluir que contiene una cierta cantidad de palabras clave. Las palabras clave se buscan tanto en el tema como en el cuerpo de los mensajes.

Para el análisis basado en este heurística es indispensable tener actualizada la lista de palabras clave. La actualización se logra con un historial suficiente de mensajes SPAM. A cada palabra clave se lo asocia un valor que significa la penalización de que sea parte de mensajes SPAM, dicho valor se da en un rango de 1 a 4, a mayor valor mayor penalización. El valor ha sido definido a prueba y error por parte del usuario con base en su experiencia en mensajes SPAM.

El proceso de análisis basado en palabras claves se describe de la siguiente manera, al momento de llegar un mensaje al servidor, el mensaje es agrupado en tokens. Cada token es buscado en la lista de palabras clave, por cada token encontrado se toma el valor asociado a dicho token y se suma al valor acumulado de todos los tokens, así mismo, se evalúa si se ha alcanzado el umbral mínimo para que un mensaje sea considerado SPAM. El valor acumulado mayor a un cierto limite (*UmbralSpam*) define a los mensajes como SPAM y un valor menor los define como no SPAM.

Una ventaja de esta heurística, es la adaptabilidad del análisis al criterio del usuario. Otra ventaja es la rapidez del proceso de análisis, debido a que el análisis termina en el momento en que se alcanza el valor mínimo del umbral que define a un mensaje como SPAM. El algoritmo 7 muestra el proceso de análisis de mensajes, utilizando la heurística de palabras clave.

Algoritmo 7 Algoritmo de análisis basado en la heurística, *palabras clave*

Require: Mensaje m, Lista de palabras clave, Valor mínimo para ser SPAM (*UmbralSpam*)
Ensure: Valor de ponderación de ser mensaje de tipo SPAM

```
1: Formar tokens con el tema (subject) y cuerpo del mensaje m
2: for cada token con tamaño mayor a 2 caracteres do
3:    Buscar el token t_i en la lista de palabras clave
4:    if el token t_i existe en la lista de palabras clave then
5:       Recuperar el valor asociado al token t_i
6:       Sumar dicho valor al valor acumulado de todos los tokens
7:       if valor acumulado > UmbralSpam then
8:          Terminar el análisis del mensaje m
9:          Colocar el mensaje m en el buzón de mensajes SPAM
10:      end if
11:   else
12:      Descartar el token t_i
13:      Continuar el análisis de los demás tokens
14:   end if
15: end for
16: Continuar el análisis del mensaje m con otro módulo del sistema
```

4.1.4. Análisis de mensajes basado en frases clave

Una *frases clave* es la asociación de n tokens y representa una característica suficiente para definir a un correo como SPAM.

Para el análisis basado en la heurística de frases clave se requiere de un proceso de selección de frases antes de ser almacenadas en una lista de frases, dicha selección se hace de una gran cantidad de mensajes de tipo SPAM. Es indispensable tener actualizada la lista de frases clave. La actualización consiste en observar, identificar y definir correctamente las frases clave. Las frases pueden ser de tamaño variado.

El proceso de análisis basado en la heurística de frases clave se describe de la siguiente manera, al momento de que un mensaje llega al servidor, se busca en el mensaje alguna de las frases clave tomadas de la lista de frases. Al encontrar alguna frase en el mensaje se termina el análisis, concluyendo que el mensaje es SPAM y se coloca en el buzón de mensajes SPAM. Si al término del análisis no se encontró ninguna frase clave, se continúa el análisis con otro módulo del sistema. En el algoritmo 8 se tiene el pseudocódigo del

proceso de análisis de mensajes haciendo uso de la heurística de frases clave.

Algoritmo 8 Algoritmo de análisis basado en la heurística, *frases clave*

Require: Mensaje m, Repositorio de frases clave
Ensure: Definición del mensaje m de tipo inválido
1: Extraer los datos tema (subject) y cuerpo del mensaje m
2: $mensaje \Leftarrow subject$ y *cuerpo* del mensaje m
3: Recuperar las frases clave de la lista de frases
4: $ListadeFrases \Leftarrow$ frases clave de la lista de frases
5: **for** cada frase f_i de la ListadeFrases **do**
6: Buscar la frase f_i en el mensaje m
7: **if** se encuentra la frase en el mensaje **then**
8: Terminar el análisis del mensaje
9: Definir el mensaje como SPAM
10: Colocar el mensaje en el buzón de correos SPAM
11: **else**
12: Continuar la búsqueda de las demás frases en el mensaje m
13: **end if**
14: **end for**
15: Continuar el análisis del mensaje m con otro módulo

La efectividad del análisis basado en esta heurística está directamente relacionada con el proceso de definición de frases clave, una frase mal seleccionada, afecta ó caso contrario, una frase bien definida, aporta efectividad en el análisis.

Una de las ventajas de esta heurística, es que permite adaptarse a las necesidades del usuario, cada usuario determina las frases clave de acuerdo a su criterio. Por un lado, existen frases que para algunos usuarios son definidas como parte de un mensaje SPAM mientras que para otros usuarios, esas frases son parte de su entorno temático, válido. Como ejemplo de una frase es *Computers4SURE*, donde algunos usuarios la consideran parte de un mensaje SPAM, mientras que para otros, la pueden considerar válida debido a que el mensaje que la contiene, lo consideran como una fuente para conocer nuevos productos en el área de computación.

4.1.5. Análisis probabilístico basado en frases

El último módulo de análisis para detectar y filtrar correos SPAM, es el modelo probabilístico para predecir si un determinado mensaje es SPAM.

Recordemos que el algoritmo involucra dos fases: la primera se refiere a la **fase de entrenamiento**, en ella se toman datos de mensajes SPAM y no SPAM y se almacenan en la lista de frases los datos son: *frases seleccionadas*, *número de ocurrencias de la frase en mensajes SPAM y en no SPAM*, y el *valor de probabilidad de que la frase pertenezca a un mensaje SPAM*. La segunda fase es la **fase de clasificación de mensajes**, aquí se realiza su análisis y con base en la información de la primera fase y de acuerdo al teorema de Bayes determina si un mensaje es SPAM o no SPAM.

Fase de entrenamiento del sistema

Una de las partes importantes que se tienen en el desarrollo del sistema es el entrenamiento. Este proceso permite darle consistencia al proceso de análisis de mensajes. Entre mejor se realice el entrenamiento habrá menos errores de clasificación.

Este proceso se encarga de recolectar y preparar los datos estadísticos. Estos datos se usan para el cálculo de probabilidad a priori, así como también para la clasificación de mensajes.

Para el proceso de entrenamiento se asume que se tiene un historial de mensajes SPAM y no SPAM. Se obtiene información de mensajes SPAM y de mensajes no SPAM, se calcula la probabilidad a priori de cada frase del mensaje de pertenecer a mensajes SPAM de acuerdo a la fórmula 3.2. Considerando su valor de probabilidad se seleccionan las frases más significativas y se actualiza su información en la lista de frases.

En la figura 4.4 se muestra la arquitectura del proceso de entrenamiento. Los procesos que la integran son:

Obtener información de mensajes SPAM agrupa los mensajes en tokens, con los tokens forma frases, para cada frase obtiene su valor de ocurrencia en los mensajes SPAM, también obtiene el valor de mensajes SPAM utilizados.

Obtener información de mensajes no SPAM agrupa los mensajes en tokens y con ellos forma frases, para cada frase calcula su valor de ocurrencia en los mensajes no SPAM, también obtiene el valor de mensajes no SPAM utilizados.

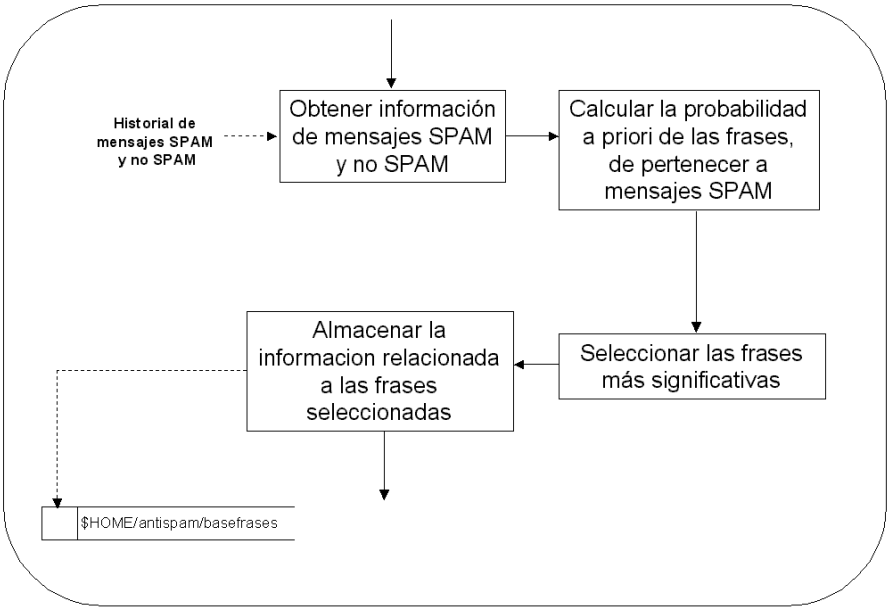

Figura 4.4: Entrenamiento del sistema de análisis de mensajes.

Buscar y obtener información para cada frase como cada frase está integrada por tokens, por eso es necesario obtener los datos de los tokens que integran cada frase. Los datos del token son: el token mismo, su ocurrencia en SPAM y en no SPAM, así como su probabilidad de ser parte de mensajes SPAM.

Calcular la probabilidad a priori de las frases considerando la información obtenida de los procesos anteriores y de acuerdo a la fórmula 3.2 se calcula la probabilidad a priori de las frases de que sean parte de mensajes SPAM.

Seleccionar las frases más significativas con base en el valor de probabilidad a priori de las frases, se seleccionan solo aquellas con valores cercanos a ceros o cercanas a uno, para ser almacenadas en la lista de frases.

Lista de frases, se refiere a las frases y su información, la cual contiene los siguientes datos: *frase*, *número de ocurrencias de las frases en mensajes SPAM*, *número de ocurrencias de las frases en mensajes no SPAM*, *probabilidad a priori de la frase de pertenecer a mensajes SPAM*. En la tabla 4.1 se tiene la estructura de la lista de frases.

Tabla 4.1: Estructura de la lista de frases

Elemento	Tipo	Descripción
frase	char(200)	Frase
$P_{\mathrm{M_{SPAM}}}$	float(0.000)	Probabilidad a priori
nSPAM	int(5)	Número de ocurrencias de la frase en mensajes SPAM
nHAM	int(5)	Número de ocurrencias de la frase en mensajes no SPAM

Fase de clasificación de mensajes

Esta fase permite definir a los mensajes como SPAM o no SPAM con base en la información de la lista de frases.

En la figura 4.5 se presenta la arquitectura del proceso de clasificación de mensajes.

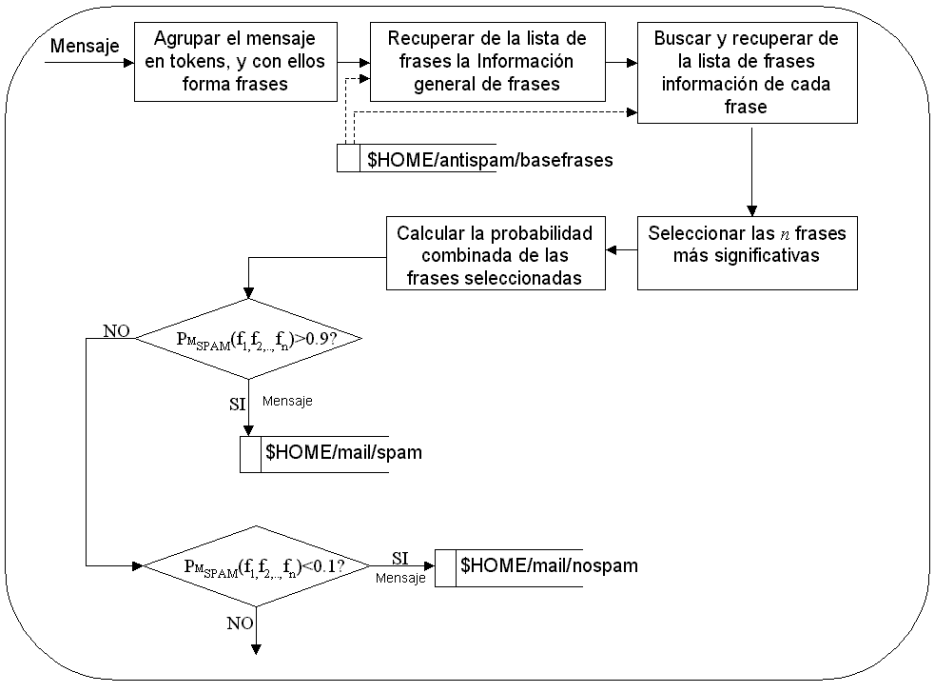

Figura 4.5: Procesos para la clasificación de mensajes.

Lista de frases, para el análisis es necesario contar con información de mensajes SPAM y no SPAM, dicha información se refiere a lo siguiente: *las frases, su número de*

ocurrencias en mensajes SPAM y en no SPAM y su valor de probabilidad de ser parte de mensajes SPAM, así como información general de las frases, como son: *número de correos SPAM* y *número de correos no SPAM*.

Lista de tokens contiene información relacionada con los tokens, y son los siguientes datos: *token*, *número de ocurrencias del token en mensajes SPAM*, *número de ocurrencias del token en mensajes no SPAM*, *probabilidad de que el token pertenezca a un mensaje SPAM*.

Una vez que se han definido la datos necesarios para realizar la clasificación, es necesario describir los procesos que se realizan en el análisis. Veamos a continuación los procesos, su definición y descripción.

Formación de frases con el contenido de cada mensaje se agrupa en tokens, con los tokens de tamaño mayor a dos caracteres se agrupan y se combinan formando frases.

Recuperar la información general de las frases de la lista de frases, se recupera la información general a las frases, los datos que incluye son: *cantidad de mensajes SPAM y no SPAM con que se creo la B.D., cantidad de frases que integra la B.D.*.

Buscar y recuperar información de las frases considerando las frases de mensaje, se busca en la lista de frases la información relacionada a cada frase y se recupera para su uso posterior.

Seleccionar las frases considerando el valor de probabilidad de las frases, se seleccionan las que tienen un valor de probabilidad cercano a cero ó cercano a uno.

Calcular la probabilidad combinada de todas las frases, este valor de probabilidad combinada ofrece la referencia para saber si el mensaje es SPAM o no SPAM. Aquí se realiza la combinación de todos los valores de probabilidad de las frases de pertenecer a un mensaje SPAM, de acuerdo a la fórmula 3.8.

Evaluar el valor de probabilidad obtenida, este valor sirve para definir los mensajes como SPAM o no SPAM. Si el valor esta cercano a 1 se consideran SPAM y en caso de que sea cercano a 0 se consideran no SPAM. Los mensajes definidos como SPAM se colocan en el buzón *$HOME/mail/spam* y los mensajes no SPAM se colocan en el buzón *$HOME/mail/nospam*.

4.2. Adaptación del sistema en el contexto de *procmail*

Como se mencionó al inicio de este capítulo que el sistema está desarrollado como una extensión a la funcionalidad de *procmail*, de tal modo que al leer un correo de la entrada estándar abrirá un archivo de reglas */etc/procmailrc* o $\$HOME/.procmailrc$, en las reglas se indica qué hacer con ese correo en función de las reglas definidas por el usuario [2].

Una de las características más importantes de procmail es que por medio de parámetros definidos en el archivo de reglas es posible ejecutar un programa de usuario.

El entorno básico de procmail y el sistema desarrollado se describe de la manera siguiente, al momento de llegar un correo se lee a través de procmail, con base en su archivo de reglas se manda llamar el sistema de análisis, el cual retorna el resultado del análisis y con base en dicho resultado decide qué hacer con el mensaje, es decir, decide en que buzón lo deposita.

4.2.1. Reglas de *procmail*

Una regla sirve para definir el comportamiento del procmail al momento de tratar el mensaje entrante. Las reglas están definidas en los archivos */etc/procmailrc* o *$HOME/.procmailrc*. A continuación se tiene el esquema general de una regla de procmail.

```
:0 [opciones][:]
* condición A
* condición B
...
* condición n
comando a ejecutar
```

Las líneas que comienzan con : 0 o : 0 : indican el inicio de una regla. La diferencia entre : 0 y : 0 : es que la segunda opción al escribir en el archivo se bloquea para tener acceso exclusivo y no pueda ser accedido por otro proceso al mismo tiempo.

Las líneas que comienzan con $*$ indican la condición de la regla, puede haber más de una condición. La línea que sigue a las condiciones se considera un proceso a ejecutar. El

archivo de reglas puede incluir líneas de comentarios, para ello se coloca el caracter # al inicio de la línea que se desea comentar.

Después de los caracteres de inicio de una regla pueden ir las siguientes opciones, solo se muestran las más útiles para el sistema desarrollado.

- H, la condición se comprobará en la cabecera del mensaje (valor por defecto).
- B, la condición se comprobará en el cuerpo del mensaje.
- D, hace la comprobación, distinguiendo entre mayúsculas y minúsculas.
- c, con esta opción, después de evaluar la regla se crea una copia del mensaje para que pueda verificar más reglas.
- w y W, espera que finalice la ejecución del comando de esa regla para poder recibir el código de salida. Con W no emite ningún mensaje sobre el error y w sí.
- f, hace que procmail actúe como un filtro. Es decir, tras ejecutarse las acciones, se genera un nuevo mensaje de salida que pasará por el resto de reglas del archivo. Sirve para modificar campos o valores de un mensaje.
- h, hace que se filtre la cabecera, solo la cabecera se pasa al comando especificado.
- b, hace que se filtre el cuerpo, solo se pasa el cuerpo al comando.

En el caso de que no se indique ninguna de las opciones anteriores, las condiciones de la regla se aplican a la cabecera (H), pasando como entrada al comando tanto la cabecera (h) como el cuerpo del mensaje (b).

Condiciones

En caso de que una regla se cumpla se ejecutará el comando especificado, pasándole el mensaje en función de las opciones indicadas. Toda condición empieza con el símbolo * y se componen por expresiones regulares. En una expresión regular se utilizan los siguientes símbolos:

- '^', indica comienzo de una línea.
- '$', indica final de una línea.

- '*', indica cero o más veces.
- '?', indica cero o una vez.
- '+', indica una o más veces.
- '.', cualquier caracter excepto un salto de línea.
- [a-z], que quede en un rango de caracteres (primero-último).
- [A-Z], que no quede en un rango de caracteres (primero-último).
- '|', permite especificar operación O ($a|b =' a'O'b'$)

Al iniciar la línea de condición también pueden existir las siguientes opciones:

- '<', permite comprobar si el archivo tiene un tamaño menor al especificado.
- '>', permite comprobar si el archivo tiene un tamaño mayor al especificado.
- '!', invierte la condición.
- '$', realiza sustituciones de variables en las expresiones regulares.
- '?', necesita el resultado que devuelve el programa especificado.

Comandos

Después de cada condición se específica un comando para que sea ejecutado, si la condición se cumple. Se tienen cuatro comandos básicos:

1. Archivo: hace que procmail añada el mensaje al final del archivo con otros posibles mensajes.
2. Directorio: hace que procmail guarde el mensaje en el directorio con un nombre propio.
3. !direccion@email.com: mediante el caracter '!' es posible enviar el mensaje a la dirección de correo especificada.

4. |programa: el caracter '|' permite ejecutar un programa/comando de Linux. La salida del programa será por la salida estándar, es posible redirigir a cualquier lado con el redireccionamiento estándar de Linux (>/dev/null, >archivo, etc), también es posible asignar la salida del programa a una variable de entorno.

La estructura del archivo de reglas definido para procmail y el análisis de mensajes se describen de la siguiente manera, al arribar un mensaje primero es analizado con base en las heurísticas estáticas posteriormente por el análisis estadístico. Este orden se debe a que el análisis basado en reglas estáticas es más rápido, además que la clasificación es a criterio del usuario. Por otro lado, el análisis estadístico requiere de más tiempo de computo para realizar la clasificación. A continuación se tiene el archivo de reglas definido para el sistema.

```
LOGFILE=$HOME/mail/LOGprocmail
FILE=COPY.$$

#ELIMINA CÓDIGO HTML
:0bc
|$HOME/herramientas/html2text -o $HOME/mail/$FILE

#ANALISIS CON LA LISTA BLANCA DEFINIDA POR EL USUARIO
:0
*? formail –x“From:” –x“From” –x“Sender:” | grep -is -f $HOME/antispam/listablanca
$HOME/mail/LISTAblancausr

#ANALISIS CON LA LISTA NEGRA DEFINIDA POR EL USUARIO
:0
*? formail –x“From:” –x“From” –x“Sender:” | grep -is -f $HOME/antispam/listanegra
$HOME/mail/LISTAnegrausr

#ANALISIS CON LA LISTA NEGRA BAJADA DESDE INTERNET
:0
*? formail “From:” –x“From” –x“Sender:” | grep -is -f $HOME/antispam/direccionesspam
$HOME/mail/LISTAnegraweb
```

```
#OBTIENE EL ASUNTO Y CUERPO DEL MENSAJE
:0
TEMA=|formail -xSubject:

:0
BODY=|formail -I “”

#ANALISIS POR FRASES EN EL ASUNTO Y CUERPO DEL MENSAJE
:0
RESPUESTAFRASES=|$HOME/antispam/detectaFrases.exe $TEMA $BODY

:0 wf
| formail -I “VALOR-SPAM: $RESPUESTAFRASES”

:0
* ^VALOR-SPAM: SI
$HOME/mail/Frasesclave

#ANALISIS POR PALABRAS CLAVE EN EL ASUNTO Y CUERPO DEL MENSAJE
:0
RESPTEMA=|$HOME/antispam/antiSPAM.exe $TEMA

:0
RESPBODY=|$HOME/antispam/antiSPAM.exe RESPTEMA=$RESPTEMA $BODY

:0 wf
| formail -I “ES-SPAM: $RESPBODY”

:0
* ^ES-SPAM: SI
$HOME/mail/Palabrasclave
```

```
#ANALISIS CON EL PROGRAMA SPAMASSASSIN
:0fw: spamassassin.lock
|spamassassin

:0
* ^X-Spam-Status: Yes
$HOME/mail/spam

:0:
$HOME/mail/nospam
```

Capítulo 5

Análisis de resultados

En este capítulo se muestra el análisis de resultados obtenidos con el sistema de clasificación de mensajes propuesto. Si bien recordamos, el sistema esta organizado por módulos, aquí se muestran los resultados obtenidos en uno de ellos. Se hace un análisis de los resultados obtenidos con el caso de estudio, al integrar los módulos.

5.1. Infraestructura del ambiente de pruebas

El entorno donde fueron desarrolladas las pruebas, es un servidor de correos de la sección de computación del CINVESTAV-IPN (*computacion.cs.cinvestav.mx*). El servidor tiene las siguientes características de hardware:

Procesador es una computadora dual con dos procesadores *Pentium III* con una velocidad de *800 Mhz*.

Memoria RAM tiene *1 GB* de memoria RAM.

Disco duro tiene un espacio de almacenamiento de *40 GB*, de los cuales cada usuario tiene asignado *100 MB* para el almacenamiento de información.

Dicho servidor tiene el siguiente software instalado:

Sistema operativo Linux, distribución Red hat versión 7.1 con una versión de kernel *2.4.2-2smp* con actualización en diversos paquetes para ofrecer mejores servicios y rendimiento.

Software para el servicio de correo, la recepción y envió de correo se realiza a través del programa *Sendmail*.

Software para la entrega de correo local, la entrega del correo en los buzones de los usuarios, se hace a través del procesador de correos local, *procmail v3.14*.

Software adicional *Crontab*, aplicación para proporcionar entradas a *cron v2.13*. Este último permite la ejecución de tareas periódicas en el sistema.

5.2. Caso de estudio

Para poder observar la efectividad del sistema se plantea el siguiente caso de estudio, de una cuenta de correo del servidor de computación se han tomado 2053 mensajes de su historial, dichos mensajes han sido seleccionados manualmente, siendo 1915 mensajes SPAM y 137 no SPAM.

El análisis de resultados obtenidos con este caso de estudio se divide en dos etapas. *La primera etapa se refiere a tomar información del historial de mensajes* para actualizar los repositorios, esta es la parte más importante, debido a que los resultados dependen de qué tan bien actualizados estén dichos repositorios.

La información consiste de lo siguiente:

Lista blanca esta lista incluye 175 direcciones de correos pertenecientes en su mayoría al personal que labora en la sección de computación del CINVESTAV-IPN, incluyendo las direcciones de correo de los doctores investigadores, del personal administrativo, así como de los alumnos.

Lista negra integrada por 16 direcciones de correos electrónicos y de algunos dominios identificados como emisores de SPAM.

Lista de palabras clave contiene 120 palabras claves y junto con su valor de penalización.

Lista de frases clave contiene 56 frases clave.

Información relacionada a las frases, esta es la información de frases que se utiliza en el análisis estadístico para la clasificación de mensajes.

La segunda etapa consiste en realizar la clasificación de los 2053 *mensajes* a través del sistema antispam desarrollado, con base en la información que se tiene. La clasificación se realiza en etapas, es decir, el análisis se realiza en el siguiente orden: módulo de lista blanca, módulo de lista negra, módulo de frases clave, módulo de palabras clave, análisis con spamassassin y por último, el análisis estadístico basado en frases. Los mensajes que no se filtren con un módulo son la entrada del siguiente módulo hasta pasar por todos. A continuación se muestran los resultados de clasificación de cada módulo, así como su efectividad.

5.2.1. Análisis basado en listas definidas

Este módulo abarca dos casos, por un lado la lista blanca, los mensajes de emisores de esa lista siempre serán aceptados. Por otro lado la lista negra incluye direcciones de correo con emisores inválidos.

El análisis con base en la lista blanca definida tiene los siguientes resultados:

Tabla 5.1: Resultados de análisis basado en la *lista blanca*

Mensajes analizados	2053
Mensajes filtrados	128
Porcentaje de efectividad	6.23 %
Mensajes filtrados acumulados	128
Porcentaje de efectividad acumulado	6.23 %
Errores de falsos positivos	0
Mensajes no filtrados	1925

De acuerdo al análisis basado en la lista negra definida, se tienen los resultados en la tabla 5.2.

Tabla 5.2: Resultados de análisis basado en la *lista negra*

Mensajes analizados	1925
Mensajes filtrados	33
Porcentaje de efectividad	1.71 %
Mensajes filtrados acumulados	161
Porcentaje de efectividad acumulado	7.84 %
Errores de falsos positivos	0
Mensajes no filtrados	1892

El análisis basado en las listas de usuarios conocidos es eficaz debido a que solo considera el origen del mensaje para comprobar si pertenece a alguna de las listas.

5.2.2. Análisis basado en frases clave

El análisis con este modulo consiste en encontrar al menos alguna frase clave en el tema o en el cuerpo del mensaje. En la tabla 5.3 se muestran algunas frases utilizadas para la clasificación. Dichas frases han sido seleccionadas de los 1915 mensajes SPAM seleccionados manualmente de un total de 2053 mensajes del historial considerado en el caso de estudio.

Tabla 5.3: Listado de algunas frases clave utilizadas en este módulo

we have every popular	before sex,
super low	you can refuse to receive
we have all	check out our offer.
new product!	big sale

Los resultados de clasificación con este módulo son los siguientes.

Tabla 5.4: Resultados de análisis basado en *frases clave*

Mensajes analizados	1892
Mensajes filtrados	589
Porcentaje de efectividad	31.13 %
Mensajes filtrados acumulados	750
Porcentaje de efectividad acumulado	36.53 %
Errores de falsos positivos	0
Mensajes no filtrados	1303

La efectividad de este módulo radica en una adecuada selección de frases, debido a la condición de que el mensaje que contenga mínimo una frase clave es definido como SPAM. Al momento de definir la lista de frases clave debe ser adaptado a cada usuario para evitar errores de falsos positivos (mensajes buenos considerados como SPAM). Esta sección cubre la adaptabilidad del sistema a los usuarios.

5.2.3. Análisis basado en palabras clave

Este módulo de análisis se encarga de buscar una serie de palabras definidas como características de mensajes SPAM. Al encontrarse algunas de ellas en el mensaje, se toma el

valor asociado a la palabra y se agrega al acumulado de todo el proceso de análisis del mensaje. Este proceso termina, en el momento en que se alcanza el límite que define a los mensajes como SPAM o cuando se ha revisado todo el contenido del mensaje. El valor de *Umbral de SPAM* se ha definido que sea 7, este valor se definió a prueba y error.

Este módulo tiene la característica de buscar palabras clave tanto en el tema como en el cuerpo del mensaje. Los resultados de clasificación con este módulo se tienen a continuación.

Tabla 5.5: Resultados de análisis basado en *palabras clave*

Mensajes analizados	1303
Mensajes filtrados	504
Porcentaje de efectividad	24.54 %
Mensajes filtrados acumulados	1254
Porcentaje de efectividad acumulado	61.08 %
Errores de falsos positivos	1
Mensajes no filtrados	799

En la tabla 5.5 se reporta un error de clasificación (falso positivo), este mensaje fue clasificado como SPAM sin serlo, debido a que en su contenido incluía ejemplos de características de mensajes SPAM, dicho contenido era con la finalidad de prevenir al destinatario de riesgos de virus. La solución a casos similares es agregar a la mayoría o todas las direcciones de correo de los usuarios conocidos a la lista blanca.

La efectividad de este módulo depende del valor de penalización asignado a las palabras clave. Si se asignan valores altos rápidamente llegará al límite para considerarse mensaje SPAM. Y si se asignan valores mínimos, sera necesario que el mensaje analizado contenga muchos palabras clave, antes de ser definido como SPAM.

5.2.4. Análisis estadístico basado en tokens *SpamAssassin*

Este análisis es cubierto por el programa *SpamAssassin*[27]. Dicho análisis consiste en clasificar mensajes con base en el análisis de sus tokens, es decir, obtiene la probabilidad estimada de que un cierto mensaje sea SPAM tomando tokens como evidencias.

En este modelo de análisis es necesario el proceso de entrenamiento para actualizar la información relacionada a los tokens. Para efectuar el entrenamiento se asume que se

tienen agrupados los mensajes SPAM, así como los mensajes no SPAM, en archivos de tipo *mbox*, ambos grupos han sido clasificados manualmente.

Entrenamiento con mensajes de tipo SPAM:

[user**@**pc ∼]$**sa** − **learn** − −*spam* − −*mbox*[*archivo*]

Entrenamiento con mensajes de tipo no SPAM:

[user**@**pc ∼]$**sa** − **learn** − −*ham* − −*mbox*[*archivo*]

Los resultados de análisis obtenidos con el *SpamAssassin* son los siguientes.

Tabla 5.6: Resultados de análisis estadístico basado en *tokens*

Mensajes analizados	799
Mensajes filtrados	104
Porcentaje de efectividad	13.01 %
Mensajes filtrados acumulados	1358
Porcentaje de efectividad acumulado	66.14 %
Errores de falsos positivos	0
Mensajes no filtrados	695

La limitada efectividad de este modelo se debe a las características de su análisis, ya que considera tokens con probabilidades cercanas a cero ó cercanas a uno, $P_{M_{SPAM}}(t_i) < 0.1$ ó $P_{M_{SPAM}}(t_i) > 0.9$, y en algunas ocasiones al integrar tokens y considerarse como un único elemento aumenta su probabilidad de ser parte de un mensaje SPAM, pero debido a las características de este modelo son discriminados algunos tokens que pueden servir para realizar la clasificación del mensaje.

5.2.5. Análisis estadístico basado en frases

Este modelo de análisis cubre las deficiencias y limitaciones del análisis estadístico basado en tokens. La ventaja de este modelo tiene su fundamento en que integra frases con aquellos tokens con probabilidades $P_{M_{SPAM}}(t_i) < 0.35$ ó $P_{M_{SPAM}}(t_i) > 0.65$. De las frases resultantes solo considera aquellas con probabilidad $P_{M_{SPAM}}(f_i) < 0.1$ ó $P_{M_{SPAM}}(f_i) > 0.9$ para obtener una probabilidad de que el mensaje sea SPAM, considerando las frases como evidencias. Los resultados de análisis se presentan en la siguiente tabla.

Los resultados de clasificación de la tabla 5.7 muestran un error de falso positivo, éste error se debe a que de los 2053 mensajes utilizados para el entrenamiento, 1915 eran

Tabla 5.7: Resultados de análisis estadístico basado en *frases*

Mensajes analizados	695
Mensajes filtrados	667
Porcentaje de efectividad	95.97 %
Mensajes filtrados acumulados	2025
Porcentaje de efectividad acumulado	98.63 %
Errores de falsos positivos	1
Mensajes no filtrados	28

mensajes de tipo SPAM contra 137 no SPAM. No resultaron más errores debido a la diferencia marcada de contenidos, por un lado el contenido de los mensajes SPAM estaba en inglés a diferencia de los no SPAM que estaba en español.

Por lo anterior, la efectividad de este módulo radica en el proceso de actualización o entrenamiento de la información relacionada a las frases. Resulta mas efectivo, cuando se realiza en entrenamiento con cantidades similares de mensajes de ambos tipos, SPAM y no SPAM, además de contenido variado. Para el entrenamiento se asume que se tienen agrupados en un archivo los mensajes SPAM y en otro los mensajes no SPAM, el entrenamiento se realiza de la siguiente manera:

[user**@**pc ~]$./*antispam/entrenaSPAM.exe*

[user**@**pc ~]$./*antispam/entrenaNOSPAM.exe*

En ambos casos, pedirá el nombre del archivo que agrupa los mensajes para el entrenamiento, es necesario indicar la ruta completa del archivo.

A continuación se muestran resultados de todos los módulos.

Los tiempos del servicio de entrega de correo se ven aumentados, debido al retraso que genera el proceso de análisis de correos para detectar SPAM. Para cuantificar los retrasos de tiempo en la entrega de correo, es necesario considerar el escenario donde se realiza el proceso de análisis, esto se refiere al equipo de cómputo. Esto cubre los aspectos de hardware(procesador, memoria RAM) y de software(sistema operativo). Otro factor que influye en el procesamiento normal de entrega de correos se debe a los diferentes métodos de análisis.

ANALIZADOS = ANALIZA	POR MODULOS = X MOD
FILTRADOS = FILTRA	ACUMULADO = ACUMU
NO FILTRADOS = NOFILTRA	ERRORES = ERR

Tabla 5.8: Tabla de resultados de todos los módulos

Módulo	Mensajes			Porcentaje		
	ANALIZA	FILTRA	NOFILTRA	X MOD	ACUMU	ERR
Lista Blanca	2053	128	1925	6.23 %	6.23 %	0
Lista Negra	1925	33	1892	1.60 %	7.83 %	0
Frases clave	1892	589	1303	28.68 %	36.51 %	0
Palabras clave	1303	504	799	24.54 %	61.05 %	1
Estadístico tokens	799	104	695	5.06 %	66.11 %	0
Estadístico frases	695	667	28	32.48 %	98.59 %	1
TOTAL	2053	2025	28		98.59 %	2

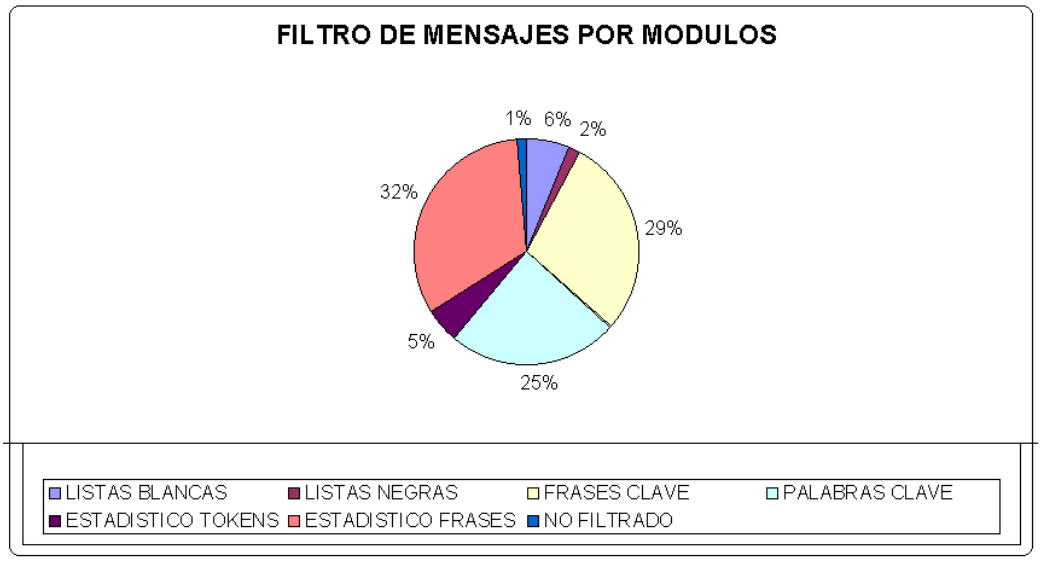

Figura 5.1: Porcentajes de clasificación por módulos.

Debemos recordar lo que se pretende en este sistema de análisis, es detectar y definir correctamente mensajes SPAM, es decir, la importancia se centra en la efectividad del análisis. Además, el destinatario no esta a la espera de sus mensajes, sino al momento de revisar su buzón es como se percata de que le ha llegado un nuevo mensaje.

5.3. Caso de estudio 2

Para un segundo caso de estudio, se considera otro conjunto de mensajes de otra cuenta de

correo del servidor de computación. Dicho conjunto esta integrado de 731 mensajes, de los cuales 272 son SPAM y 427 no SPAM, ambos conjuntos han sido seleccionados manualmente. A continuación se muestran los resultados, de manera resumida, de la clasificación de los mensajes por medio del sistema antispam desarrollado.

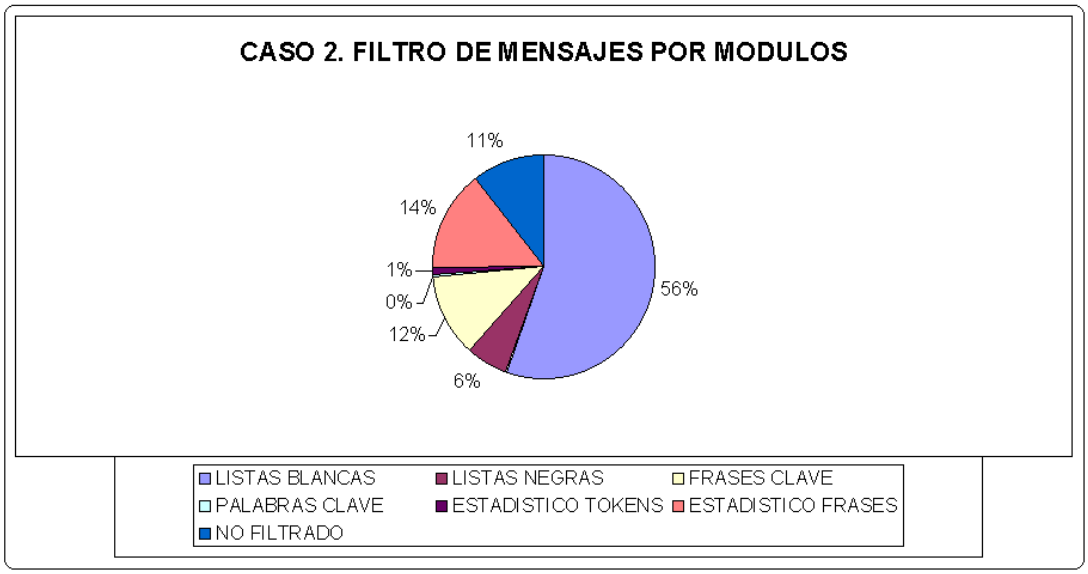

Figura 5.2: Porcentajes de clasificación por módulos (caso 2)

Tabla 5.9: Tabla de resultados de todos los módulos (caso 2)

Módulo	Mensajes			Porcentaje		
	ANALIZA	FILTRA	NOFILTRA	X MOD	ACUMU	ERR
Lista Blanca	731	407	324	55.67 %	55.67 %	0
Lista Negra	324	44	280	6.01 %	61.68 %	0
Frases clave	280	87	193	11.90 %	73.58 %	0
Palabras clave	193	3	190	0.41 %	73.99 %	0
Estadístico tokens	190	8	182	1.09 %	75.08 %	0
Estadístico frases	182	104	78	32.48 %	89.30 %	0
TOTAL	731	653	78		89.30 %	0

ANALIZADOS = ANALIZA	POR MODULOS = X MOD
FILTRADOS = FILTRA	ACUMULADO = ACUMU
NO FILTRADOS = NOFILTRA	ERRORES = ERR

Capítulo 6

Conclusiones

El servicio de correo electrónico ha llegado a ser parte esencial entre los medios de comunicación electrónicos, debido a su gran utilidad y beneficios. Pero ante este auge se han presentado abusos en dicho servicio. Tal abuso se refiere a la presencia del correo SPAM en los buzones de los usuarios, dichos mensajes provocan pérdida de tiempo para ser eliminados, así como la disminución del rendimiento de los recursos computacionales.

El presente trabajo de tesis muestra el desarrollo de una herramienta para disminuir y en algunos casos eliminar el problema del SPAM. Dicha herramienta está integrada por módulos, los cuales cada uno se encarga de un tipo de análisis en busca de SPAM. Así mismo, trabajando en conjunto aportan en mayor medida para disminuir ó eliminar el problema del SPAM.

El análisis principal del sistema desarrollado está basado en el algoritmo de Paul Graham, descrito en la sección 2.5. El algoritmo propuesto en este trabajo incluye mejoras importantes, entre otras, fue adaptado para predecir si un mensaje es SPAM considerando frases, logrando con ello tener evidencias más precisas para la clasificación de mensajes.

6.1. Contribuciones

En el presente trabajo de tesis se muestra el desarrollo de un sistema encargado de detectar y filtrar correos SPAM. El enfoque de dicho sistema es el de integrar métodos y técnicas distintas para el análisis.

El sistema de análisis está integrado por reglas estáticas, heurísticas, así como por un

análisis probabilístico. Todo esto brinda mayor efectividad al clasificar mensajes. Debido a las características de análisis permite ser un sistema adaptativo de acuerdo a las necesidades de los usuarios.

A continuación se muestran las características más importantes del sistema de análisis desarrollado:

- Permite detectar mensajes SPAM con un porcentaje de efectividad, que va desde 95 % a 98 %.
- Implementa reglas estáticas, heurísticas, y un modelo estadístico, logrando con esto mayor porcentaje de efectividad.
- En la parte dinámica implementa un algoritmo basado en la estadística y la probabilidad para predecir si el mensaje es SPAM considerando frases de cierto tamaño, como evidencias.
- Permite adaptarse a las necesidades de los usuarios debido al uso de las heurísticas.
- La parte estática está integrada por reglas estáticas y heurísticas como son: listas conocidas, palabras clave, frases clave.
- Al integrar las reglas estáticas y las heurísticas, hace que el sistema mantenga un equilibrio entre efectividad, eficiencia y adaptabilidad. Con el algoritmo probabilístico bayesiano se logra efectividad, con las listas conocidas se logra un buen rendimiento y con las heurísticas de palabras y frases clave permite al sistema ser adaptable a los usuarios.
- El modelo dinámico permite que el sistema sea mantenible, ya que puede ser entrenado de acuerdo a las características de nuevos mensajes SPAM.

En base a los resultados de los casos de estudio presentados en el capitulo 5, se puede observar que el sistema de análisis propuesto y desarrollado realiza una clasificación efectiva, casi sin errores. Los resultados de efectividad deben estar ligados a los de eficiencia. Y este sistema los cumple, debido a que combinan aspectos estáticos y dinámicos.

6.2. Trabajo futuro

El trabajo de tesis presentado cumple con los objetivos propuestos al inicio, que eran detectar y filtrar los mensajes SPAM. Sin embargo, será de gran utilidad contar con una herramienta complementaria al actual desarrollo. Estos complementos serían en el aspecto de preparación y mantenimiento de la información de manera automática.

Como se mencionó en algunas secciones, los modelos desarrollados requieren de un buen entrenamiento para poder obtener buenos resultados en el análisis. Actualmente, estos procesos de preparación de la herramienta se hacen de manera manual. Es deseable que dicha herramienta logre su preparación, entrenamiento y mantenimiento de manera automática, logrando que sea adaptativo de manera automáticamente, tanto para los usuarios como para la detección de mensajes con contenido novedoso, truncando con ello las intenciones de los emisores de SPAM.

Bibliografía

[1] B. Costales y E. Allman; *Sendmail*. O'Reilly Associates, 1997.

[2] Sitio del proyecto procmail; *Procmail Documentation Project*. http://pm-doc.sourceforge.net/, 2004.

[3] P. Graham; *A plan for spam*. http://www.paulgraham.com/spam.html, 2002.

[4] P. Graham; *Better bayesian filtering*. http://www.paulgraham.com/better.html, 2003.

[5] P. Graham; *Probability*. http://www.paulgraham.com/naivebayes.html, 2002.

[6] M. Sahami, S. Dumais y E. Horvitz; *A Bayesian Approach to Filtering Junk E-Mail*. Madison, WI, 1998.

[7] W. W. Cohen; *Learning rules that clasify e-mail*. AAAI Simposium de Informática de Verano, 1996.

[8] Druker, H.; Donghui Wu; Vapnik, V.N.; *Support Vector Machines for Spam Categorization*. Neural Networks, IEEE Transactions on Volume 10, Issue 5, 1999. pp. 1048-1054.

[9] Ivey, K.C.; *Spam: the plague of junk E-mail*. Computer Applications in Power, IEEE Volume 11, Issue 2. 1998. pp. 15-16.

[10] Yamai, N.; Okayama, K.; Miyashita, T.; Maruyama, S.; Nakamura, M.; *A protection method against massive error mails caused by sender spoofed spam mails*. Applications and the Internet, 2005, The 2005 Symposium on 31 Jan.-4 Feb. 2005. pp. 384-390.

[11] Pelletier, L.; Almhana, J.; Choulakian, V.; *Adaptive filtering of spam*. Communication Networks and Services Research, 2004, Proceedings. Second Annual Conference on 19-21 May 2004. pp. 218-224.

[12] Hess, A.; Klaue, J.; *A video-spam detection approach for unprotected multimedia flows based on active networks*. Euromicro Conference, 2004. Proceedings. 30th 2004 pp. 461-465

[13] Holmes, N.; *In Defense of Spam*. Computer Volume 38, Issue 4, April 2005. pp. 88-87.

[14] Matsumoto, R.; Du Zhang; Meiliu Lu; *Some empirical results on two spam detection methods*. Information Reuse and Integration, 2004. IRI 2004. Proceedings of the 2004 IEEE International Conference on 8-10 Nov. 2004. pp. 198-203.

[15] Mane, S.; Srivastava, J.; San-Yin Hwang; Vayghan, J.; *Estimation of false negatives in classification*. Data Mining, 2004. ICDM 2004. Proceedings. Fourth IEEE International Conference on 1-4, Nov. 2004. pp. 475-478.

[16] Clark, J.; Koprinska, I.; Poon, J.; Srivastava, J.; San-Yin Hwang; Vayghan, J.; *A neural network based approach to automated e-mail classification*. Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on 13-17 Oct. 2003. pp. 702-705.

[17] Asociación de usuarios de Internet; *Estadísticas del tráfico de SPAM*. http://pe-pi-ii.com/estadisticas/, 2004.

[18] The Web's Richest Source; *The Deadly Duo: Spam and Viruses*. http://www.clickz.com/stats/sectors/email/article.php/3483541#table1, 2005.

[19] Alejandro Ángeles; *Spam, amenaza cercana*. http://www.eluniversal.com.mx/pls/-impreso/noticia_supl.html?id_articulo=18873&tabla=articulos, 2005.

[20] Centro de alerta temprana sobre virus y seguridad informática; *Anti-Spam*. http://alerta-antivirus.red.es/utiles/ver.php?tema=U&articulo=11, 2005.

[21] Trivedi & Kishor S.; *Probability and Estatics*. Prentice Hall. 1982. pp. 23-39.

[22] Meyer, P. L.; *Probabilidad y aplicaciones estadísticas*. Addison-Wesley Iberoamericana. 1992. pp. 43-54.

[23] Drake, A. W.; *Fundamentals of applied probability theory*. McGraw-Hill. pp. 13-27.

[24] Casorla-Quevedo, M.A. & Escolano Ruíz, F.; *Un enfoque bayesiano para la extracción de características y agrupamiento en visión artificial*. Ph.D. thesis, Departamento de Ciencia de la Computación e Inteligencia Artificial. 2000. pp. 145-146.

[25] MathPages; *Combining Probabilities*. http://www.mathpages.com/home/-kmath267.htm, 2002.

[26] P. Graham; *Ejemplo de un mensaje de tipo SPAM* . http://www.paulgraham.com/lib/paulgraham/spam2.txt, 2002.

[27] Open-Source Spam Filter; *The Apache SpamAssassin Project*. http://spamassassin.apache.org/, 2005.

[28] Greg Louis; *Fisher Method*. http://www.bgl.nu/bogofilter/tuning.html, 2004.