

2017

Automatización de respuestas en canales digitales para centros de atención al cliente

UNED / Máster Inteligencia Artificial Avanzada
Trabajo Fin de Máster



Contenidos

1	Introducción	7
1.1	Caso de estudio: Sistema de venta de entradas de Patrimonio Nacional.....	8
1.2	Objetivos	13
1.3	Estructura del documento	13
2	Definición del problema	14
2.1	Características del problema	14
2.2	Modelo de interacción.....	15
2.3	Aspectos cualitativos	15
3	Estado del arte	16
3.1	Primeras aproximaciones	16
3.2	Visión global.....	23
4	Corpus de entrenamiento y test	27
4.1	Extracción de los mensajes	28
4.2	Normalización de los mensajes	30
4.3	Utilización de lexicón	32
4.4	Métricas del corpus	36
5	Construcción de clasificadores.....	40
5.1	Representación de las instancias.....	43
5.2	Balanceo de datos.....	44
6	Resultados clasificación.....	45
6.1	Clase 3. Resultados clasificación.....	45
6.2	Clase 33. Resultados clasificación.....	48
6.3	Clase 25. Resultados clasificación.....	50
6.4	Clase 15. Resultados clasificación.....	53
6.5	Clase 18. Resultados y análisis.....	55
6.6	Análisis clasificadores binarios.....	56
6.7	Análisis de la composición de clasificadores	59
7	Conclusiones.....	62
8	Futura mejoras	64
9	Bibliografía	66
10	Anexos	69

10.1	Lexicón	69
10.2	Librerías y herramientas	72

Tablas

Tabla 1 – Principales plantillas utilizadas en el servicio de atención al cliente de Patrimonio Nacional	11
Tabla 2 - Ejemplos de emails remitidos desde el formulario de contacto de la web de venta de entradas. Por legislación omitimos los datos identificativos del cliente.	12
Tabla 3 - Emails remitidos directamente por el cliente al centro de atención al cliente. Por legislación omitimos los datos identificativos del cliente.....	12
Tabla 4- Número de instancia por clase de respuesta y peso respecto del total de mensajes utilizados.....	29
Tabla 5- Procesos de filtrado de mensajes para la construcción del corpus	31
Tabla 6 - Longitud promedio y distancia Jaccard promedio entre todos los mensajes del corpus	36
Tabla 7 - Para cada clase longitud promedio de mensajes positivos y negativos y distancia Jaccard entre todos los mensajes de la clase positiva y negativa (interclase) y distancia dentro de la clase positiva (intraclase).	36
Tabla 8- Diferencias en distancia Jaccard interclase y intraclase para todas las clases del corpus	37
Tabla 9 - Número de mensajes duplicados por clase	39
Tabla 10 - Parámetros de configuración para SVM en WEKA	42
Tabla 11- Número de instancias positivas y negativas de entrenamiento por clase y ratio de balanceo.....	43
Tabla 12 - Número de instancias positivas y negativas de test por clase	44
Tabla 13 – Entrenamiento clase 3. Resultados para los experimentos con MNB y SVM sobre las distintas configuraciones de oversampling.....	46
Tabla 14 – Test clase 3. Resultados de todos los clasificadores para MNB y SVM	46
Tabla 15 – Clase 3. Ajuste básico del parámetro C para SVM con kernel lineal. Resultados mostrados sobre datos de test.	46
Tabla 16 - Clase 3. Estudio para un ajuste básico del parámetro C para SVM con kernel polinomial de grado 2. Resultados mostrados sobre datos de test.....	47
Tabla 17 - Clase 3. Resultados para KNN + Jaccard	47
Tabla 18 – Clase 3. Ajuste del parámetro K para controlar la certeza en la predicción. 48	
Tabla 19 – Entrenamiento clase 33. Resultados de los experimentos para MNB y SVM sobre las distintas configuraciones de oversampling.....	48
Tabla 20 – Test clase 3. Resultados de los experimento para MNB y SVM sobre las distintas configuraciones de oversampling.	49
Tabla 21 – Clase 33. Ajuste básico del parámetro C para SVM con kernel lineal. Resultados mostrados sobre datos de test.	49
Tabla 22 - Clase 33. Ajuste básico del parámetro C para SVM con kernel polinomial de grado 2. Resultados mostrados sobre datos de test.	49
Tabla 23 - Clase 33. Resultado KNN + Jaccard	49

Tabla 24- Clase 33. Ajuste del parámetro K para controlar la certeza en la predicción	50
Tabla 25- Clase 25. Resultados de los experimento en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling.....	51
Tabla 26 - Clase 25. Resultados de los experimento en test para MNB y SVM sobre las distintas configuraciones de oversampling.....	51
Tabla 27 - Clase 25. Ajuste básico del parámetro C para SVM con kernel lineal. Resultados mostrados sobre datos de test.....	51
Tabla 28 - Clase 25. Resultados para KNN + Jaccard	52
Tabla 29 - Clase 25. El ajuste del umbral para las probabilidades de salida obtenidas para SVM con kernel lineal.....	52
Tabla 30 – Clase 15. Resultados en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling.....	53
Tabla 31 - Clase 15. Resultados en test para MNB y SVM sobre las distintas configuraciones de oversampling.....	53
Tabla 32 - Clase 15. Resultados KNN + Jaccard	53
Tabla 33 - Clase 18. Resultados en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling.....	55
Tabla 34 - Clase 18. Resultados en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling.....	55
Tabla 35 - Clase 18. Resultados para KNN + Jaccard	55
Tabla 36 - Mejores resultados de cobertura y precisión para todas las técnicas y clases estudiadas. La columna “Over.” muestra el grado de oversampling utilizado. La columna de precisión para SVM incluye el incremento en precisión sobre el resto de técnicas.....	57
Tabla 37 - Curvas PR para los mejores clasificadores de cada técnica para todas las clases.....	58

Ilustraciones

Ilustración 1 - Formulario web actual de solicitud de información y notificación de incidencias para los usuarios del sistema de compra de entradas online	10
Ilustración 2 - Intuición del concepto de solapamiento espacial entre los n clasificadores binarios. La línea verde muestra el criterio del clasificador para la clase A, la azul para la clase B. Aunque ambas separan perfectamente sus muestras no hay garantía de que una muestra no vista caiga en un espacio que ambas clases consideraran un positivo.	25
Ilustración 3 - Distribución frecuencias Jaccard para todo el corpus.....	37
Ilustración 4 - Clase 3. Distribución de frecuencias sobre distancia Jaccard	38
Ilustración 5 - Clase 32. Distribución de frecuencias sobre distancia Jaccard	38
Ilustración 6 - Clase 25. Distribución de frecuencias sobre distancia Jaccard	38
Ilustración 7 - Clase 15. Distribución de frecuencias sobre distancia Jaccard	38
Ilustración 8 - Clase 18. Distribución de frecuencias sobre distancia Jaccard	38
Ilustración 9 – Clase 3. La precisión de SVM con un kernel polinomial de grado 2 débilmente ajustado claramente supera en precisión al resto de configuraciones.	47
Ilustración 10 - Clase 33. La precisión de SVM con un kernel polinomial de grado 2 débilmente ajustado claramente supera en precisión al resto de configuraciones.	50
Ilustración 11 - Clase 25. SVM con kernel polinomial de grado 2 y $C=0.1$ supera ampliamente en precisión al resto de técnicas.....	52
Ilustración 12 - Clase 15. SVM con kernel lineal y $C=0.01$ supera ampliamente en precisión a MNB y Jaccard.....	54
Ilustración 13 - Clase 15. Muy buenos resultados para todas los clasificadores . SVM con kernel lineal sin ajuste de C es de nuevo la mejor técnica.....	56
Ilustración 14 - Curva precisión recall para la clase positiva (se envía email automático) para el clasificador global formado por la composición de los clasificadores binarios de cada clase.....	60

1 Introducción

Internet lo ha cambiado todo. Quizás esta sea una de las afirmaciones más recurrentes en la última década. La realidad es que esta frase va camino de convertirse en una verdad indiscutible a una velocidad de vértigo. Usar el término revolución quizás sea un poco pretencioso, pero la realidad es que el movimiento masivo de servicios del mundo real al mundo virtual es uno de los síntomas más claros de esta nueva realidad. Sin embargo, este escenario de digitalización de servicios lleva parejo la necesidad de disponer de centros de atención al cliente que puedan resolver las dudas y problemas que sufren los usuarios. Es posible que a largo plazo estos centros parcialmente o en su totalidad sean reemplazados por bots, pero hoy en día paradójicamente son una parte indispensable de cualquier sitio de comercio electrónico o sede electrónica en Internet.

Un centro de atención al cliente en esencia es un equipo humano más un conjunto de canales de comunicaciones con los clientes. Habitualmente la disponibilidad horaria obliga a que estos centros tengan una alta disponibilidad de personal, 12 o 24 horas 365 días al año suele ser un requisito muy habitual en función de la naturaleza nacional o internacional del servicio. En cuanto a los canales de comunicaciones, aunque en los últimos tiempos nuevos canales como los chats online, o las aplicaciones de mensajería instantánea han hecho aparición, los dos canales principales siguen siendo el teléfono y el email.

El trabajo de los operadores en estos centros consiste en recepcionar consultas de los usuarios o incidencias en los sistemas y tratar de ofrecer una solución de forma precisa y rápida. De los canales de comunicaciones sin ninguna duda el más beneficioso para el trabajo de estos centros es el email. La naturaleza asíncrona del correo y un contacto más aséptico con el cliente ofrece numerosas ventajas frente al uso del teléfono. En contrapartida tenemos el problema del volumen de email que diariamente tienen que gestionar. Al contrario de lo que ocurre con el teléfono, que cuando el número operadores ocupados alcanza el máximo, los clientes tienen que esperar o desistir, los servidores de correo pueden aceptar cientos de miles de mensajes diariamente que deben recibir respuesta. El problema es aún mayor cuando los equipos de trabajo no están correctamente dimensionados para la carga que reciben. La consecuencia directa de este escenario son clientes que reciben sus respuestas tardíamente o fuera de plazo con un efecto negativo en la imagen y cuenta de balances de las entidades.

Mayoritariamente, es habitual encontrarse en estos centros, que un número de solicitudes o incidencias de los clientes se repite constantemente generando siempre la misma respuesta. Bajo esta circunstancia, los operadores o los responsables de los centros construyen plantillas que les permiten reducir el tiempo empleado en redactar las respuestas. Estas plantillas quedan disponibles dentro del software de gestión de tickets utilizado por el centro (el término ticket referencia en el sector a cada una de

las consultas o incidencias recibidas). Cuando un operador recibe un nuevo ticket lee el email recibido y si es una de las preguntas/respuestas predefinida selecciona de la lista de plantillas disponible la más adecuada. A continuación envía la respuesta directamente o realiza alguna edición si es necesaria antes del envío.

Aunque la utilización de plantillas supone una drástica reducción del tiempo empleado por los operadores en cada ticket, todavía necesitan emplear una ingente cantidad de tiempo en atender diariamente decenas, cientos o miles de estos casos de forma completamente rutinaria y repetitiva. Si pensamos en el ciclo de vida de cualquiera de estos servicios online, el coste asociado a estas tareas repetitivas a lo largo de periodos de 5, 10, 15 o más años es enorme para cualquier institución o empresa.

En este trabajo, asumiendo una serie de restricciones, estudiamos la viabilidad de automatizar este tipo de tareas básicas y repetitivas. Es decir, es viable construir un sistema que automáticamente puede determinar si un email recibido pertenece a una de las preguntas/respuestas habituales y emitir una respuesta **precisa** al cliente sin supervisión de un operador. Creemos que un sistema de este tipo podría tener un importante impacto en la reducción de carga de trabajo de estos centros y en consecuencia en los costes asociados. Es importante recalcar que hablamos de respuesta sin supervisión y no de automatización en la selección de plantilla. Esta última deja la decisión final de envío al operador garantizando una respuesta 100% correcta, sin embargo, la reducción en tiempo y costes es en la práctica inexistente dado que los operadores por norma sólo necesitan elegir una plantilla de un desplegable o una lista. La autentica reducción de costes proviene de evitar que los emails lleguen a los operadores.

A lo largo del trabajo utilizaremos como escenario de estudio el servicio de atención al cliente vinculado al servicio de venta de entradas de Patrimonio Nacional. Un escenario que por nuestra vinculación laboral nos permite acceder en detalle a la información y operativa utilizada.

1.1 Caso de estudio: Sistema de venta de entradas de Patrimonio Nacional

Desde mayo de 2014 la empresa Grupo Meana S.A., es adjudicataria del sistema de venta de entradas de Patrimonio Nacional. Esta entidad es la encargada de gestionar recintos como el Palacio Real De Madrid, San Lorenzo del Escorial, Real Palacio de Aranjuez y otra docena de monumentos históricos en la geografía española. El sistema además de ofrecer la habitual venta en taquilla dispone de un sistema de venta anticipada online utilizado tanto por público como por operadores turísticos. Como parte de la adjudicación, nuestra empresa está obligada a ofrecer un servicio de soporte 365/12h a los usuarios del sistema a través de teléfono y correo electrónico. Actualmente el soporte vía email se implementa a través de personal de la empresa que debe atender un volumen anual de aproximadamente 6.000 emails de clientes

sólo de este servicio. El mismo personal también realiza el soporte del servicio de venta de entradas de otras entidades como Instituto Nacional de las Artes Escénicas, Festival de Granada, etc. así como la atención telefónica de todos ellos.

Los clientes remiten sus mensajes a través de dos vías: su cliente habitual de email o el formulario habilitado en la web de venta de entradas en la sección de contacto (Ilustración 1). Todos los emails recibidos son ubicados por el servidor de correo externo en un buzón destinado exclusivamente a este servicio. La herramienta interna de gestión de tickets consulta periódicamente este buzón y muestra a los operadores los nuevos emails ya con un código de seguimiento asociado (ticket). En nuestro caso, la herramienta utilizada es OTRS, una aplicación de código abierto desarrollada en PERL sobre una base de datos MySQL. La herramienta cuenta con una interfaz web donde los operadores pueden atender los nuevos casos, seguir casos ya abiertos o dar por cerrado un caso. OTRS también permite crear distintas colas de mensajes, donde cada cola representa la atención al cliente para un determinado servicio o buzón de correo. Asociado a cada cola OTRS permite definir un conjunto de plantillas que el usuario puede seleccionar en el momento de construir una respuesta mediante un desplegable. Al realizar la selección OTRS copia el contenido de la plantilla en el cuerpo del mensaje de la respuesta.

A los pocos meses de poner en marcha el sistema de venta de entradas de Patrimonio Nacional, los operadores habían desarrollado un conjunto de unas 30 plantillas que cubren aproximadamente el 80% de los tickets recibidos y en su mayor parte necesitan nula o muy poca edición para la construcción de la respuesta. Este dato del 80% es un dato estimado por el equipo de soporte ya que como veremos más adelante OTRS permite la utilización de plantillas pero no registra su utilización en el seguimiento de los casos. Para ese 80% un tercio de las plantillas desarrolladas cubren el 80-90% de los mensajes procesados (de nuevo es una estimación de soporte). Estos dos datos nos dejan con una estimación a priori de que en el peor de los casos el 65% de todos los mensajes recibidos son procesados directamente con plantillas o una mínima edición de estas.

Ticket reservations and sale of tickets service

Sales and reservations telephone	Support service telephone	Support e-mail
902 044 454	902 044 414	correo@entradaspatrimonio.es

Contact

If you wish, you can use the following form to get in contact with us and request information, make any comment or send us a report of an incident with your purchase or reservation.

Name

Telephone*

E-mail*

Repeat e-mail*

Subject*

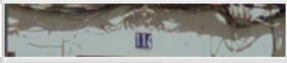
Question ▼


Venue

San Lorenzo of El Escorial ▼

Confirmation code

Message*



[Privacy & Terms](#) 

*Fields with an asterisk are obligatory.

Ilustración 1 - Formulario web actual de solicitud de información y notificación de incidencias para los usuarios del sistema de compra de entradas online

La Tabla 1 muestra el listado de las principales plantillas actualmente en uso:

Plantilla de respuesta	Descripción
Ayuda y tutoriales	Proporciona información y enlaces a video tutoriales de cómo comprar entradas o realizar reservas en anticipado.
Reservas agotadas	Informa a los usuarios que la reserva o compra que quiere realizar no es posible
No devolución	Informa a los usuarios que Patrimonio Nacional no realiza devoluciones en caso de error en la compra por parte de los usuarios.
Entradas gratuitas	Informa a los usuarios que condiciones tienen que cumplir para acceder a entradas gratuitas y cómo y dónde se pueden adquirir
Facturas	Informa a los usuarios cómo y dónde obtener las facturas asociadas a sus compras.
Visitas guiadas	Informa a los usuarios como se compran y realizas las visitas guiadas.
Anulación de reservas de colegios	Informa a los usuarios de grupos educativos como realizar la anulación de sus reservas.
Guías para colegios	Informa a los usuarios de grupos educativos como deben realizar la visita guiada.
Guías para público	Informa al público como se adquieren entradas para la visita guiada.
Bonos	Informa a los usuarios qué tipo de bonos hay disponibles y como adquirirlos
Guías para agencias	Informa cómo deben gestionar las agencias las visitas guiadas para sus grupos.
Guías para grupos culturales	Informa cómo deben gestionar los grupos las visitas guiadas.

Tabla 1 – Principales plantillas utilizadas en el servicio de atención al cliente de Patrimonio Nacional

Aunque el servicio sólo está obligado a atender mensajes recibidos en idioma español e inglés, como se puede intuir fácilmente el idioma y formato de los mensajes dada la naturaleza de la web (el sistema vende a todos los países del mundo) abarca todo el espectro posible en cuanto a vocabulario, redacción y formato. Esto incluye traducciones al español poco afortunadas de idiomas como chino o coreano.

A modo de ejemplo las Tabla 2 y Tabla 3 muestras dos mensajes de cada una de las dos fuentes posibles.

Formulario de contacto
<p>Mensaje 1</p> <pre><html> <body style="font-family:Arial;font-size:14px;color:#58585a;width:800px;"> <div style="border:solid 1px #58585a;padding:40px;">
Tipo de incidencia: Pregunta
Datos usuario: email_cliente
Nombre: nombre_cliente
Teléfono: teléfono_cliente
Email: email_cliente
Recinto: Palacio Real de Aranjuez
Localizador: 005703800C
Mensaje:
 Necesito añadir dos entradas jubilado para la misma fecha y hora (12h 30m 08/12/2016) y el sistema no me deja Espero contestación Un saludo nombre_contacto nombre_cliente

 </div> </body> </html></pre>
<p>Mensaje 2</p> <pre><html> <body style="font-family:Arial;font-size:14px;color:#58585a;width:800px;"> <div style="border:solid 1px #58585a;padding:40px;">
Tipo de incidencia: Incidencia o reclamación
Datos usuario:
Nombre: nombre_cliente
Teléfono: teléfono_cliente
Email: email_cliente
Recinto: Palacio Real de Madrid
Localizador:005701847M
Mensaje:
 Como puedo solucionar que al escribir mi correo me he equivocado en un número. Al solicitar mis entradas

 </div> </body> </html></pre>

Tabla 2 - Ejemplos de emails remitidos desde el formulario de contacto de la web de venta de entradas. Por legislación omitimos los datos identificativos del cliente.

Remitidos directamente
<p>Mensaje 1</p> <p>Queremos visitar el Palacio Real de Aranjuez el día 8/12/2016 a las doce de la mañana un grupo de 19 personas Nos comuniquen si hay que reservar, tiempo que dura la visita, costo del guía y precio de la entrada. Gracias nombre_contacto nombre_cliente teléfono_cliente</p>
<p>Mensaje 2</p> <p>Buenos días, he comprado 7 entradas para visitar el Palacio el día 6 de diciembre a las 12 de la mañana y he eliminado el mensaje con el localizador para poder imprimirlas. Por favor si pueden reenviarlas. Gracias, nombre_cliente</p>

Tabla 3 - Emails remitidos directamente por el cliente al centro de atención al cliente. Por legislación omitimos los datos identificativos del cliente.

Cómo se puede apreciar, los emails remitidos desde el formulario de contacto cuentan con un encabezado que describe la incidencia o solicitud de información a tratar para facilitar la respuesta de los operadores. Por otra parte aunque en los mensajes mostrados en la Tabla 3 no se aprecia, los mensajes remitidos por los clientes en su amplia mayoría poseen formato HTML. También en numerosos casos se incluyen firmas o citados derivados de correos previos o reenvíos.

1.2 Objetivos

En este trabajo perseguimos analizar la viabilidad de una respuesta automática de email basada exclusivamente en el contenido de los mensajes recibidos. Nuestra idea es evaluar si un planteamiento con la menor dependencia de dominio posible es viable, haciendo factible su generalización.

Nuestro trabajo se extiende a las siguientes tareas:

- Análisis del problema de la respuesta automática de email en centros de atención al cliente.
- Generalización del problema para hacer viable su posible replicación en otros dominios.
- Estudio de aproximaciones anteriores en la literatura sobre idénticas o similares problemáticas.
- Generación de un corpus sobre un caso de estudio y análisis cuantitativo del mismo.
- Construcción de diferentes aproximaciones al problema mediante aprendizaje supervisado tanto basadas en kernel como en memoria.
- Evaluación y comparativa de los resultados obtenidos por las aproximaciones sobre el corpus generado.

1.3 Estructura del documento

El apartado 2 delimita el problema abordado, mostrando sus restricciones y el modelo de interacción que sigue con los usuarios. También especifica una serie de aspectos cualitativos que consideramos clave para una posible implementación.

El apartado 3 presenta un estudio de las aproximaciones seguidas en la literatura en tareas relativas a la clasificación de email, así como una visión global de las fortalezas y debilidades de los principales enfoques utilizados.

En el apartado 4 mostraremos el proceso seguido para la extracción de los mensajes de correo utilizados en la construcción de nuestro corpus. Tareas de preprocesado aplicadas para filtrar y homogeneizar estos mensajes, construcción de un lexicón y en qué elementos de los mensajes se aplica. Cerramos el apartado con un estudio de métricas sobre el corpus final que pueden dar una idea aproximada del problema a abordar.

El apartado 5 se centra en los criterios de selección de clasificadores y las configuraciones más adecuadas a aplicar a cada uno de ellos. Como se aborda el problema de la falta de balanceo de los datos y que experimentos se programan para intentar paliar esta situación.

El apartado 6 discute los resultados obtenidos en la clasificación tanto en entrenamiento como en test. Presentamos los resultados por clase así como una visión combinada de todos.

Cerramos el documento con las conclusiones generales del trabajo así como aspectos susceptibles de mejora.

2 Definición del problema

2.1 Características del problema

Mostrado el problema general que abordamos, así como un escenario representativo del mismo, planteamos definir la respuesta automatizada de casos de soporte como una tarea de clasificación de textos supervisada con las siguientes características:

- Existe una o más respuestas que suponen un trabajo constante y repetitivo para los operadores.
- Debe existir un conjunto de plantillas predefinidas que cubran el conjunto de respuestas a automatizar. Cada plantilla deberá ser independiente del resto y sólo debe corresponderse a un tipo de respuesta.
- Los mensajes recibidos deben estar más cercanos a solicitudes de información que a resolución de incidencias. En el primero de los casos el uso de plantillas de respuesta es una solución natural. El segundo implica normalmente la necesidad de tareas de extracción de información, acciones en el sistema y redacción ad-hoc de la respuesta. Según el nivel de complejidad de estas tareas el uso de plantillas puede encajar o no.
- Las plantillas tienen un ciclo de vida largo o no expiran y no una validez temporal corta en el tiempo. En caso de tener una validez temporal está debe ser periódica durante un periodo largo.
- El proceso de clasificación se apoya en información de dominio fácilmente parametrizable mediante un lexicón que permite etiquetar elementos característicos del problema.
- La clasificación está orientada a precisión, es preferible responder pocos emails automáticamente con una tasa de acierto muy alta, que tener tasas bajas y responder automáticamente muchos emails. Esta necesidad implica la determinación de umbrales que controlen cuando se debe emitir la respuesta y cuando no.

- El proceso de clasificación tendrá en cuenta solamente información textual del asunto y cuerpo del mensaje facilitando así la generalización del problema a otros dominios. Aunque es posible disponer de información adicional vinculada al remitente, esta será ignorada.
- La clasificación es monolingüe. Se puede extender fácilmente a un número indeterminado de idiomas por replicación.
- La actualización del sistema puede ser abordada mediante batch learning. El aprendizaje online no será considerado.

2.2 Modelo de interacción

La interacción tanto con clientes como operadores es asíncrona, indisolublemente unida a la naturaleza del email, siendo innecesaria una respuesta en tiempo real. Si movemos el foco a cómo es la interacción con el usuario, el sistema se comporta como un intermediario entre los clientes y los operadores del centro de atención:

- Por cada mensaje interceptado se determina la capacidad de emitir una respuesta con un alto grado de fiabilidad y en caso afirmativo se realiza el envío de un email con la plantilla de respuesta seleccionada al cliente.
- Si no puede determinar una plantilla de respuesta el email sigue su curso habitual y llega al centro de atención donde un operador atiende el caso.
- Si el sistema no puede determinar una respuesta con alta precisión o se considera que la entrega de respuestas erróneas es crítica, el email de respuesta puede incluir una opción que permita a los usuarios reencolar el mensaje para que sea atendido por un operador.
- Los usuarios también pueden optar por la opción de reencolar si a pesar de obtener la respuesta correcta desean ser atendidos por un operador.

Tanto la recepción como respuesta y reencolado se desarrollan en un escenario asíncrono donde es necesario cubrir una limitación temporal que usualmente no debe exceder 24 horas. Sin embargo es deseable que la respuesta se emita en la menor cantidad de tiempo posible o al menos en un tiempo igual al que utilizan los operadores (en el caso de estudio inferior a 1 hora).

2.3 Aspectos cualitativos

Para terminar nos gustaría resaltar dos aspectos cualitativos que afectan a este trabajo y, aunque no van a ser tratados en el estudio, deben ser tenidos en cuenta si se llega a construir finalmente un sistema como el propuesto.

El primero de ellos está vinculado a la “frialidad” de una respuesta prefijada. Si el sistema sólo retorna una plantilla predefinida los clientes pueden percibir la respuesta como poco natural. Actualmente los operadores humanos utilizan plantillas para las respuestas, pero en bastantes casos retocan en algún grado la plantilla para adaptarla

a la incidencia recibida. Este retoque humaniza la respuesta retornada y la relaciona más estrechamente con la solicitud recibida. Este problema está fuera del ámbito de estudio.

La segunda hace referencia a que de todos los elementos que intervienen el más delicado para la implantación de una respuesta automática son las personas, independientemente de la viabilidad técnica. Un sistema sin la suficiente precisión sería percibido por los clientes como una atención deficiente, al tener que realizar más de una consulta para resolver su incidencia y como un intento de abaratar costes a costa de personal humano. En el otro lado, los operadores argumentarían que ellos podrían haber resuelto el problema en la primera respuesta y ahora tienen que tratar con un cliente potencialmente con peor actitud (o enfado según la gravedad de la incidencia) al haber tenido que perder más tiempo. A todo esto habría que añadir la preocupación implícita que supone la injerencia de un sistema que aunque pretende ser una ayuda pueda sustituir parcialmente su labor. En nuestra opinión la mejor forma de abordar o minimizar esta situación es tener el sistema de respuesta automática en un segundo plano en un entorno de producción durante un largo periodo de tiempo, de forma que se pueda cuantificar exactamente la capacidad del mismo comparando las respuestas automáticas con las de los operadores. En un periodo por ejemplo de 3 meses se puede recabar la suficiente información como para determinar con certeza si un sistema con estas características puede ser implantado.

3 Estado del arte

3.1 Primeras aproximaciones

Nuestro trabajo se centra en la clasificación de correo electrónico, un área particular dentro del campo más amplio de la clasificación de textos vinculado a disciplinas como el procesamiento del lenguaje natural (NLP, Natural Language Processing), el aprendizaje automático (ML, Machine Learning) y la recuperación de información (IR, Information Retrieval). Todas estas disciplinas han recibido gran atención y esfuerzo investigador en los últimos años ante la creciente demanda que supone la aparición de Internet y la necesidad de acceder y procesar automáticamente ingentes cantidades de información textual no estructurada. Por otro lado, y a pesar de la aparición de los servicios de mensajería instantánea, el correo electrónico sigue siendo uno de los métodos centrales de comunicación por su carácter eficaz, rápido y sobre todo gratuito. El correo electrónico es una herramienta básica en la operativa de muchas empresas e indispensable en centros de atención al cliente. Desgraciadamente el email también tiene una contrapartida, el esfuerzo que supone para los usuarios atender y gestionar diariamente centenares de correos en sus bandejas de entrada.

Bajo este escenario surgen los primeros trabajos vinculados a la clasificación de emails, que como veremos están segmentados en dos enfoques principales:

1. Ayudar a los usuarios a clasificar el correo recibido en carpetas (categorías), de forma que estos puedan eliminar la tediosa y repetitiva tarea de clasificar manualmente o determinar que atender primero.
2. Filtrar el correo basura o SPAM. La eliminación de SPAM se ha convertido con los años en un problema de proporciones descomunales lo que ha centrado la atención de muchos investigadores, encontrando numerosos trabajos en la literatura sobre este asunto.

Estas dos líneas de trabajo no son excluyentes, el filtrado de SPAM es un caso particular de clasificación binaria y puede ser un paso previo a la clasificación en categorías. Si establecemos una analogía vemos que nuestra propuesta puede encajar en cualquiera de las dos líneas. Por un lado la selección entre unas carpetas de destino es el equivalente a la selección de una categoría (plantilla) de respuesta, aunque en nuestro caso con categorías mucho más concretas. Por otro lado, al igual que para el SPAM el proceso completo se puede ver como n decisiones binarias donde n es el número de categorías de respuesta automática. En el primer caso tendríamos un sistema con una única respuesta con n valores posibles, en el segundo n respuestas si/no que deberían ser combinadas para obtener la respuesta final.

Retomando el contexto histórico de las investigaciones en el campo, los primeros trabajos en el ámbito de la clasificación de email se centraron en sistemas de reglas sobre palabras clave, creadas manualmente o inducidas, y en las técnicas de similitud de documentos heredadas del campo de la recuperación de información (IR, Information Retrieval) fundadas en la comparación de representaciones vectoriales de los mensajes.

Uno de los primeros trabajos en clasificación de email es el de Cohen [2] donde compara dos técnicas, el tradicional enfoque de IR con TD-IDF y RIPPER, un algoritmo para la inducción de reglas que calculan palabras clave (keywords) representativas de cada categoría a clasificar. RIPPER es una mejora de un algoritmo previo de inducción de reglas de nombre IPER que el propio Cohen presentó en [1] con la intención de igualar capacidad de clasificación obtenida por el algoritmo C4.5 pero buscando un menor coste computacional. El trabajo muestra una capacidad de clasificación muy similar entre ambos sistemas, con la característica de que RIPPER consigue inducir muy buenos conjuntos de reglas con números relativamente pequeño de ejemplos (unos pocos cientos). Aunque los resultados de ambas técnicas son muy similares Cohen defiende la ventaja de RIPPER al ser el conjunto de reglas más fácilmente interpretable y editable por los usuarios. A diferencia del trabajo aquí presentado que se centra exclusivamente en asunto y cuerpo del mensaje Cohen incluye remitente y destinatario como elemento de entrada a los clasificadores. De hecho, en las implementaciones en sistemas de filtrado o clasificación de correo estos suelen ser atributos habituales del proceso de clasificación ya que es información muy relevante,

especialmente con el SPAM. En nuestro trabajo proponemos obviarlos y que el sistema de decisión se centre únicamente en la información textual porque creemos que este enfoque es más versátil y porque no siempre se va a contar con la información de un remitente, por ejemplo un formulario de contacto público online, o porque existe un número muy alto de clientes que usan el servicio de soporte en una única ocasión.

MAILCAT [18] es un ejemplo de la utilización de las técnicas de IR en un complemento para un producto comercial que clasifica los emails en las 3 categorías más probables. Este sistema usa una modificación TF-IDF donde cada clase es representada por una ponderación de los vectores de la misma y la similitud es una variante del tradicional coseno de IR. El rasgo más característico es que los autores modifican el algoritmo para usar aprendizaje online (Online Learning), al necesitar una adaptación continua a los nuevos emails recibidos por el usuario. El autor muestra tasas de exactitud entre el 80-90%. Estas tasas aunque pueden ser altas en otros contextos son insuficientes para una respuesta automática de email.

Aproximadamente en las mismas fechas, y de nuevo más próximo a IR que a ML, basado en CBR (Case Base Reasoning) está FALLQ [4]. Se trata de un sistema de recuperación de documentos para una empresa del sector del software de telecomunicaciones. Cada unidad de información almacenada en el sistema (caso) sigue una estructura de terna formada por el texto libre de la pregunta del usuario, atributos adicionales y el texto libre de respuesta del centro de soporte. Para relacionar peticiones con casos existentes se realiza un parsing de los casos para construir lo que en el autor denomina entidad de información (IE). Un IE es sólo una forma más general de referirse a una palabra, concepto o término clave, pero que puede encapsular variaciones morfológicas, sinónimos, abreviaturas, expresiones o referencias en otros idiomas al mismo concepto. Para esta construcción se necesitan listas de palabras claves que deben ser proporcionadas al sistema. Una métrica de similitud, que ha sido definida en base a similitud lingüística de palabras obtenidas de un tesoro, permite relacionar los conjuntos de IEs de peticiones y la base de datos de casos.

Existe una analogía clara entre recuperar un documento para entregar a un usuario y la selección de una plantilla para responder un email. A diferencia de nuestro enfoque general, FALLQ se desarrolla como ayuda al centro de atención de la empresa (las decisiones finales pasan por un operador) y no como una herramienta para sustituir o descargar parcialmente el trabajo de los empleados. No proporciona datos numéricos respecto a la precisión y cobertura del sistema y deja demasiadas problemáticas abiertas.

Ligeramente más tardío pero muy interesante e implantado finalmente en una agencia de noticias económicas en Italia, es el sistema FACILE presentado en [5] donde se

integran técnicas de extracción de información (IE, Information Extraction) con coincidencia de patrones (PM, Pattern Matching) con el objetivo de conseguir una clasificación de textos adaptable. PM se usa para realizar una clasificación superficial y lograr identificar características generales de los textos para centrar sin mucho detalle el área al que pertenecen. Para esta decisión el autor se apoya en trabajos previos [6] donde se demuestra que PM puede producir buenos y rápidos resultados de clasificación sobre conjuntos de clases grandes. Una vez conocida el área aproximada IE refina la categorización final. La mayor problemática que presenta este sistema es la dependencia tanto en la etapa de PM como de IE de la incorporación de conocimiento externo lo que puede elevar muchos los costes de puesta en producción y mantenimiento. En el caso de PM, usuarios sin una información específica en lingüística pueden proporcionar esta información, mientras que para la parte de IE se necesita formación específica para configurar el sistema. Además, el sistema se estructura en etapas presentando los resultados de precisión y cobertura de forma aislada para cada etapa y no de forma global para todo el sistema.

No disponemos de información sobre si el sistema FALLQ ha llegado a estar en producción, ni durante cuánto tiempo y con qué éxito lo ha hecho FACILE.

Los trabajos anteriores estarían dentro de una primera época pre año 2000 donde todavía las técnicas de aprendizaje automático (ML, Machine Learning) habrían hecho una aparición reducida en el campo de la clasificación de email. El trabajo de Sahami [8] marca un primer acercamiento a un enfoque basado en ML, con la primera referencia académica a la utilización de un clasificador bayesiano para el filtrado de SPAM. Sahami muestra como Naive Bayes (NB) puede obtener unos excelentes resultados. A pesar de estos buenos resultados, son los trabajos de Paul Graham [9] y [10] los que definitivamente consiguen consolidar la utilización de filtros bayesianos para la clasificación del correo, especialmente debido a la reducción de falsos positivos. Los falsos positivos son el mayor problema en la clasificación de SPAM ya que implica que correo legítimos vayan a parar a la bandeja de descartes, siendo este el caso que mayor perjuicio genera al usuario. Entre estos trabajos podemos encontrar también el sistema IFILE [11], este sistema también presenta NB multinomial sobre bolsas de palabras (bag of words) pero no para filtrar SPAM sino para sugerir las categorías de email (carpetas) más probables al usuario. También SPAMCOP [19] utiliza NB para filtrar SPAM con resultados altos pero no suficientes. Lo más interesante de SPAMCOP es que entre sus experimentos plantea una alternativa donde utiliza trigramas de letras y no de palabras obteniendo unos resultados muy similares al enfoque más clásico de NB. El autor también muestra la fuerte variación en la clasificación cuando los conjuntos de datos no están balanceados (pero no relaciona el problema con el clasificador) y cómo las técnicas bayesianas pueden ofrecer muy buenos resultado con pocos datos (unas pocas decenas de mensajes).

Todos estos trabajos tienen rasgos comunes:

- Tienen tasas de clasificación altas sin la necesidad de muchos datos de entrenamiento (aunque no siempre suficientes para un sistema en producción).
- Más capacidad de generalización siendo mucho menos inestables que los sistemas basados en reglas, muy afectados por cambios en los contenidos textuales de los mensajes.
- Las técnicas bayesianas pueden ofrecer una probabilidad de salida al clasificar, presentando una clara ventajas sobre palabras clave que sólo ofrecen una respuesta binaria.
- Consiguen mejores tasas de exactitud mediante la inclusión de información de dominio o relacionada. En el caso de Sahami utiliza información de sistemas de filtrado de palabras clave basado en reglas para inclusión de más características al clasificador, de igual forma utiliza información del remitente, números de adjuntos, etc. Graham por su parte es un gran conocedor del problema del SPAM y sus excelentes resultados vienen de no buscar un clasificador puro, sino de sesgar la entrada para evitar la clasificación de falsos positivos o evitar prácticas de los spammers para reducir la utilidad de los filtros.
- Aplican técnicas de reducción de la dimensionalidad y/o stop words. Un problema recurrente en el análisis de textos cuando se enfoca cada token como una característica de entrada a un clasificador es la enorme dimensionalidad que presenta el problema, habitualmente hablamos de miles o decenas de miles de características. Sahami elimina palabras de baja frecuencia y computa el coeficiente de información mutua (MI, Mutual Information) entre cada característica y la clase, *estima* que los mejores 500 atributos son suficientes para obtener buenos resultados. IFile al tratarse de un sistema pensado para ser embebido en un cliente de correo utiliza una reducción con carácter temporal, eliminar palabras más antiguas con baja frecuencia, mientras conserva nuevas con baja frecuentes y antiguas con alta.
- Menor coste. Con inducción de reglas, los costes de producción manual de clasificadores basados en palabras clave se disparan.

Estas características fueron el punto de partida para que NB, especialmente la variante multinomial sobre bolsas de palabras, fuera rápidamente adoptado en muchos productos comerciales tanto de filtrado de SPAM como de clasificación de textos. Su fácil implementación, rápido entrenamiento, rápida respuesta con en general alta precisión, poca tendencia al sobreajuste y la facilidad de actualizar el modelo estadístico subyacente de forma incremental favorecieron su adopción. No obstante, y a pesar del éxito, en la misma época otros trabajos mostraban la bondad de otras técnicas de clasificación con iguales o mejores resultados, especialmente las máquinas de vectores soporte (SVM – Support Vector Machines) [28].

Es curioso que a la vez que el uso de NB se popularizaba y mostraba su eficacia, trabajos que comparaban distintas técnicas de clasificación de textos pusieran de manifiesto sus limitaciones. [20] muestra muy claramente estas deficiencias a la vez que propone métodos para paliarlas, acercándose mucho a los resultados obtenidos por SVMs en conjuntos de datos estándar. El estudio referencia varias debilidades sistémicas de la versión más extendida de NB, la variante multinomial (MNB):

- Resultados sesgados por datos no balanceados en las clases de entrenamiento lo que mueve más masa de probabilidad hacia las clases dominantes.
- Resultados sesgados por vectores de pesos que pueden presentar variaciones importantes en módulo debido a que la dependencia entre características no tiene que ser la misma dentro de todas las clases.
- MNB asume un modelo multinomial del texto que no necesariamente tiene que corresponderse al modelo que sigue un texto real a clasificar.
- Diferentes longitudes de mensajes afectan a la clasificación final.

Es muy interesante el planteamiento de este trabajo ya que en vez de descartar la técnica frontalmente, es consciente de que en ML las técnicas con menores errores suelen ser las más costosas computacionalmente y tras aplicar las correcciones oportunas a NB se sigue teniendo un algoritmo muy eficaz. La mayor desventaja que suponen las correcciones aportadas es que MNB pierde la capacidad de que el clasificador sea adaptado a aprendizaje online de forma sencilla.

Retomando las alternativas a NB, posiblemente [14] es el primer trabajo en utilizar SVM para la clasificación de SPAM. El autor muestra alguna de las ventajas de utilizar SVMs en comparación con Ripper, Rocchio y Boosting Decision Trees:

- Menor sensibilidad a conjuntos de datos no balanceados.
- Mejor dispersión del error
- Rápidas en ejecución
- Sólo es necesario ajustar un único parámetro para las SVM lineales.

Otro rasgo diferencial en el uso de SVMs es que el propio algoritmo determina si una característica (palabra) es determinante o no, eliminando la necesidad de utilización de listas de stop words o lematización. Sin embargo, el autor utiliza lematización para favorecer la capacidad de generalización y elimina términos con menos de 3 ocurrencias para obviar errores ortográficos. También comenta la influencia en la generalización del modelo que puede tener la reducción de dimensionalidad previa, aunque siempre aumentando el coste global de construcción del clasificador. En este trabajo la representación binaria de características es la que obtiene mejores resultados (no se utiliza la frecuencia de aparición de una palabra sólo si esta aparece o no). En la parte de inconvenientes menciona los altos tiempos de entrenamiento

para grandes conjuntos de datos cuando se usan SVMs no lineales. En ese sentido la introducción de la técnica SMO (Sequential Minimal Optimization) [30] dio un gran impulso a esta técnica al reducir notablemente los tiempos de entrenamiento.

Otro trabajo que compara SVM, TF-IDF y un modelo del lenguaje con unigramas en la clasificación de email en categorías es [17]. Según los autores la elección de SVM no es aleatoria y un análisis preliminar les llevo a elegir esta técnica tras una comparación directa con redes neuronales (perceptrón de capa única), árboles de decisión y LDA (Linear Discriminant Analysis). El estudio pone de manifiesto la necesidad de evolución temporal de los clasificadores según se va recibiendo nuevo email y como algunas técnicas directamente no se adaptan a este escenario. También destaca la importancia del coste computacional de las mismas. Extiende el vector de características a la cabecera, asunto y cuerpo aplicando reducción de dimensionalidad (que estima empíricamente en 1000 atributos para SVM). Plantea un clasificador binario por carpeta donde cada clasificador (SVMs) ofrece una probabilidad de salida que puede ser ordenada. En los resultados obtenidos en el trabajo los autores aplican el proceso de clasificación diferentes conjuntos de emails mostrando resultados interesantes:

- Ningún clasificador es consistentemente mejor que el resto. Este es un dato sorprendente porque pone en entredicho la superioridad de SVM reportada por otros autores.
- La tasa de aciertos varía más entre los conjuntos de emails de entrenamiento que entre clasificadores.
- SVM se comporta mejor cuando las categorías finales son más densas (más número de mensajes), mientras que cuando son más dispersas TF-IDF tiene una ligera ventaja.
- Si se incluyen las cabeceras en la clasificación existe poca diferencia en proporcionar o no el cuerpo del mensaje al clasificador.

Los resultados de este trabajo son de poca utilidad para un sistema automático de respuesta ya que la evaluación usada para medir un “acierto” es que el email clasificado pertenezca a uno de las 5 primeras categorías que los clasificadores retornan en la salida (la salida está ordenada por probabilidad). Por otro lado que la eliminación del cuerpo del mensaje (para reducción de coste computacional) no suponga un elemento diferencial en la precisión nos hace intuir que hay una fuerte dependencia del remitente en los conjuntos de datos evaluados y/o hay poca variabilidad en el asunto en cada categoría.

Además de NB y SVM otros investigadores plantearon alternativas para la clasificación mediante técnicas basadas en memoria y no en kernel. Así [15] realiza una comparación de NB y técnicas basadas en memoria (MBL), en concreto una variante de KNN (K-Nearest Neighbor) sobre un corpus estándar para de nuevo abordar la tarea

de filtrar SPAM. Para la comparativa sigue el mismo planteamiento de [8], utiliza un representación vectorial binaria de los atributos (1 si la palabra existe en el mensaje, 0 en caso contrario), reduce la dimensionalidad utilizando MI y lematiza los mensajes. A diferencia de [8] el autor incluye en su trabajo la influencia del número de atributos en la clasificación. Un rasgo muy interesante de este trabajo es que repite el enfoque seguido de ponderar la salida del clasificador. Como comentamos anteriormente el error de clasificar un email legítimo como SPAM es mucho más severo que el de clasificar SPAM como legítimo. Dado que para cada instancia el clasificador puede otorgar una probabilidad de salida, sólo se clasifica una entrada como SPAM si la razón entre ambas probabilidades es superior a un umbral prefijado λ :

$$\frac{Prob(spam)}{Prob(legítimo)} > \lambda$$

Cuanto mayor la constante más precisión alcanza el sistema a costa de una menor cobertura. Tanto NB como KNN muestran en el estudio resultados prácticamente idénticos, sin embargo el autor referencia una ligera ventaja para las técnicas MBL cuando el valor de λ es muy alto. [16] extiende esta comparación incluyendo SVM lineales de nuevo con la ponderación de resultados para minimizar la clasificación de email legítimos como SPAM. Sus conclusiones son claras: SVM son mejores cuando el ajuste de λ se mantiene en un rango bajo-alto, pero cuando es necesario que sea muy alto NB se comporta mejor. Al contrario que en [15] en este caso las técnicas MBL quedan claramente por detrás para λ alto. Sin embargo otros autores muestran que para valores de λ alto los resultados de NB no son estables y que los datos presentados no se pueden generalizar.

[24] realiza un estudio exhaustivo de 14 técnicas de clasificación de textos (no email) sobre corpus estándar con miles de categorías y KNN (MBL) muestra unos excelentes resultados con la menor degradación cuando el número de categorías crece en varios órdenes de magnitud.

3.2 Visión global

Tras la revisión de trabajos anterior creemos que podemos establecer analogías claras entre los estudios realizados en el área y este trabajo mediante un planteamiento que permita abordar nuestro problema con cierta confianza, al menos a priori. Recordamos que el principal objetivo de este estudio es conseguir una respuesta automática de email con muy alta precisión, una alta cobertura, aunque deseable, es un factor secundario en esta fase.

La primera analogía es que podemos relacionar las altas tasas de precisión en la clasificación de SPAM con procesos de clasificación binaria. Aunque el SPAM tiene una naturaleza inherentemente binaria, creemos que un planteamiento de n clasificadores

binarios, uno por cada posible plantilla de respuesta, ayudaría a conseguir una mayor precisión.

- Con un enfoque binario podemos usar más instancias, o al menos disponer de un número mucho mayor de muestras para cada categoría. De esta forma la clasificación como SPAM equivale a seleccionar una respuesta automática y la no respuesta sería equivalente a la detección de mensajes legítimos. Sabemos que más instancias para entrenamiento y test conllevan de forma general mejores clasificadores (tratamos el problema del balanceo más adelante).
- El enfoque de n clasificadores creemos que aporta más flexibilidad ya que no restringe la utilización de un único tipo de clasificador o configuración para todas las clases. Según el volumen y características de los datos se podría optar por diferentes técnicas o configuraciones en busca de mayor precisión.
- A priori n clasificadores nos permitirían cubrir un mayor número de clases con más solvencia. En nuestro problema actualmente contamos con aproximadamente 20 clases, pero en otros dominios este número podría ser mucho mayor.
- Si el clasificador o clasificadores empleados pueden entregar una medida de probabilidad podemos utilizar este valor como una forma de controlar directamente la precisión final del sistema. Como hemos visto en varios trabajos, consideraríamos la respuesta automática cuando la relación entre la probabilidad de pertenencia a la clase (SPAM) y la de no pertenencia (legítimo) superase un umbral.
- El valor de probabilidad en la salida se puede utilizar para combinar los resultados parciales.

Obviamente el enfoque de n clasificadores binarios también tiene numerosas desventajas, pero consideramos que si estas desventajas son abordables la precisión para nosotros es más importante. Entre los inconvenientes podemos citar:

- Es necesario entrenar n clasificadores, mayor coste. Si hay más de un tipo o configuración el coste es aún mayor.
- El clasificador debe ofrecer un nivel de certeza en la salida, sino debe adaptarse para que genere uno.
- Es necesario ajustar n umbrales de decisión.
- La acumulación de todas las instancias para entrenar cada clasificador necesariamente conlleva conjuntos de datos no balanceados y sabemos que las técnicas vistas hasta el momento no se comportan bien con estos tipos de conjuntos de datos.

Además es necesario implementar algún mecanismo de combinación de las salidas de los n clasificadores, si más de un clasificador da positivo para un mensaje. Planteamos tres opciones simples para esta combinación.

- Elegir la clase más probable.
- Elegir la clase que retorne el grado de certeza más alto en clasificación.
- Descartar la respuesta automática.

Si lo que buscamos es únicamente alta precisión, a priori la opción más adecuada en caso de que más de un clasificador de positivo es no enviar respuesta. Una situación así puede ser un síntoma de que los modelos construidos para cada clasificador se solapen espacialmente y es mejor evitar emitir una respuesta automática para el mensaje clasificado.

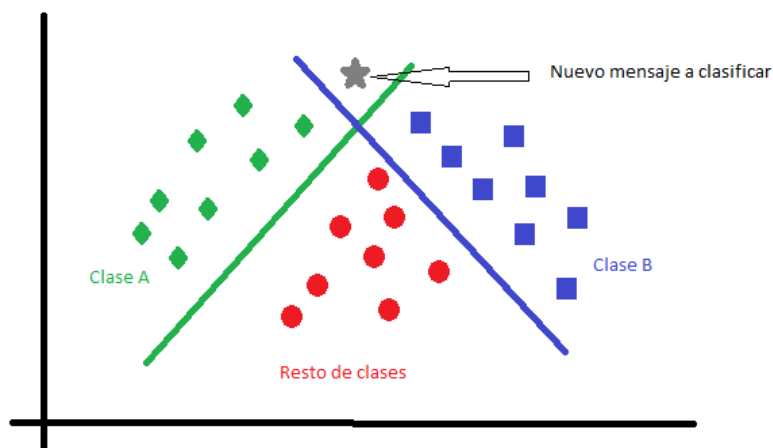


Ilustración 2 - Intuición del concepto de solapamiento espacial entre los n clasificadores binarios. La línea verde muestra el criterio del clasificador para la clase A, la azul para la clase B. Aunque ambas separan perfectamente sus muestras no hay garantía de que una muestra no vista caiga en un espacio que ambas clases consideran un positivo.

La Ilustración 2 muestra esta idea de solapamiento espacial sobre un espacio bidimensional y clasificación lineal. No obstante un análisis de los resultados combinados de todos los clasificadores puede arrojar más luz sobre el mejor método de composición.

Otro factor que ha recibido poca atención a lo largo de los trabajos revisados es la cantidad de datos necesario para cada clasificador. Hemos podido ver como algunos trabajos referencian que NB (o MNB) puede obtener valores altos de exactitud con un número de muestras bajo, mientras otros hacen ver que SVM necesita una mayor densidad de datos [17] para lograr un buen entrenamiento, así como un número de características más alto [16]. De hecho es ampliamente aceptado que SVM suele necesitar una cantidad de datos mayor que NB para ofrecer buenos resultados [21]. Sin embargo, no creemos que estos resultados sean suficientemente representativos

para generalizar. Por poner un contraejemplo sencillo: un conjunto con un número pequeño de instancias y que modela muy bien la población, linealmente separable gracias a un conjunto pequeño de características representativas, seguramente permita obtener un clasificador prácticamente óptimo. Es decir, al final es el conjunto de datos es el que marca la diferencia. Trabajos como [13] muestran como de forma general el tamaño del dataset de entrenamiento puede tener mucha más influencia que el clasificador elegido. Es más, con suficientes datos las mejoras obtenidas dejan en entredicho sistemas basados en votación (voting), donde varios clasificadores actúan coordinadamente para mejorar los resultados parciales de cada uno de ellos [22].

Respecto a la problemática de los datos no balanceados sabemos que tanto NB como KNN y SVM presentan problemas bajo esta circunstancia, de hecho prácticamente todos los clasificadores se ven afectados. Hay numerosas soluciones en la literatura para intentar paliar estos problemas que pasan por un amplio espectro, desde muy simples, como por ejemplo modificar las prioridades a priori de las clases en NB o incrementar la penalización del error en SVM para una de las clases; a soluciones complejas que modifican la estructura o el comportamiento interno de los clasificadores. Otro enfoque al problema de los conjuntos no balanceados sería abordarlo desde la perspectiva de los datos. Existen diferentes algoritmos que permiten realizar un re muestreo de los conjuntos de datos para su rebalanceo. Este re muestreo puede ser tanto ascendente (oversampling) como descendente (undersampling). Habitualmente se combinan ambos, el oversampling genera nuevas muestras sintéticas para la clase minoritaria utilizando como referencia las instancias existentes, mientras que el undersampling elimina muestras de las clases mayoritarias. SMOTE es uno de los algoritmos de oversampling más utilizados para este propósito pero no el único.

Llegados a este punto, lo ideal sería poder obtener el clasificador que mejor cubra los siguientes aspectos de forma ordenada:

1. Alta precisión. Aunque algún trabajo muestra buenos resultados, en general se considera que KNN, en cuanto a precisión para clasificación binaria, está por detrás de MNB y SVM. No obstante, hemos de tener en cuenta que ninguno de los estudios presentados evalúa la precisión teniendo en cuenta una reducción en la cobertura.
2. Capacidad de tratar con conjuntos de datos no balanceados. Las tres técnicas se ven afectadas, quizás la peor en este caso es NB. Aquí creemos que el re muestreo a nivel de conjuntos de datos es la mejor opción
3. Certeza como valor de salida en la clasificación. NB presentan de forma nativa, SVM con el uso de calibradores y KNN necesitaría una adaptación para poder asociar un valor de certeza.

Otro apartado muy relevante en la clasificación de textos es el procesamiento previo de los datos con la intención de reducir la dimensionalidad de los mismos. Al contrario que ocurre con otro tipo de problemas, la clasificación de textos presenta una dimensionalidad muy alta que eleva los costes computacionales e introduce gran cantidad de ruido en los procesos de clasificación. Un preprocesado de los textos, o de las representaciones que se hayan elegido, ayuda a reducir estos costes y mejorar la capacidad de generalización de las técnicas. Hemos visto a lo largo de los estudios técnicas recurrentes para este fin, siendo las más habituales:

- Eliminación de stop words.
- Lematización.
- Eliminación de términos de baja frecuencia
- Mutual Information (MI)

Otras técnicas como Document Frequency (DF), Information Gain (IG), χ^2 -text o Term Strength son también técnicas más sofisticadas de reducción. Especialmente DF e IG han reportado muy buenos resultados [25].

Otro factor que ha mostrado ser una gran ayuda en la capacidad de generalización de los clasificadores es la utilización de un lexicón de propósito general o vinculado al dominio. Así podemos tener lexicones de carácter más general que permitan el reemplazo por etiquetas en el texto de número, fechas, importes, etc. u otros más orientados a dominio. En el segundo caso para nuestro problema ejemplos serían: nombre de museos (“Palacio Real”), tipos de tarifas (“Reducida”), identificación de tamaños de grupo (“25 personas”), etc. De nuevo experimentos previos realizados con esta técnica en el caso de estudio han conseguido incrementos importantes en la precisión de los resultados.

4 Corpus de entrenamiento y test

En esta sección describimos el desarrollo de un corpus específico para el caso de estudio presentado, el canal de correo electrónico del centro de atención al cliente del sistema de venta de entradas de Patrimonio Nacional.

Los emails a ser clasificados por el sistema provienen de dos orígenes diferentes: emails emitidos desde el formulario de contacto online y emails enviados directamente a la dirección de soporte. En ambos casos los emails son encolados hacia un sistema de gestión de tickets para ser atendidos por los operadores. En nuestro dominio, Patrimonio Nacional, el servicio utiliza la herramienta OTRS, un software de código abierto para la gestión y seguimiento de tickets de soporte que almacena tanto los mensajes de los clientes como las respuestas de los operadores en una base de datos relacional (MySQL).

4.1 Extracción de los mensajes

El conjunto de datos ha sido extraído directamente de la base de datos del sistema de atención al cliente (OTRS). Para ello se ha desarrollado un software a medida que se conecta directamente al almacenamiento relacional utilizada por OTRS y extrae de las tablas mensajes formados por pares pregunta-respuesta. Este software acepta una descarga incremental proporcionándole un id de ticket a partir del cual se quiere realizar la extracción.

Los mensajes obtenidos se corresponden a aproximadamente a un periodo temporal de 1 año y fueron extraídos en dos fases. En la primera fase se extrajeron aproximadamente 3000 mensajes no etiquetados. Esta falta de etiquetas es debida a que el sistema de tickets permite definir plantillas (clases) de respuesta que los operadores pueden seleccionar de forma simple a la hora de responder, sin embargo no registra internamente esta asignación, de forma que no es posible relacionar la respuesta emitida con una clase concreta (plantilla). Para etiquetar estos mensajes se ha seguido un procedimiento automático que utiliza una métrica de similitud de textos para identificar la plantilla más similar a la respuesta redactada por el operador. En concreto hemos utilizado el coeficiente de solapamiento o coeficiente Szymkiewicz-Simpson junto con un umbral de 0.95.

$$Overlap(X, Y) = \frac{|X \cap Y|}{\min(|X|, |Y|)} > 0.95$$

El coeficiente genera una salida en el intervalo [0,1] donde 0 indica ningún solapamiento y un valor de 1 un solapamiento total de los textos. Aunque el umbral es muy alto y deja muchos mensajes fuera que podrían haber sido incluidos, preferimos garantizar una asignación de etiqueta correcta que contar con más información a riesgo de que esta sea imprecisa. Mediante este proceso se etiquetaron aproximadamente 1300 mensajes.

En la segunda fase para evitar volver a necesitar este proceso de etiquetado automático y evitar perder una cantidad importante de información, modificamos las plantillas dentro del sistema OTRS para asignarles un código textual al final de las mismas. De esta forma, a posteriori, siempre que un operador selecciona una plantilla podemos identificar cual es la plantilla utilizada mediante el análisis del texto enviado con expresiones regulares. Con este procedimiento hemos extraído aproximadamente otros 2000 mensajes.

El dataset resultante cuenta con 3309 mensajes, siendo cada mensaje un par pregunta-respuesta. La Tabla 4 muestra el número de mensajes por categoría así como el peso que tiene en el conjunto de datos final. Como se puede apreciar las 5 primeras clases cubren el 71,20% de los mensajes extraídos.

Clase (plantilla de respuesta)	Nº instancias	Peso	Acumulado
03-PATRIMONIO_AyudaTutoriales	1117	33,76%	33,76%
33-PATRIMONIO_ReservasAgotadas	352	10,64%	44,39%
25-PATRIMONIO_NoRefund	328	9,91%	54,31%
15-PATRIMONIO_EntradasGratis	325	9,82%	64,13%
18-PATRIMONIO_Facturas	234	7,07%	71,20%
42-PATRIMONIO_VisitasGuiadas	147	4,44%	75,64%
11-PATRIMONIO_ColegioAnularReserva	140	4,23%	79,87%
36-PATRIMONIO_GuiasColegios	98	2,96%	82,83%
22-PATRIMONIO_GuiasIndividual	97	2,93%	85,77%
05-PATRIMONIO_Bonos	96	2,90%	88,67%
17-PATRIMONIO_GuiasAgencias	57	1,72%	90,39%
39-PATRIMONIO_GuiasCulturales	48	1,45%	91,84%
20-PATRIMONIO_Fuentes	43	1,30%	93,14%
30-PATRIMONIO_Reclamaciones	41	1,24%	94,38%
31-PATRIMONIO_RecuperarContraseña	34	1,03%	95,41%
32-PATRIMONIO_Reservas	24	0,73%	96,13%
40-PATRIMONIO_Visita Especial	19	0,57%	96,71%
19-PATRIMONIO_Faltan_Datos	16	0,48%	97,19%
21-PATRIMONIO_GratuidadesExtra	13	0,39%	97,58%
06-PATRIMONIO_CartaPresentación	12	0,36%	97,94%
10-PATRIMONIO_GuiasTuristicos	11	0,33%	98,28%
16-PATRIMONIO_EntradasReducidas	9	0,27%	98,55%
38-PATRIMONIO_Stradivarius	9	0,27%	98,82%
26-PATRIMONIO_NoRefundENG	7	0,21%	99,03%
04-PATRIMONIO_AyudaTutorialesEng	6	0,18%	99,21%
41-PATRIMONIO_VisitaIndividual	6	0,18%	99,40%
07-PATRIMONIO_Casitas	5	0,15%	99,55%
08-PATRIMONIO_CertificadoWeb	4	0,12%	99,67%
14-PATRIMONIO_DarBajaUsuario	4	0,12%	99,79%
01-PATRIMONIO_AccesoPreferente	3	0,09%	99,88%
34-PATRIMONIO_ReservasAgotadasEng	2	0,06%	99,94%
02-PATRIMONIO_AranjuezParticularidades	1	0,03%	99,97%
24-PATRIMONIO_InfoValle	1	0,03%	100,00%
23-PATRIMONIO_Huelgas	0	0,00%	100,00%
28-PATRIMONIO_Photos	0	0,00%	100,00%
37-PATRIMONIO_SendaRiofrio	0	0,00%	100,00%

Tabla 4- Número de instancia por clase de respuesta y peso respecto del total de mensajes utilizados

4.2 Normalización de los mensajes

Antes de utilizar los mensajes en el proceso de construcción de los clasificadores es necesario un proceso de normalización de los mismos. Las motivaciones para llevar a cabo esta tarea son múltiples:

- El formato en el que se recibe el email puede ser tanto textual como HTML. El marcado HTML no aporta información significativa a la clasificación así que debe ser eliminado.
- Para los clientes que son empresas o instituciones los emails recibidos pueden contener firmas correspondientes al nombre de la empresa, razón social, información de contacto, normativa legales, etc. Es necesario también eliminar esta información.
- Eliminamos también los encabezamientos y despedidas formales ya que no son significativos.
- Los correos pueden citar uno o más correos anteriores donde cada correo puede corresponderse a diferentes respuestas o solicitudes. Eliminamos el texto citado, la relación temporal de mensajes no es tenida en cuenta.

Existen otra serie de tareas que son recurrentes en toda la literatura sobre clasificación de textos y que aplicamos también en este trabajo.

- Homogeneización de textos. Reducción a minúsculas, eliminación de signos de puntuación y caracteres utilizados como separadores. El mensaje se convierte a una secuencia de palabras separadas por blancos.
- Eliminación de palabras con alta frecuencia independientes de la clase final conocidas como stop words.
- Eliminación de palabras con baja frecuencia (por ejemplo palabras derivadas de errores de escritura).
- Identificación y etiquetado de información independiente de dominio: fechas, horas, números, cantidades monetarias, etc.
- Identificación y etiquetado de entidades dependientes de dominio: nombres de museos, nombres de precios, tamaños de grupo, etc.
- Detección del idioma.

El orden en que se aplican los procesos anteriores en el preprocesado de los mensajes es el siguiente:

Orden	Descripción del filtrado
1	Eliminación de marcado HTML de la respuesta. Para este paso se ha utilizado la librería HtmlAgilityPack. Mediante un proceso iterativo se recorren todos los nodos con un contenido textual, se extrae el texto de cada uno de ellos y se concatena. Antes de retornar el texto concatenado se aplica una decodificación del texto para transformar las HTML Entities a su carácter asociado en la codificación usada.
2	Eliminación de los citados de las respuestas emitidas por los operadores. Este es el caso más sencillo ya que todos los mensajes emitidos a partir de una plantilla tienen una despedida común. Se identifica la posición de esta despedida y se elimina todo el contenido de ese punto en adelante.
3	Detección de la categoría de la respuesta. Primero se aplica una expresión regular al mensaje para la detección de la existencia del código de identificación de plantilla, si se detecta este código se asigna la clase etiquetada. Si no se detecta el código se cuantifica la similitud del texto con cada una de las plantillas utilizando el coeficiente de solapamiento. Si el valor de similitud más alto está por encima del umbral preestablecido (0.95) el mensaje se etiqueta como perteneciente a esa categoría.
4	Detección del idioma de la pregunta. Usamos una librería que permite discriminar con bastante exactitud (no es perfecta) el idioma de los emails. Si el idioma es distinto del español lo descartamos.
5	Eliminación de HTML del contenido de la pregunta. Idéntico al punto 1.
6	Eliminación de despedidas, firmas y citados de las preguntas de los clientes. Mediante exploración visual de aproximadamente 300 mensajes se extrajo en primera instancia un pequeño conjunto de textos comunes utilizados como despedidas en los correos. Estos textos se utilizan como marcadores de posición. Una vez detectada su ubicación en el texto se elimina todo el contenido de su posición en adelante. Según se van agregando nuevos mensajes al dataset está lista necesita ser revisada o ampliada periódicamente. La lista está disponible en los anexos de este documento.
7	Eliminación de las cabeceras incluidas en los emails recibidos desde el formulario de la web. Se realiza mediante una expresión regular.
8	Aplicación del lexicón de dominio a los mensajes. Básicamente el lexicón utilizado es un conjunto de expresiones regulares que se utilizan para realizar búsquedas dentro del cuerpo de los mensajes y reemplazar con una representación de más alto nivel los patrones encontrados por etiquetas.
9	Eliminación de todo tipo de tildes, diéresis y caracteres no utilizados para representar texto como parte del juego de caracteres básico. En este paso también se transforma todo el texto a minúsculas.
10	Tokenización del texto. Para esto proceso utilizamos los filtros de tokenización de la librería OpenNLP.
11	Finalmente el texto del asunto y el cuerpo del mensaje procesado es concatenado.
12	Eliminación de los términos con frecuencia inferior a 5. Esto elimina términos no relevantes o que son consecuencia de errores de escritura.

Tabla 5- Procesos de filtrado de mensajes para la construcción del corpus

El resultado final es una lista de palabras en minúscula y separadas por espacios en blanco. Es importante recalcar que un mensaje para nosotros es un par pregunta-respuesta y que la etiqueta de clase se asigna a partir de la respuesta del operador mientras que el corpus de entrenamiento y test se genera desde la pregunta del cliente.

El paso 12 aunque en principio trivial, tiene una gran importancia ya que permite una fuerte reducción en la dimensionalidad del problema. Como veremos más adelante como parte de este trabajo utilizamos una representación vectorial de los mensajes, sin esta reducción el vocabulario del corpus está formado por 9572 palabras que se convertirían en 9572 componentes de estos vectores. Tras aplicar la eliminación de los términos con frecuencia menor que 5 el vocabulario se reduce a 1647 palabras.

No abordamos procesos de lematización, experimentos en fases iniciales sobre un conjunto más reducido de mensajes nos permitió comprobar que la lematización tiene un efecto muy positivo en la cobertura pero negativo en la precisión. Como veremos la utilización de lexicón es más adecuada para aumentar tanto la precisión como la cobertura en nuestro problema.

4.3 Utilización de lexicón

Previamente nos referíamos a la identificación y etiquetado tanto de información de dominio como genérica. Dentro de la disciplina del Procesamiento del Lenguaje Natural a este proceso de etiquetado se le conoce como identificación de entidades (NER – Named Entities Recognition). El proceso consiste en localizar palabras o grupos de palabras que representan nombres de personas, organizaciones, ubicaciones, fechas y horas, cantidades, etc. y etiquetar esa agrupación con una de las categorías mencionadas. NER es un campo de estudio donde se han realizado importantes progresos siendo el estado del arte actual para el idioma inglés próximo a la capacidad de los humanos en la misma tarea. Los enfoques en la literatura para abordar este problema son principalmente tres: basados en gramáticas, modelos estadísticos y aprendizaje automático. Los primeros tienen un coste muy alto y necesitan de expertos en lingüística computacional, el segundo necesita grandes cantidades de datos anotados y en general todos sufren de problemas de aplicación a través de diferentes dominios.

Existen diversos componentes y librerías software de código abierto públicas para abordar esta labor. Una de las más conocida es OpenNLP (basado en modelos de máxima entropía) que proporciona modelos para diversos lenguajes, entre ellos el español.

Sin embargo, la aplicación de los modelos NER para español de OpenNLP en nuestro dominio presentan malos resultados. Este deficiente comportamiento nos llevó a plantear la eliminación de estos filtros e integrar la identificación como parte de

nuestro lexicón. Este lexicón se compone de un archivo de texto que recoge un conjunto de expresiones regulares y su correspondiente etiqueta para reemplazo.

El etiquetado que recoge el lexicón abarca los siguientes elementos:

Entidad	Label	Descripción
Museos	<Museo>	Nombres de museos de Patrimonio Nacional. Una única etiqueta para todos los posibles museos. Ejemplo: "El Escorial"
Albarán	<Albaran>.	Identificación de un código de albarán. Una única etiqueta para todos los códigos.
Localizador de compra	<Localizador>	Identificación del localizador de una compra. Una única etiqueta para todos los códigos.
Número de integrantes de grupo	<NumeroIntegrantes>.	Modela expresiones del tipo 20 personas, 15 entradas, 25 billetes, 12 estudiantes, etc. Una única categoría.
Tarifas	<Tarifa>	Nombres de precios regulares de entradas. Una única categoría.
Bono	<Bono>	Diferentes referencias al concepto de bono. Una única categoría.
Importes	<Importe>	Cantidades económicas. Una única categoría
Gratuidad	<Gratuidad>	Diferentes referencias al concepto de gratuidad. Una única categoría.
Discapacitados	<Discapacitados>.	Diferentes referencias al colectivo de discapacitados.
Desempleados	<Desempleados>	Diferentes referencias al colectivo de desempleados.
Jubilados	<Jubilados>.	Diferentes referencias al colectivo de jubilados.
Entradas	<Entradas>	Referencias a número de entradas. Una única categoría.
Curso escolar	<CursoEscolar>	Referencias a cursos escolares. Una única categoría.
Online	<Online>	Distintas representaciones del concepto online. Una única categoría.
NIF	<NIF>	Número de identificación fiscal nacional. Una única categoría
Edad	<Edad>	Edad válida. Una única categoría.
Emails	<Email>	Direcciones de email válidas. Una única categoría.
Urls	<URL>	URLs válidas. Una única categoría.
Teléfonos	<Telefono>	Teléfonos válidos. Una única categoría.
Códigos postales	<CP>	Códigos postales válidos. Una única categoría.
Fecha	<Fecha>	Todos los formatos de fecha corta y larga utilizados por los clientes (cantidad de formatos mucho mayor de lo que a priori se podría pensar). Una única categoría.
Hora	<Hora>	Todos los formatos de hora corta y larga utilizados por los cliente (de nuevo cantidad de formatos muy dispar y sin seguir estándares). Una única categoría.
Números	<Numero>	Números enteros. Una única categoría.

Es importante mencionar que las expresiones regulares utilizadas presentan numerosas variantes para la mayor parte de las identificaciones: uso o no de artículos, alias, número de espacios en blanco variables entre palabras, etc. Esto puede provocar que los tiempos de casado de los patrones se disparen, siendo necesaria la utilización de timeouts para que esta fase del preprocesado de mensajes no sea un lastre en coste temporal respecto al total del proceso. La consecuencia es que en situaciones puntuales no existe garantía de que se llegue a realizar el etiquetado de algunos elementos del léxico.

Mostramos el resultado de aplicar todas estas transformaciones a alguno de los mensajes de nuestro corpus:

Mensaje 1
Original
Reservas grupos escolares Buenos días, Estoy haciendo reservas para visitas del palacio Real de Madrid para grupos escolares a traves de su web. He recibido bien los albaranes de confirmacion de entradas (gratuitas) Mi pregunta es : la escuela debe llevar una carta de presentacion de la escuela el dia de la visita o hay que mandarla antes por email? Gracias por su ayuda, Nombre_cliente
Normalizado
reservas grupos escolares estoy haciendo reservas visitas del tfmmuseo grupos escolares traves web recibido bien albaranes confirmacion tfmentrada tfmgratuidad pregunta escuela debe llevar carta presentacion escuela dia visita hay mandarla antes email

Mensaje 2
Original
Incidencia o reclamación ... <body style="font-family:Arial;font-size:14px;color:#58585a;width:800px;"> <div style="border:solid 1px #58585a;padding:40px;"> Tipo de incidencia: Incidencia o reclamación Datos usuario: 68522 – email_cliente Nombre: nombre_cliente Teléfono: teléfono_cliente Email: email_cliente Recinto: Palacio Real de Madrid Localizador:005115368J Mensaje: He realizado la compra de dos entradas para el día 15 de Octubre al Palacio Real y resulta al momento de eligirlas ha habido un error y mis entradas son para el 14 de OCTubre. Me es imposible ir el 14 porque no estoy en el país siquiera. He llamado en el mismo momento de realizar la compra y me dicen que no se puede realizar una devolución ni cambio. Le ruego que me faciliten el cambio. Un saludo Nombre_cliente Email_cliente ...
Normalizado
incidencia reclamacion realizado compra dos tfmentrada dia tfmfecha tfmmuseo resulta momento eligirlas habido error mis tfmentrada son tfmfecha imposible tfmnumero porque estoy pais siquiera llamado mismo momento realizar compra dicen puede realizar devolucion cambio ruego faciliten cambio

Mensaje 3

Original

Re: Compra de entradas Web Patrimonio Nacional

Buenos tardes, acabo de comprar unas entradas on line para la casa del labrador para el lunes 7 a las 10.00 h. Me gustaría saber si habría posibilidad de cambiarlas para el martes 8 a la misma hora, pues me he equivocado de día.

Gracias!

El 06/12/2015 13:11, email_cliente escribió:

El 06/12/2015 13:09, <noreply@entradaspatrimonio.es> escribió:

Nombre_cliente,

Su compra de entradas realizada el día 06/12/2015 13:08:00 se ha procesado correctamente

El localizador de su compra es: 002703792

Le adjuntamos sus entradas en formato pdf, recordando que debe imprimir las entradas y mostrarlas el días de la visita.

Condiciones generales de visita:

- La organización se reserva el derecho de cancelar las visitas sin previo aviso por causas imprevistas de fuerza mayor. En estos casos Patrimonio Nacional devolverá el precio de la entrada o se les cambiará la visita por otro día.
- No está permitido el acceso con mochilas, bolsas u objetos grandes, paraguas, comidas y bebidas en los recorridos museísticos. En los jardines no está permitido el acceso con animales, bicicletas, comidas y bebidas.
- No se permite en el interior de los Museos el uso de cámaras fotográficas, ni de vídeo ni las de los teléfonos móviles. Sí está permitido en espacios abiertos como plazas, lonjas, patios, etc., siempre que no se contemplen con fines comerciales o publicitarios.

Saludos

Patrimonio Nacional

www.patrimonionacional.es

Venta y reserva: 902 044 454

Soporte técnico: 902 044 414

correo@entradaspatrimonio.es

Normalizado

compra tfmentrada web patrimonio nacional buenos tardes acabo comprar tfmentrada tfmonline tfmmuseo tfmfecha tfmnumero tfmfecha gustaria saber habria posibilidad cambiarlas tfmnumero misma hora pues equivocado dia

4.4 Métricas del corpus

En este apartado mostramos algunas métricas significativas para la descripción del corpus. Las métricas han sido extraídas del dataset completo sin separar entrenamiento y test. Presentamos tanto datos globales como por clase de forma independiente.

Las métricas que utilizamos son:

- Longitud promedio y desviación para todos los mensajes del corpus en número de palabras.
- Longitud promedio y desviación de los mensajes positivos y negativos para cada clase en número de palabras.
- Distancia Jaccard promedio y desviación entre todos los mensajes del corpus.
- Distancia Jaccard promedio y desviación entre todos los mensajes positivos y todos los mensajes negativos para cada clase (interclase).
- Distancia Jaccard promedio y desviación entre todos los mensajes positivos para cada clase (intraclase).
- Gráficas de la distribución de frecuencias sobre distancia Jaccard intraclase e interclase para cada clase y sobre todo el corpus.

Todo el corpus	
Longitud mensajes	35,87±24,35
Distancia Jaccard	0,068±0,044

Tabla 6 - Longitud promedio y distancia Jaccard promedio entre todos los mensajes del corpus

Por clase				
	Longitud (+)	Longitud (-)	Jaccard interclases	Jaccard intraclases
Clase 3	35,15±20,20	36,24±26,20	0,069±0,043	0,093±0,049
Clase 33	47,26±29,31	34,52±23,32	0,075±0,043	0,097±0,044
Clase 25	38,44±27,03	35,59±24,02	0,060±0,037	0,087±0,052
Clase 15	30,73±16,20	36,43±25,01	0,065±0,041	0,092±0,050
Clase 18	35,35±23,57	35,91±24,40	0,041±0,032	0,087±0,065

Tabla 7 - Para cada clase longitud promedio de mensajes positivos y negativos y distancia Jaccard entre todos los mensajes de la clase positiva y negativa (interclase) y distancia dentro de la clase positiva (intraclase).

De la información de la Tabla 6 y Tabla 7 podemos extraer una serie de rasgos interesantes. La distancia entre mensajes interclase es consistentemente menor que la distancia intraclase. Para todas las clases analizadas hay una mayor cohesión de mensaje dentro de la clase positiva que entre la positiva y la negativa. Las gráficas de distribución de frecuencias muestran visualmente esta mayor cohesión, donde acumulación de probabilidad más a la derecha muestra mayor similitud (línea roja).

	Interclase	Intraclase	Diferencia
Clase 3	0,069±0,043	0,093±0,049	+0,014
Clase 33	0,075±0,043	0,097±0,044	+0,022
Clase 25	0,060±0,037	0,087±0,052	+0,027

Clase 15	0,065± 0,041	0,092± 0,050	+0,027
Clase 18	0,041± 0,032	0,087± 0,065	+0,045

Tabla 8- Diferencias en distancia Jaccard interclase y intraclase para todas las clases del corpus

La Tabla 8 cuantifica esta diferencia de distancia. Como se puede ver los datos son bastante homogéneos salvo dos excepciones. La clase 3 muestra la menor distancia entre mensajes positivos y negativos y la clase 18 muestra la mayor distancia entre ambos. Veremos posteriormente como esta información es consistente con los resultados de clasificación que obtenemos.

En lo que respecta a la longitud de los mensajes podemos ver que la longitud promedio está en 35 palabras con una desviación alta de 24 palabras. Por clase los datos son bastantes similares exceptuando la clase 33 donde la longitud media asciende a 47 palabras y la clase 15 que muestra los mensajes más cortos y con menor variabilidad. El resto de datos son bastante similares a las métricas globales del corpus.

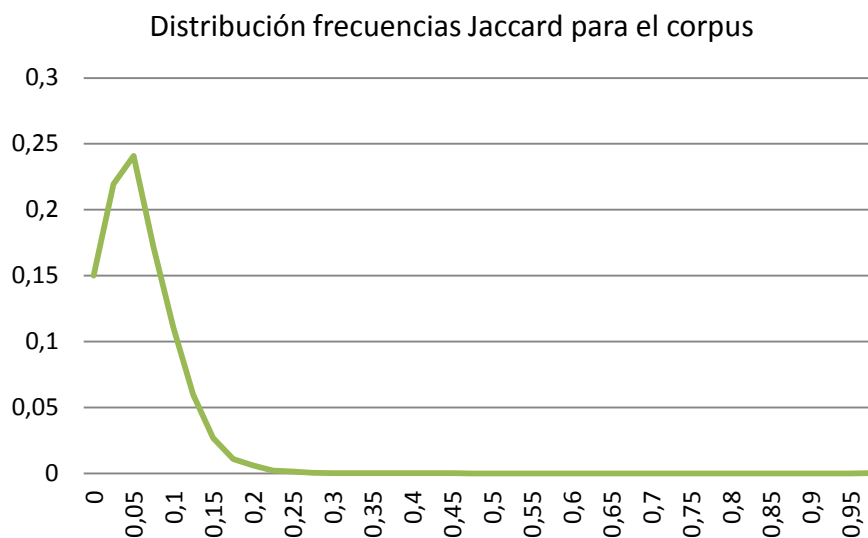


Ilustración 3 - Distribución frecuencias Jaccard para todo el corpus

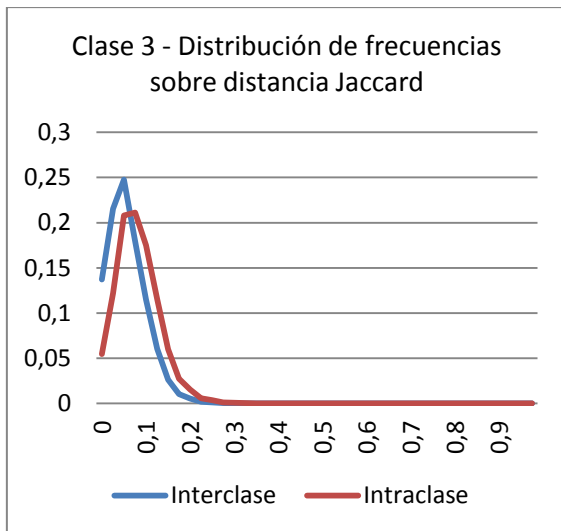


Ilustración 4 - Clase 3. Distribución de frecuencias sobre distancia Jaccard

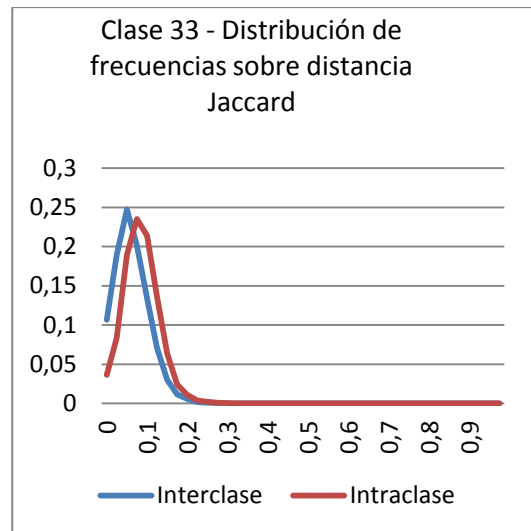


Ilustración 5 - Clase 32. Distribución de frecuencias sobre distancia Jaccard

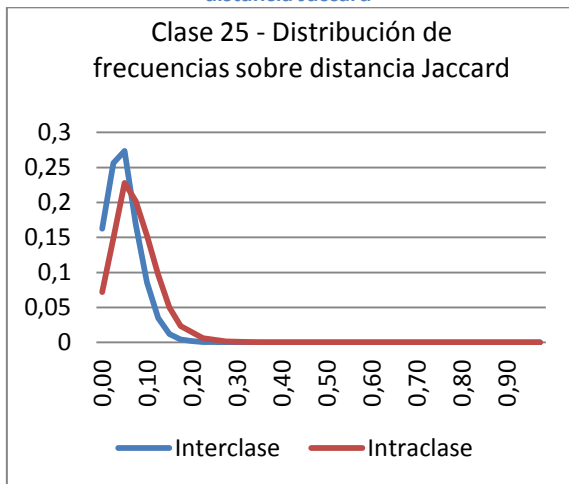


Ilustración 6 - Clase 25. Distribución de frecuencias sobre distancia Jaccard

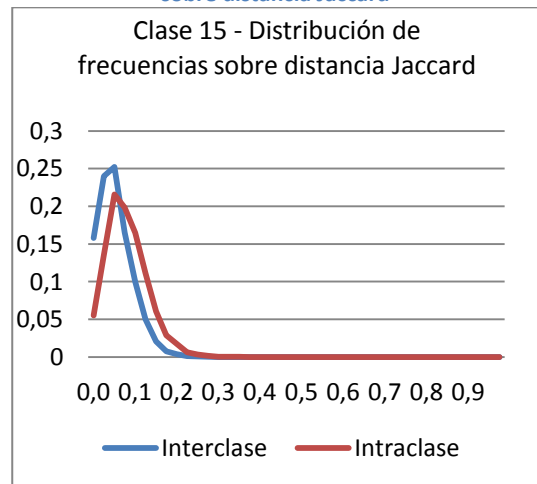


Ilustración 7 - Clase 15. Distribución de frecuencias sobre distancia Jaccard

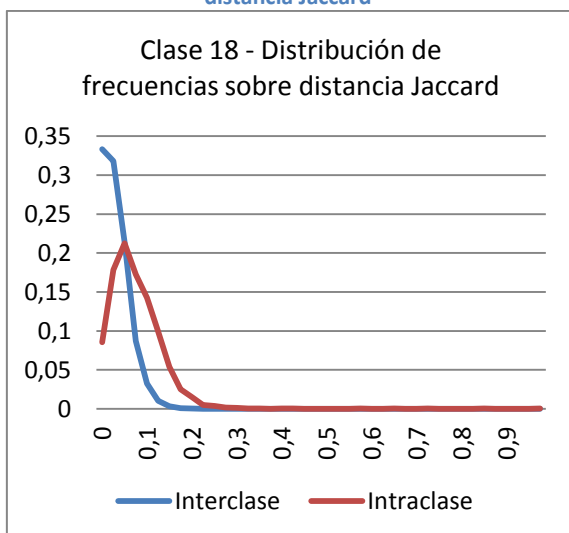


Ilustración 8 - Clase 18. Distribución de frecuencias sobre distancia Jaccard

El análisis de distancias Jaccard sobre los mensajes también nos ha permitido detectar un pequeño porcentaje de mensajes duplicados que podrían afectar a la precisión final en test de los clasificadores.

	Total muestras	Total duplicados	Porcentaje
Clase 3	1117	29	2,60%
Clase 33	328	8	2,44%
Clase 25	352	4	1,44%
Clase 15	325	8	2,46%
Clase 18	234	12	5,13%

Tabla 9 - Número de mensajes duplicados por clase

Los motivos por los que se producen estos duplicados son varios:

1. El primero de los motivos corresponde a mensajes que no son duplicados en origen, sino que después de aplicar todas las transformaciones, especialmente el lexicón, acaba como un mensaje idéntico a otro. En numerosas ocasiones los clientes mandan mensajes con la misma estructura y pequeños cambios, como si usasen una plantilla o simplemente porque copian y pegan otro mensaje. Esto ocurre especialmente con la clase 18 de solicitud de facturación.
2. Otros duplicados se generan como consecuencia del proceso de eliminación de despedidas, firmas y citados. Hemos detectado que algunos clientes comienzan sus mensajes con palabras que este proceso puede identificar como despedidas, por ejemplo "Cordial saludo". El resultado son mensajes con muy pocas palabras que pueden duplicarse. Esta situación tiene una incidencia muy baja.
3. Quizás el caso más habitual que hemos detectado son clientes que envían sus mensajes por duplicado. No sabemos el motivo de este comportamiento, quizás usuarios con poca experiencia en el uso de las herramientas, falta de confianza en la entrega a destino, errores puntuales en las interfaces de usuario, etc.

Aunque para el proceso de generación del conjunto de entrenamiento se ha seguido un procedimiento aleatorio a partir de todas las muestras disponibles (pueden existir duplicados en entrenamiento), para el conjunto de test se han eliminado explícitamente todos los duplicados existentes. De esta forma el dataset base a partir del cual se binarizan los conjuntos de test para cada clase no puede sesgar los valores de precisión obtenidos.

5 Construcción de clasificadores

Como quedó patente en el estudio del estado del arte las técnicas basadas en Naive Bayes y las máquinas de vectores soporte son las dos técnicas más extendidas en la clasificación de email, quedando las técnicas basadas en memoria en un lugar secundario. En este trabajo vamos a abordar la construcción de clasificadores que determinen si enviar o no un email de forma automática abordando tres técnicas: MNB, SVM y KNN como técnica basada en memoria.

NB en su variante multinomial (MNB) sobre bolsas de palabras es sin duda la técnica más utilizada en la clasificación de emails, comportándose como un clasificador lineal. No obstante, NB está considerado como un algoritmo con high-bias lo que provoca que tengamos que asumir que el error que va a generar cuando el número de muestras crezca va a ser mayor que el de otras técnicas (SVM en nuestro caso). Por otro lado, ese peor comportamiento asintótico confiere a MNB la capacidad de reducir la tendencia al sobreajuste de sus modelos. Múltiples trabajos han mostrado los problemas sistémicos de MNB [20] [31], sin embargo, con algunos “ajustes” sigue teniendo unos resultados y precisiones excelentes en muchos problemas, especialmente cuando el número de muestras disponible para el entrenamiento es pequeño. Si además de estos criterios tenemos en cuenta el muy bajo coste computacional, MNB es una gran opción. Bajo estas premisas descartar MNB en nuestro estudio sería poco sensato, sin embargo utilizar la técnica básica tampoco lo sería mucho más. Los trabajos en este campo han mostrado que existen algunas transformaciones básicas que mejoran más que notablemente la capacidad de clasificación de MNB: la normalización de longitud de los mensajes y las transformaciones TF-IDF. Aunque no es una transformación en sí misma, contar con clases balanceadas reduce el sesgo en el cálculo de la probabilidad final. Otras transformaciones como la complementación para compensar balanceo, ajuste en la distribución de probabilidad asumida en el texto, normalización de vectores de clases han mostrado que tiene menos impacto en la capacidad de clasificación [31].

Nuestra segunda técnica, las máquinas de vectores soporte, han mostrado en múltiples trabajos su adecuación y sobresalientes resultados en múltiples tareas relativas a la clasificación de textos. El trabajo de Joachims [26] muestra claramente las ventajas de la utilización de esta técnica en este tipo de problemas:

Primero las propias características de la clasificación de textos encajan bien en la naturaleza de la técnica:

- Espacio de características con alta dimensionalidad.
- Pocas características irrelevantes.
- Vectores de instancias dispersos.
- Muchos problemas son linealmente separables.

Segundo la propia naturaleza de la técnica proporciona ventajas sustanciales en este tipo de tareas:

- Se elimina la necesidad de un proceso de selección de características previo.
- La fuerte base matemática de la técnica la hace muy robusta en todo tipo de problemas.
- Bajo sobreajuste.
- Es factible aplicarlas con pocos o ningún ajuste de parámetros.

SVM en su forma más básica es un clasificador lineal. Cuando los datos no son linealmente separables se recurre al conocido como “truco del kernel”. Un kernel es una función que mapea los datos de su dimensión de origen a un espacio de mayor dimensionalidad intentando buscar la separación lineal en ese nuevo espacio. Lo interesante es que aunque el algoritmo opere linealmente en el espacio transformado equivalentemente está operando no linealmente en el espacio origen pudiendo así encontrar los planos de separación buscados. En tareas de NLP y clasificación de textos, tanto los kernels lineales como polinomiales de grado 2 son las opciones más usadas.

Tanto MNB como SVM son técnicas que a partir de un conjunto de datos de entrenamiento construye un modelo que posteriormente es usado para clasificar muestras no vistas previamente. Nuestra última técnica en estudio, K vecinos más próximos (KNN – K Nearest Neighbourhood), no construye un modelo sino que compara cada instancia a clasificar con todas las muestras disponibles. A continuación toma una decisión sobre la clase final en función de la clase más frecuente entre el número K (predefinido) de muestras más similares. Además del parámetro K el otro elemento clave de esta técnica es la medida de similitud. Existen múltiples métricas que pueden ser aplicadas a esta técnica, algunas de ellas como distancia euclídea o city-block opera en representaciones vectoriales otras como Jaccard u overlap pueden operar directamente sobre representaciones textuales.

Abordadas de forma general las características de estas tres técnicas, necesitamos seleccionar la configuración más adecuada para la construcción de un clasificador binario para cada clase. Así planteamos los siguientes experimentos:

- Multinomial Naive Bayes sobre una representación vectorial de las muestras de entrenamiento con normalización de longitud y transformación TF-IDF (variables continuas).
- SVM con un kernel lineal y la misma representación vectorial utilizada para MNB.
- KNN con K=1 y Jaccard como métrica de distancia.

La configuración elegida para MNB no necesita mucho más explicación. La decisión de utilizar un kernel lineal es su bajo coste computacional en relación a otro tipo de kernels y que para el problema que abordamos podría a priori ser suficiente (si el problema es linealmente separable la utilización de un kernel más complejo puede no suponer ninguna ventaja). La decisión de usar Jaccard sobre una representación textual en vez de vectorial era buscar una alternativa muy diferente a los enfoques basados en kernel, que en caso de ser viable presenta también numerosas ventajas.

Para la realización de los experimentos de MNB y SVM usaremos la herramienta WEKA mientras que para los experimentos con KNN usaremos una implementación basada en la librería Accord.Net. El clasificador usado en WEKA para MNB es la clase NaiveBayesMultinomial disponible en el namespace weka.classifiers.bayes y para SVM, SMO en el namespace weka.classifiers.functions. Para ambos clasificadores usamos la configuración por defecto sin el ajuste de ningún parámetro. En WEKA MNB no tiene ningún parámetro significativo a mencionar mientras que para SMO utilizamos la siguiente configuración:

Parámetros WEKA SVM (SMO)	
C	1.0
Epsilon	1.0E-12
Normalización de datos	Activada
Kernel	Polinomial grado 1 (lineal)
Tolerancia	0.001
weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"	

Tabla 10 - Parámetros de configuración para SVM en WEKA

Llegados a este punto es lógico que el lector se cuestione por qué se usa una de las mejores configuraciones posibles para MNB (normalización de longitud y transformación TF-IDF) y sin embargo para SVM se opta por una opción básica sin ningún ajuste. Reiteradamente toda la literatura muestra la superioridad de SVM sobre NB, nuestra intención aquí es tener una idea aproximada de como de cierta es esa "afirmación" para nuestro problema. Esto no descarta que exploremos tanto un ajuste de la configuración de SVM como otros kernels que pudiesen conseguir mejor predicción.

Otra "afirmación" recurrente en el campo de Machine Learning es que NB se comporta mejor cuando se dispone de pocos datos, mientras que SVM tiene mejor comportamiento cuando se cuenta con una cantidad razonable o grande de datos. Independientemente que el concepto pocos/muchos es bastante etéreo y con bastante seguridad va a depender del problema, sabemos que para nuestro problema (email) la cantidad de información no va a dejar de crecer. Este crecimiento constante es muy importante a la hora de tomar decisiones y evaluar resultados ya que el estudio

se está realizando sobre una foto fija y debería incluir también las necesidades a largo plazo.

5.1 Representación de las instancias

Tras extraer los mensajes de OTRS y preprocesarlos cada instancia cuenta únicamente con dos variables, el texto de la pregunta del mensaje y la clase a la que pertenece. Para MNB y SVM necesitaremos extraer una representación vectorial sobre variable continua mientras que para KNN utilizaremos una representación de Bernuilli donde cada componente del vector indica si la palabra está presente en el texto o no. La representación de Bernuilli para KNN es necesaria porque la librería que utilizamos no soporta distancia Jaccard sobre representaciones textuales.

En el primer caso (MNB/SVM) para la vectorización utilizamos el filtro StringToWordVector de WEKA con las transformaciones TF y IDF activas y normalización de longitud de documento para todos los datos. Con el preprocesado realizado para la tokenización es suficiente con separar las palabras por el carácter espacio en blanco. La aplicación de este filtro genera 1646 características correspondientes a cada una de las entradas del vocabulario con frecuencias mayores de 5.

El dataset resultante contiene todo nuestro corpus, en el siguiente paso es necesario separa los conjuntos de test y entrenamiento. La separación se realiza utilizando otro filtro de WEKA StratifiedRemoveFolds. Primero lo aplicamos para separar el 20% de las instancias y generar un conjunto de test, para luego invirtiendo la selección quedarnos con el 80% de los datos para entrenamiento. El último paso consisten en preparar un conjunto de entrenamiento y otro de test binarizado para cada una de las clases bajo estudio. Para ello, partiendo de los datasets vectorizados aplicamos el filtro RenameNominalValues transformando el código de numérico de la clase objetivo al valor "S" y el del resto de clases al valor "N". Este proceso se repite por cada clase para el conjunto de entrenamiento y test. Los archivos resultantes son editados manualmente para eliminar los anteriores valores numéricos de las clases ya que no siguen siendo necesarios y entorpecen la lectura de resultados.

Estas son las características de los datasets de entrenamiento y test obtenidos:

Datasets de entrenamiento				
Clase	Total	Positivos	Negativos	Ratio balanceo
Clase 3	2647	901	1746	1:1.94
Clase 33	2647	277	2370	1:8.56
Clase 25	2647	263	2384	1:9.06
Clase 15	2647	262	2385	1:9.10
Clase 18	2647	188	2549	1:13.55

Tabla 11- Número de instancias positivas y negativas de entrenamiento por clase y ratio de balanceo.

Datasets de test			
Clase	Total	Positivos	Negativos
Clase 3	638	210	428
Clase 33	638	72	566
Clase 25	638	61	577
Clase 15	638	61	577
Clase 18	638	44	594

Tabla 12 - Número de instancias positivas y negativas de test por clase

5.2 Balanceo de datos

Como se puede apreciar en la Tabla 11 los conjuntos de entrenamiento presentan una fuerte falta de balanceo en los datos. Excepto para la clase mayoritaria que es de 1:2 el resto de clases se mueven desde el 1:8.5 de la clase 33 a 1:23.5 de la clase 11. Recordamos que las clases incluidas en el estudio solo cubren el 71,20% de las respuestas emitidas, si nos adentráramos a tener una cobertura del 90% tendríamos que enfrentarnos a ratios por encima de 1:60.

El problema de la falta de balanceo ya fue previsto durante nuestro planteamiento inicial al usar n clasificadores binarios, y sabíamos que es una problemática que deberíamos abordar. Bajo nuestro criterio es más importante tener la mayor cantidad de información posible para la construcción de los clasificadores y paliar el problema del balanceo, que afrontar el estudio con conjuntos mejor balanceados pero con una cantidad de información menor. Obviamente esto no pretende ser una afirmación ya que cualquier de los dos enfoques podría ser válido dependiendo del conjunto de datos a tratar.

Dentro de la literatura relacionada con el tratamiento de datos no balanceados se han seguido dos enfoques bien diferenciados para tratar el problema:

- Ponderación de las muestras. Otorgando mayor peso a las clases minoritarias.
- Resampling de las muestras. El resampling puede incluir tanto oversampling como undersampling o ambos.

Dentro de la categoría de resampling nos parece especialmente interesante el algoritmo SMOTE [29] que combina la generación de nuevas muestras sintéticas para la clase minoritaria junto con undersampling de la mayoritaria. El oversampling de SMOTE opera del siguiente modo: para cada muestra de entrenamiento se localizan las n muestras más cercanas y mediante el cálculo de un desplazamiento aleatorio se interpola una muestra ubicada entre ambas.

Para nuestro problema como primera aproximación hemos optado sólo por aplicar oversampling manteniendo intacta la clase mayoritaria para los conjuntos de entrenamiento de MNB y SVM. Excepto para la clase 3 donde hemos aplicado

oversampling de 1.25, 1.50, 1.75 y 2 veces el número de muestras, para el resto de clases incrementamos mediante muestras sintéticas en 2, 4, 6 y 8 veces el número inicial.

WEKA cuenta con una implementación de SMOTE para variable continua que se puede descargar a través del Package Manager de la herramienta. Aplicamos este filtro para generar cada uno de los nuevos conjuntos de entrenamiento.

Es importante recalcar que el proceso de oversampling se aplica solo a MNB y SVM. Nuestro enfoque basado en memoria de KNN con distancia Jaccard, donde buscamos únicamente la muestra más similar, no se ve afectado por el problema del balanceo y la aplicación de estas técnicas no tiene sentido.

6 Resultados clasificación

En este apartado presentamos los resultados obtenidos en los experimentos para cada uno de los clasificadores binarios construidos. Los resultados para MNB y SVM son presentados conjuntamente y de forma aislada KNN. Para KNN no utilizamos oversampling y por tanto contamos sólo con los resultados para un único clasificador.

Los resultados se agrupan para cada clase por técnica y cantidad de oversampling utilizado. Las métricas mostradas son:

- ACC. Exactitud global del clasificador. Este valor es meramente orientativo ya que para nuestro problema tiene poca representatividad.
- ROC. Área bajo la curva ROC para la clase positiva. Para clases no balanceadas esta medida es preferible a las métricas de error estándar.
- PREC. Precisión de la clase positiva. $TP / (TP + FP)$.
- Recall. Cobertura de la clase positiva. $TP / (TP + FN)$.

Todos los resultados presentados para todos los clasificadores fueron obtenidos mediante cross-validation con 10 folds.

6.1 Clase 3. Resultados clasificación

La clase 3 se corresponde a la clase mayoritaria en el dataset con una representatividad 33.76%. Esta clase responde a los usuarios con información relativa a dónde y cómo adquirir entradas o reservar en venta anticipada.

La Tabla 13 muestra los resultados obtenidos en entrenamiento para MNB y SVM para las distintas configuraciones de oversampling (10-cross fold). A priori, esta tabla puede parecer poco significativa (exceptuando la primera fila), ya que obviamente se va a producir un incremento en la precisión y recall con la inclusión de muestras sintéticas para la clase minoritaria. Muchas de estas nuevas muestras caerán dentro de las superficies ya delimitadas por los clasificadores hinchando artificialmente las métricas.

CLASE 3	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	78.617%	0,872	0,772	0,658	76.691%	0,742	0,666	0,655
Smote x1.25	80.362%	0,890	0,823	0,717	80.049%	0,793	0,761	0,738
Smote x1.50	82.370%	0,903	0,864	0,763	82.661%	0,828	0,842	0,779
Smote x1.75	82.842%	0,909	0,864	0,793	83.865%	0,840	0,873	0,804
Smote x2	84.019%	0,915	0,881	0,818	85.203%	0,851	0,897	0,827

Tabla 13 – Entrenamiento clase 3. Resultados para los experimentos con MNB y SVM sobre las distintas configuraciones de oversampling

La Tabla 14 muestra la evaluación de nuevo de todos estos clasificadores sobre el conjunto de test (no visto en entrenamiento).

CLASE 3	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	78.213%	0,866	0,824	0,629	76.019%	0,823	0,643	0,634
Smote x1.25	78.213%	0,868	0,819	0,630	76.019%	0,816	0,629	0,638
Smote x1.50	78.683%	0,867	0,819	0,637	77.273%	0,829	0,671	0,650
Smote x1.75	78.527%	0,867	0,824	0,634	75.912%	0,816	0,688	0,633
Smote x2	78.056%	0,867	0,810	0,630	75.862%	0,741	0,690	0,620

Tabla 14 – Test clase 3. Resultados de todos los clasificadores para MNB y SVM

Las conclusiones a la vista de estos datos son claras, el oversampling para la clase 3 realmente no aporta nada y el clasificador con los datos originales tiene la misma capacidad de predicción. No hay diferencias prácticamente en precisión entre ambas técnicas aunque MNB presenta una cobertura un 18% mayor en promedio.

Además sabemos que SVM no tiene ningún tipo de ajuste y que la comparación directa entre ambas técnicas no es equilibrada. Un ajuste grueso del parámetro C (Tabla 15) muestra que para C=0.1 SVM incrementa la precisión en un 5% y la cobertura en un 6%.

CLASE 3	Kernel Lineal			
	Acc	ROC	Recall	Prec
C=0.01	78.370%	0,858	0,662	0,675
C=0.1	79.624%	0,867	0,700	0,687
C=1.0	76.019%	0,823	0,643	0,634
C=10.0	72.884%	0,784	0,614	0,584
C=100.0	71.787%	0,777	0,581	0,570
C=1000.0	72.570%	0,771	0,595	0,581

Tabla 15 – Clase 3. Ajuste básico del parámetro C para SVM con kernel lineal. Resultados mostrados sobre datos de test.

Si exploramos un poco más la configuración de SVM y pasamos a usar un kernel polinomial de grado 2 de nuevo con una búsqueda básica para ajustar el parámetro C, nos encontramos que de nuevo incrementamos la precisión en un 4% adicional volviendo a la cobertura original del 64% (Tabla 16). Es claro que un ajuste más fino podría mejorar aún más los resultados.

CLASE 3	Kernel polinomial grado 2			
	Acc	ROC	Recall	Prec
C=0.01	78.370%	0,856	0,681	0,668
C=0.1	78.527%	0,875	0,805	0,638
C=1.0	77.900%	0,868	0,576	0,699
C=10.0	80.251%	0,877	0,638	0,728
C=100.0	80.251%	0,876	0,638	0,728
C=1000.0	80.251%	0,877	0,638	0,728

Tabla 16 - Clase 3. Estudio para un ajuste básico del parámetro C para SVM con kernel polinomial de grado 2. Resultados mostrados sobre datos de test.

Los resultados para KNN con distancia Jaccard están por detrás en precisión del resto de clasificadores.

KNN + Jaccard			
Acc	Error	Recall	Prec
74,43%	0,051	0,697	0,605

Tabla 17 - Clase 3. Resultados para KNN + Jaccard

La Ilustración 9 compara los cuatro clasificadores presentados: KNN, MNB, SVM con kernel lineal y C=0.1 y SVM con kernel polinomial de grado 2 y C= 10.0

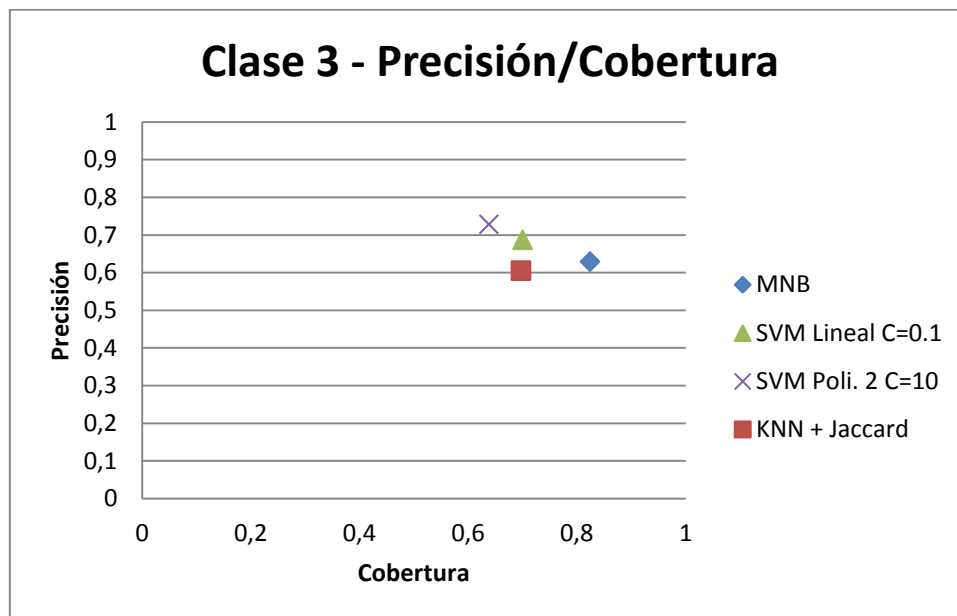


Ilustración 9 – Clase 3. La precisión de SVM con un kernel polinomial de grado 2 débilmente ajustado claramente supera en precisión al resto de configuraciones.

Para intentar elevar la precisión de los clasificadores SVM incluimos en el proceso de entrenamiento la calibración de modelos probabilísticos utilizando Logistic Regression. Esta calibración nos proporciona una probabilidad de salida para cada instancia clasificada tanto para la clase positiva como para la clase negativa. Siguiendo el mismo planteamiento utilizado por [15] asignamos una respuesta positiva cuando:

$$\frac{\text{Probabilidad positiva}}{\text{Probabilidad negativa}} > \lambda$$

Donde λ es una constante real que en la configuración base toma un valor 1.0. Con las probabilidades de salidas otorgadas por la calibración e incrementando el valor de λ podemos controlar cual es la certeza que el clasificador otorga a la clase positiva.

λ	Precisión	Cobertura
1	0,728	0,638
11	0,787	0,333
21	0,761	0,257
31	0,774	0,229
41	0,767	0,219
51	0,769	0,190
61	0,800	0,171
71	0,814	0,167
81	0,814	0,167
91	0,846	0,157
101	0,838	0,148

Tabla 18 – Clase 3. Ajuste del parámetro K para controlar la certeza en la predicción.

El ajuste de la razón entre las probabilidades de salida consigue un aumento de precisión sustancial pero a costa de una fuerte reducción de la cobertura. Para $\lambda = 20$ un incremento en precisión del 6% fuerza una reducción de la cobertura de un 30%, para $\lambda = 100$ obtenemos un 11% más de precisión a costa de una reducción de cobertura casi de un 50%.

6.2 Clase 33. Resultados clasificación

La clase 33 se corresponde a la segunda clase en el dataset con una representatividad del 10.64%. Esta clase responde a los usuarios con información relativa a reservas de visitas agotadas.

Siguiendo el mismo patrón que utilizamos para presentar la información en la clase 3, la Tabla 19 muestra los resultados de entrenamiento de los clasificadores con oversampling de la clase minoritaria en 2, 4, 6 y 8 veces el número de muestras inicial. La Tabla 20 muestra los resultados de los clasificadores generados en test.

CLASE 33	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	87.420%	0,911	0,755	0,441	90.970%	0,728	0,498	0,580
Smote x2	90.253%	0,895	0,928	0,677	92.476%	0,968	0,847	0,776
Smote x4	91.662%	0,983	0,964	0,810	94.020%	0,946	0,960	0,866
Smote x6	92.708%	0,986	0,978	0,863	94.792%	0,954	0,986	0,898
Smote x8	93.873%	0,989	0,985	0,898	95.552%	0,957	0,989	0,924

Tabla 19 – Entrenamiento clase 33. Resultados de los experimentos para MNB y SVM sobre las distintas configuraciones de oversampling

CLASE 33	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	86.207%	0,914	0,681	0,430	88.872%	0,888	0,528	0,507
Smote x2	86.520%	0,898	0,667	0,436	89.028%	0,876	0,528	0,514
Smote x4	87.304%	0,889	0,653	0,456	88.558%	0,885	0,528	0,494
Smote x6	88.558%	0,880	0,653	0,495	88.558%	0,893	0,556	0,494
Smote x8	87.304%	0,873	0,639	0,455	89.185%	0,886	0,556	0,519

Tabla 20 – Test clase 3. Resultados de los experimento para MNB y SVM sobre las distintas configuraciones de oversampling.

Para la clase 33 el oversampling de nuevo parece no aportar nada a SVM, sin embargo sí se ve una mejoría respecto a la configuración base para MNB, especialmente en la configuración Smotex6. De forma general SVM presenta una ventaja en precisión de aproximadamente un 8% sobre MNB mientras este de nuevo tiene mejor cobertura, en este caso 15%. Un ajuste grueso del parámetro C arroja un incremento de precisión del 9% y cobertura del 6% (Tabla 21). Movernos a un kernel polinomial de grado 2 con C=0.01 eleva la precisión un 12% (Tabla 22) y la cobertura un 10%. Para C=0.1 tenemos aún mejor precisión pero la caída en la cobertura es muy marcada, usamos el valor C=0.01 como referencia.

CLASE 33	Kernel lineal			
	Acc	ROC	Recall	Prec
C=0.01	90.909%	0,912	0,583	0,600
C=0.1	90.282%	0,918	0,556	0,571
C=1.0	89.028%	0,876	0,528	0,514
C=10.0	87.931%	0,862	0,528	0,469
C=100.0	86.207%	0,857	0,667	0,429
C=1000.0	81.191%	0,831	0,694	0,338

Tabla 21 – Clase 33. Ajuste básico del parámetro C para SVM con kernel lineal. Resultados mostrados sobre datos de test.

CLASE 33	Kernel polinomial grado 2			
	Acc	ROC	Recall	Prec
C=0.01	91.693%	0,909	0,639	0,630
C=0.1	91.693%	0,916	0,389	0,757
C=1.0	91.379%	0,916	0,333	0,774
C=10.0	22.727%	0,727	1,000	0,127
C=100.0	22.727%	0,727	1,000	0,127
C=1000.0	22.727%	0,727	1,000	0,127

Tabla 22 - Clase 33. Ajuste básico del parámetro C para SVM con kernel polinomial de grado 2. Resultados mostrados sobre datos de test.

Los resultados para KNN con distancia Jaccard vuelven a estar alejados de SVM:

KNN + Jaccard			
Acc	Error	Recall	Prec
88,062%	0,080	0,432	0,438

Tabla 23 - Clase 33. Resultado KNN + Jaccard

La Ilustración 10 compara las cuatro mejores configuraciones de los clasificadores en test. Para esta clase queda patente que MNB y Jaccard son malas opciones y que de nuevo usar un kernel polinomial de grado 2 eleva la precisión del clasificador.

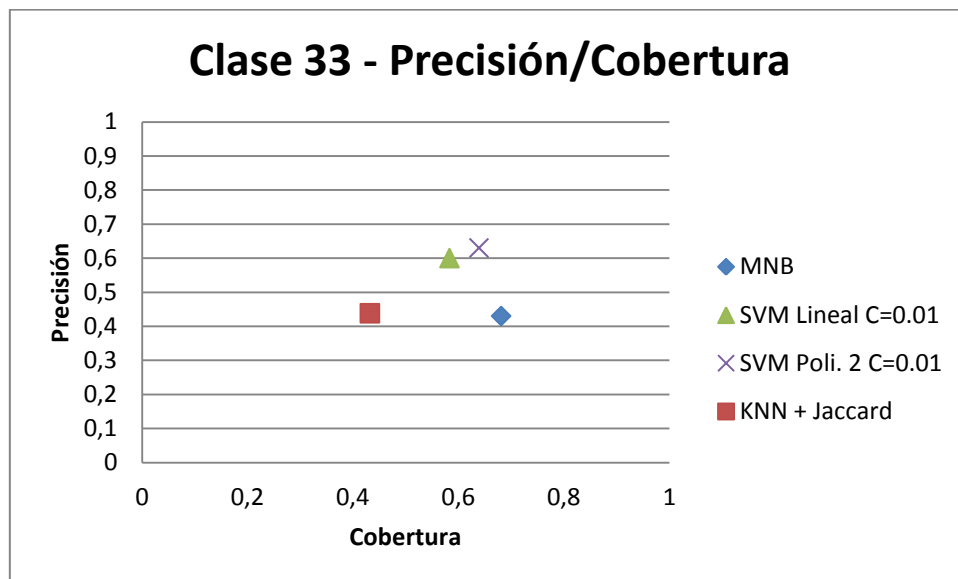


Ilustración 10 - Clase 33. La precisión de SVM con un kernel polinomial de grado 2 débilmente ajustado claramente supera en precisión al resto de configuraciones.

De nuevo, utilizamos la técnica de controlar el umbral de la razón entre las probabilidades de salida para incrementar la precisión del clasificador final.

K	Precisión	Cobertura
1	0,630	0,639
201	0,780	0,444
401	0,784	0,403
601	0,788	0,361
801	0,788	0,361
1001	0,839	0,361
1201	0,839	0,361
1401	0,839	0,361
1601	0,828	0,333
1801	0,821	0,319
2001	0,815	0,306

Tabla 24- Clase 33. Ajuste del parámetro K para controlar la certeza en la predicción

Los resultados son muy similares a los visto para la clase 3. Un aumento de la precisión, en este caso más acusado, a costa de una reducción fuerte de la cobertura.

6.3 Clase 25. Resultados clasificación

La clase 25 se corresponde a la tercera clase en el dataset con una representatividad del 9.91%. Esta clase responde a los usuarios con información relativa a la no posibilidad de cambio tras la compra.

CLASE 25	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	94.560%	0,983	0,945	0,994	95.996%	0,864	0,745	0,834
Smote x2	95.498%	0,991	0,979	0,811	97.560%	0,967	0,954	0,914
Smote x4	96.624%	0,994	0,994	0,905	98.108%	0,984	0,991	0,949
Smote x6	97.198%	0,995	0,994	0,939	98.410%	0,986	0,997	0,964
Smote x8	97.415%	0,995	0,992	0,954	98.329%	0,984	1,000	0,966

Tabla 25- Clase 25. Resultados de los experimento en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling

CLASE 25	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	95.141%	0,981	0,869	0,697	95.925%	0,954	0,754	0,807
Smote x2	95.455%	0,978	0,869	0,716	95.925%	0,954	0,689	0,857
Smote x4	95.455%	0,975	0,836	0,729	95.768%	0,956	0,639	0,886
Smote x6	95.455%	0,972	0,836	0,729	95.455%	0,958	0,689	0,808
Smote x8	95.298%	0,971	0,820	0,725	95.768%	0,955	0,705	0,827

Tabla 26 - Clase 25. Resultados de los experimento en test para MNB y SVM sobre las distintas configuraciones de oversampling

Para la clase 25 si podemos observar una mejoría en precisión para los conjuntos sobremuestreados con SMOTE para las dos técnicas, más notable en SVM. Smotex4 obtiene los mejores resultados con un incremento del 8% y una reducción de cobertura del 11% sobre la versión sin sobremuestreo. De nuevo SVM arroja mejores resultados de precisión sin ningún tipo de ajuste, con un incremento mínimo del 11% y un máximo del 16%, mientras que MNB tiene mejor cobertura con un incremento entre un 11% y un 20%.

Repetimos primero el ajuste de C para el kernel lineal de SVM, pero en este caso vamos a utilizar el conjunto de entrenamiento con SMOTE 400% dado que parece ofrecer una mejora notable. Ningún valor de C de los explorados arroja una mejor precisión sobre los datos previos con C=1.0. Sin embargo, el kernel de grado 2 sí genera una precisión un 5% mayor y un incremento de cobertura del 6% para C=0.1.

CLASE 25 Smote x4	Kernel lineal			
	Acc	ROC	Recall	Prec
C=0.01	96.395%	0,970	0,787	0,828
C=0.1	96.709%	0,972	0,754	0,885
C=1.0	95.768%	0,956	0,639	0,886
C=10.0	92.163%	0,945	0,852	0,559
C=100.0	92.163%	0,945	0,852	0,559
C=1000.0	92.163%	0,945	0,852	0,559

Tabla 27 - Clase 25. Ajuste básico del parámetro C para SVM con kernel lineal. Resultados mostrados sobre datos de test.

CLASE 25 Smote x4	Kernel Polinomial grado 2			
	Acc	ROC	Recall	Prec
C=0.01	96.395%	0,975	0,705	0,896
C=0.1	96.709%	0,975	0,705	0,935
C=1.0	96.709%	0,971	0,836	0,823
C=10.0	96.709%	0,971	0,836	0,823
C=100.0	96.709%	0,971	0,836	0,823
C=1000.0	96.709%	0,971	0,836	0,823

Los resultados para Jaccard siguiendo la tónica del resto de clases vistas, muy alejados de SVM aunque en este caso por encima en precisión de MNB.

KNN + Jaccard			
Acc	Error	Recall	Prec
94,35%	0,071	0,671	0,737

Tabla 28 - Clase 25. Resultados para KNN + Jaccard

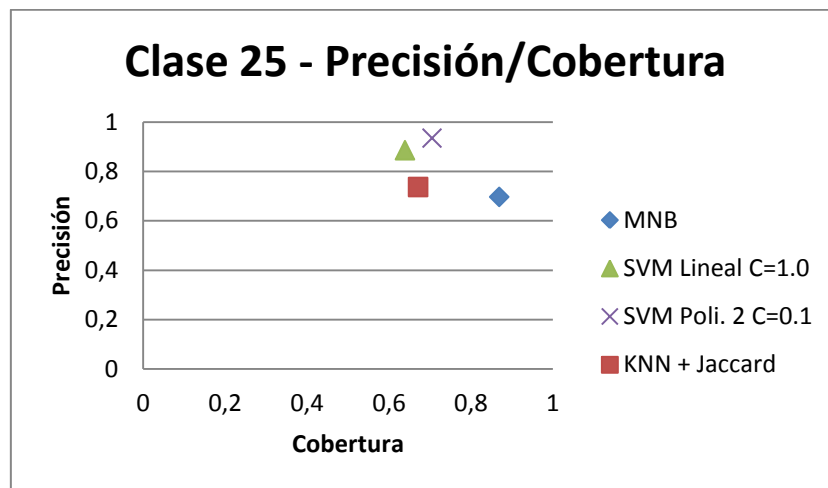


Ilustración 11 - Clase 25. SVM con kernel polinomial de grado 2 y C=0.1 supera ampliamente en precisión al resto de técnicas.

Utilizar la técnica de controlar la razón entre las probabilidades de salida para dirigir la precisión del clasificador fina en este caso tiene una utilidad prácticamente nula.

K	Precisión	Cobertura
1	0,935	0,705
101	0,943	0,541
201	0,933	0,459
301	0,926	0,410
401	0,920	0,377
501	0,920	0,377
601	0,913	0,344
701	0,909	0,328
801	0,909	0,328
901	0,909	0,328
1001	0,905	0,311

Tabla 29 - Clase 25. El ajuste del umbral para las probabilidades de salida obtenidas para SVM con kernel lineal.

6.4 Clase 15. Resultados clasificación

La clase 15 se corresponde a la cuarta clase en el dataset con una representatividad del 9.82%. Esta clase responde a los usuarios con información relativa a la adquisición de entradas gratuitas.

Los resultados en entrenamiento y test muestran que el sobremuestreo para esta clase no aporta ninguna mejora en el mejor clasificador. SVM vuelve a estar un 16% por encima en precisión sobre MNB con aproximadamente 14% menos de cobertura.

CLASE 15	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	91.915%	0,940	0,798	0,565	94.635%	0,927	0,710	0,738
Smote x2	93.778%	0,980	0,929	0,772	96.597%	0,986	0,922	0,893
Smote x4	94.669%	0,989	0,960	0,877	97.524%	0,996	0,987	0,936
Smote x6	95.198%	0,991	0,966	0,917	97.928%	0,997	0,992	0,958
Smote x8	95.983%	0,992	0,976	0,941	98.304%	0,998	0,994	0,970

Tabla 30 – Clase 15. Resultados en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling

CLASE 15	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	93.887%	0,968	0,803	0,645	95.141%	0,938	0,656	0,800
Smote x2	94.044%	0,960	0,721	0,677	94.671%	0,931	0,656	0,755
Smote x4	93.887%	0,955	0,705	0,672	94.514%	0,940	0,689	0,724
Smote x6	93.103%	0,950	0,672	0,631	94.671%	0,929	0,656	0,755
Smote x8	93.260%	0,947	0,656	0,645	95.141%	0,935	0,656	0,800

Tabla 31 - Clase 15. Resultados en test para MNB y SVM sobre las distintas configuraciones de oversampling

KNN + Jaccard			
Acc	Error	Recall	Prec
93,140%	0,075	0,634	0,654

Tabla 32 - Clase 15. Resultados KNN + Jaccard

Quizás el rasgo más característico es que los resultados en test sorprendentemente son aproximadamente un 7% mejores que en entrenamiento. Sabemos que los datos en test no han sido vistos previamente y que no presentan ningún duplicado, así que existen dos posibles explicaciones:

- El conjunto de datos separado para test es muy representativo de la clase y los clasificadores construidos identifican perfectamente muchas de sus instancias.
- WEKA calcula la validación cruzada con menos datos que el clasificador final generado lo que puede provocar variaciones en la capacidad real de clasificación cuando se aplica al conjunto de test.

Para la clase 15 realizar un ajuste grueso de C para un kernel polinomial de grado 2 no arroja ninguna ventaja sobre el ajuste del clasificador lineal para C=0.01, obteniéndose la misma precisión pero un 10% de cobertura adicional.

CLASE 15	Kernel lineal			
	Acc	ROC	Recall	Prec
C=0.01	95.925%	0,943	0,639	0,907
C=0.1	96.552%	0,952	0,754	0,868
C=1.0	95.141%	0,938	0,656	0,800
C=10.0	85.110%	0,905	0,836	0,375
C=100.0	41.379%	0,730	0,984	0,139
C=1000.0	41.379%	0,730	0,984	0,139

CLASE 15	Kernel polinomial grado 2			
	Acc	ROC	Recall	Prec
C=0.01	94.984%	0,938	0,541	0,892
C=0.1	53.762%	0,917	0,984	0,169
C=1.0	54.075%	0,916	0,967	0,169
C=10.0	53.919%	0,916	0,967	0,168
C=100.0	54.075%	0,916	0,967	0,169
C=1000.0	54.075%	0,916	0,967	0,169

La Ilustración 12 muestra la capacidad de clasificación de las tres técnicas para esta clase.

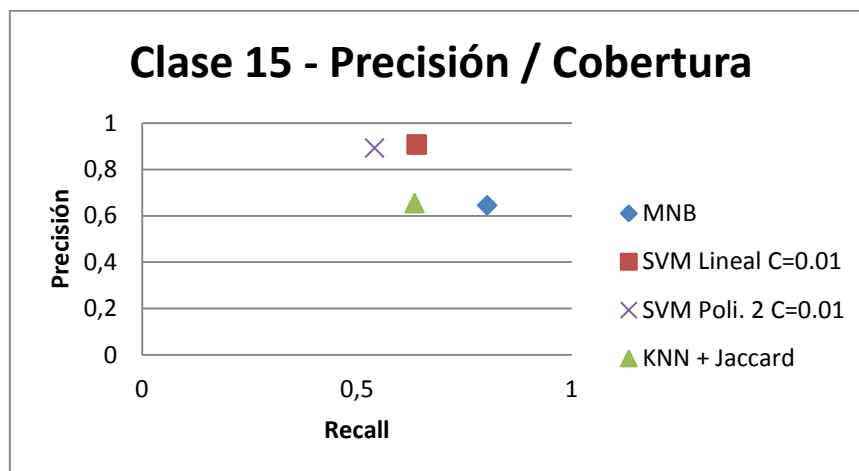


Ilustración 12 - Clase 15. SVM con kernel lineal y C=0.01 supera ampliamente en precisión a MNB y Jaccard.

El ajuste de umbral de la razón entre las probabilidades de salida de las clases no muestra ninguna mejora en el control de la precisión.

K	Precisión	Cobertura
1	0,907	0,639
101	0,880	0,361
201	0,905	0,311
301	0,900	0,295
401	0,889	0,262
501	0,882	0,246
601	0,875	0,230
701	0,875	0,230
801	0,867	0,213
901	0,867	0,213
1001	0,867	0,213

6.5 Clase 18. Resultados y análisis

La clase 18 se corresponde a la quinta clase en el dataset con una representatividad del 7,07%. Esta clase responde a los usuarios con información relativa a facturación de compras.

En las siguientes tablas se muestran los resultados en entrenamiento y test.

CLASE 18	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	97.922%	0,997	1,000	0,774	99.471%	0,978	0,957	0,968
Smote x2	98.554%	0,999	0,997	0,904	99.612%	0,993	0,989	0,982
Smote x4	99.097%	0,999	1,000	0,963	99.7509 %	0,998	0,999	0,991
Smote x6	99.331%	0,999	1,000	0,979	99.777 %	0,998	1,000	0,993
Smote x8	99.369%	0,999	1,000	0,984	99.823%	0,999	1,000	0,995

Tabla 33 - Clase 18. Resultados en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling

CLASE 18	MNB				SVM Lineal			
	Acc	ROC	Recall	Prec	Acc	ROC	Recall	Prec
No oversampling	98.433%	0,995	0,955	0,840	99.530 %	1,000	0,932	1,000
Smote x2	98.433%	0,991	0,932	0,854	99.530 %	1,000	0,932	1,000
Smote x4	98.903%	0,985	0,932	0,911	98.433%	0,999	1,000	0,815
Smote x6	98.903%	0,982	0,932	0,911	99.687 %	1,000	0,977	0,977
Smote x8	98.903%	0,980	0,932	0,911	99.687 %	1,000	0,955	1,000

Tabla 34 - Clase 18. Resultados en entrenamiento para MNB y SVM sobre las distintas configuraciones de oversampling

KNN + Jaccard			
Acc	Error	Recall	Prec
98,700%	0,0438	0,914	0,902

Tabla 35 - Clase 18. Resultados para KNN + Jaccard

Los resultados para la clase 18 son casi perfectos para SVM con un 100% de precisión y cobertura muy cercana al mismo valor. El oversampling no aporta nada a SVM, aunque sí parece mejorar ligeramente MNB en las configuraciones Smotex4 a x8. Jaccard en esta ocasión supera a MNB para las dos configuraciones con menor oversampling.

Para la clase 18 un clasificador SVM lineal con $C=1.0$ posibilita una respuesta completamente automática.

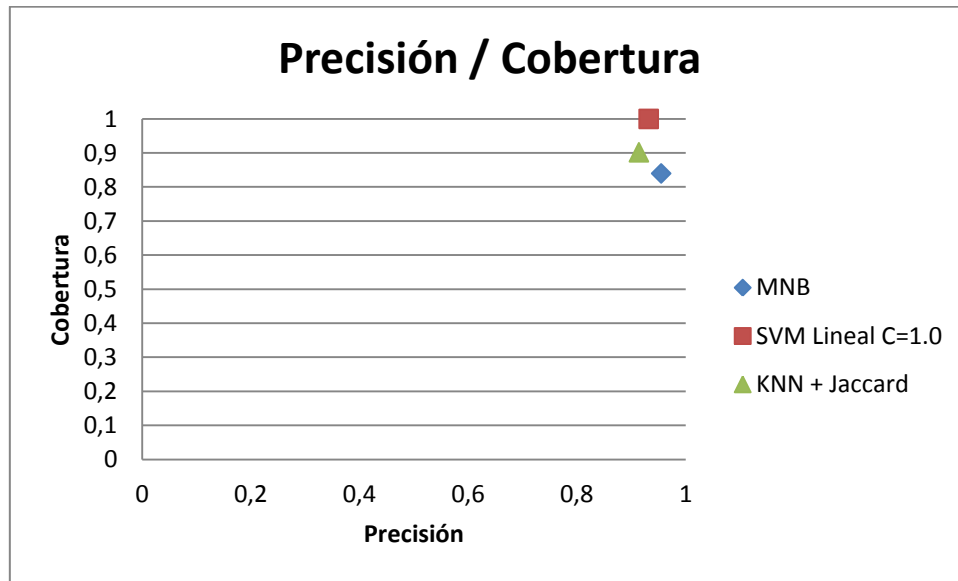


Ilustración 13 - Clase 15. Buenos resultados para todos los clasificadores. SVM con kernel lineal sin ajuste de C es la mejor técnica.

6.6 Análisis clasificadores binarios

La Tabla 37 muestra las curvas de precisión/cobertura por clase para los clasificadores seleccionados de cada técnica. El criterio de selección que hemos seguido consiste en elegir el clasificador con la precisión más alta para la clase positiva con al menos una cobertura del 50% de las muestras de esa clase. Aunque en algún caso podríamos haber elegido clasificadores con precisiones más altas, los resultados muestran decrementos muy fuertes en la cobertura para esos casos. El dataset con el que contamos no abarca el total de los mensajes recibidos en el centro de soporte en el intervalo de tiempo considerado, lo que implica que si optamos por coberturas muy bajas la fracción final de mensajes tratados podría ser excesivamente pequeña. Buscamos alta precisión pero no a toda costa. Por otro lado, hemos descartado la opción de controlar la razón entre las probabilidades de salida. Con el criterio anterior sólo para la clase 33 podría tener relevancia, para el resto la reducción de cobertura es excesiva.

Como se puede apreciar (Tabla 37), para una cobertura total de la clase positiva, el dominio de SVM sobre MNB y KNN es claro. SVM es la mejor opción para todas las clases para una cobertura igual o superior al 30%. KNN es la curva con menor área claramente dominada por el resto.

Dejando de lado la clase 18, si nos planteamos buscar alta precisión sacrificando el nivel de cobertura por debajo del 30% vemos:

- La clase 3 no tiene ninguna técnica dominante.
- MNB presenta una ligera ventaja sobre SVM para las clases 33 y 25.
- SVM es la mejor opción para la case 15.

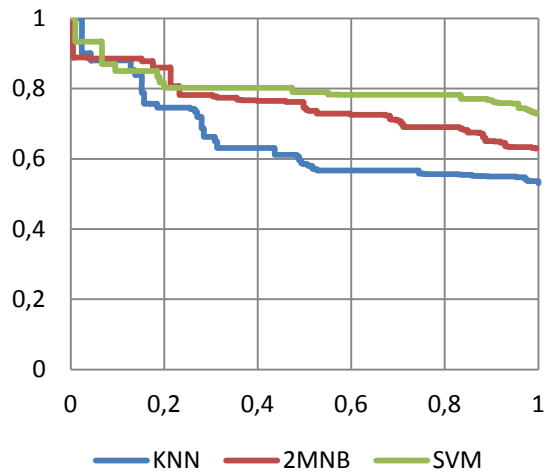
Excepto para la clase 25 donde MNB mantiene una precisión perfecta hasta el 30% de cobertura, para el resto de clases y técnicas ya en valores bajos de cobertura se produce un rápido descenso de la precisión. Para el 20% de cobertura todos los clasificadores están muy cerca o por debajo del 90% de precisión (exceptuando de nuevo la clase 18).

La Tabla 36 muestra de forma resumida los resultados obtenidos para todas las clases y técnicas. La última columna muestra la ventaja de SVM en precisión respecto de la segunda mejor técnica para una cobertura completa.

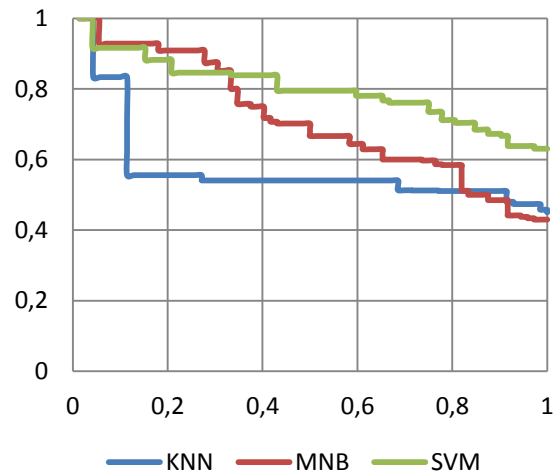
	KNN			MNB			SVM			
	Smote	Recall	Prec.	Smote	Recall	Prec.	Kernel	Smote	Recall	Prec.
Clase 3	No	0,697	0,605	No	0,824	0,629	Poly2	No	0,638	0,728 (+9.9%)
Clase 33	No	0,432	0,438	x6	0,653	0,495	Poly2	No	0,639	0,630 (+13.5%)
Clase 25	No	0,671	0,737	x4	0,836	0,729	Poly2	x4	0,705	0,935 (+20.2%)
Clase 15	No	0,634	0,654	x2	0,721	0,677	Lineal	No	0,639	0,907 (+13.0%)
Clase 18	No	0,914	0,902	No	0,932	0,911	Lineal	No	0,932	1,000 (+8.9%)

Tabla 36 - Mejores resultados de cobertura y precisión para todas las técnicas y clases estudiadas. La columna "Over." muestra el grado de oversampling utilizado. La columna de precisión para SVM incluye el incremento en precisión sobre el resto de técnicas.

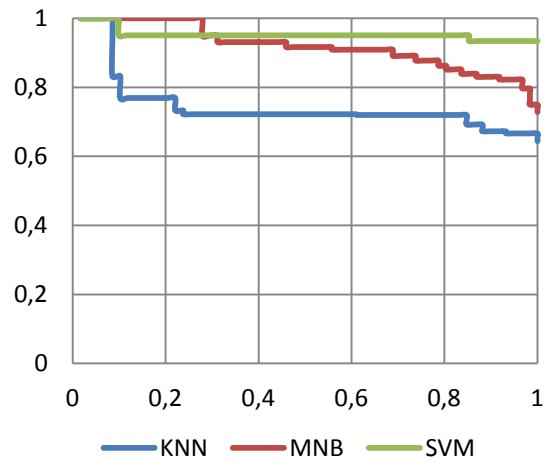
Clase 3 - Curva PR



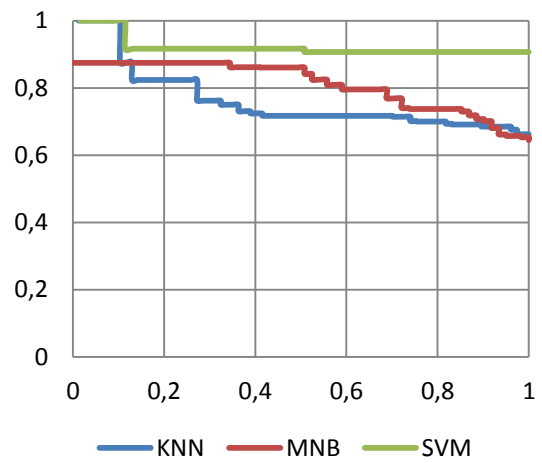
Clase 33 - Curva PR



Clase 25 - Curva PR



Clase 15 - Curva PR



Clase 18 - Curva PR

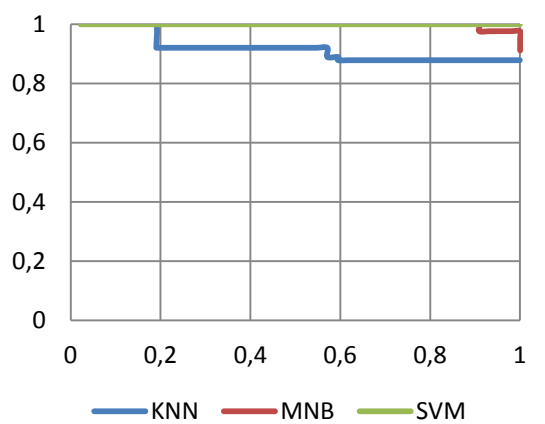


Tabla 37 - Curvas PR para los mejores clasificadores de cada técnica para todas las clases

6.7 Análisis de la composición de clasificadores

En este apartado presentamos los datos combinados de los 5 mejores clasificadores binarios obtenidos en un clasificador compuesto. Esto nos dará una idea aproximada de la capacidad real del sistema. Los resultados se han obtenido recalculando los modelos en un dataset ampliado con las muestras disponibles a fecha actual. El nuevo dataset cuenta con 3791 muestras con la siguiente distribución:

	Entrenamiento 80%	Test 20%
Clase 3	981	248
Clase 33	307	83
Clase 25	321	63
Clase 15	330	79
Clase 18	224	49
Otros	869	237
Total	3032	759

Para la combinación de los resultados individuales de los clasificadores binarios hemos optado por utilizar el valor de certeza de salida del clasificador. De esta forma, en caso de que más de un clasificador binario de positivo se elige como ganador el clasificador que haya entregado un mayor nivel de certeza. Hemos optado por esta opción en este experimento porque creemos interesante ver si este enfoque es una alternativa viable a nuestro planteamiento inicial, que trataba de evitar el problema del solapamiento espacial entre clasificadores binarios descartando las muestras con múltiples positivos.

La Ilustración 14 muestra la curva de precisión/cobertura para el clasificador compuesto. Para un 100% de cobertura el clasificador obtiene una precisión del 76,36%. La curva PR muestra un comportamiento por encima del 90% de precisión hasta una cobertura del 46,74%.

Métricas básicas del clasificador compuesto	
Accuracy	68,68%
Precision	76,36%
Recall	78,49%
True positives	365
False positives	113
False negatives	100
True negatives	181

Curva PR - Composición clasificadores

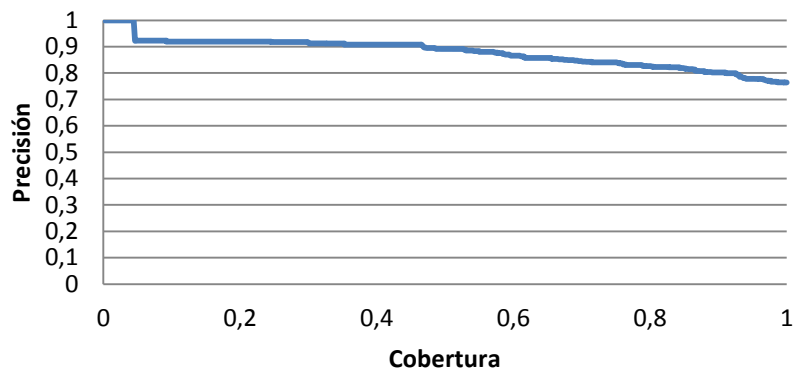


Ilustración 14 - Curva precisión recall para la clase positiva (se envía email automático) para el clasificador global formado por la composición de los clasificadores binarios de cada clase.

En primer lugar, y relativo a la activación de múltiples clasificadores binarios para una entrada, vemos que en el 4.87% de los casos, 37 muestras, *exactamente* dos clasificadores dan una respuesta positiva simultánea, y nunca más de 2 clasificadores dan un positivo simultáneamente. De estas 37 muestras la opción de elegir la salida con más certeza toma la decisión correcta en 24 de los 37 positivos obtenidos (64,86%). Este error supone un 2.5% del total de muestras que deberían haber generado una respuesta automática del sistema. En nuestra opinión, si el solapamiento espacial es pequeño es preferible optar por descartar la respuesta, si es grande o no aceptable es preferible optar por el clasificador con mayor certeza o en combinación con un umbral.

En cuanto a los errores, los más significativos para el problema son los falsos positivos. Estos errores pueden ser debidos a que el sistema emite una respuesta automática cuando no debería haber emitido ninguna o cuando envía una respuesta de una categoría errónea. En las 759 muestras clasificadas el sistema presenta 113 falsos positivos. Para el conjunto de test la proporción de los dos tipos de error mencionados está repartida al 50% (56 casos vs 57 casos).

Si analizamos el caso de las 57 veces que el sistema responde una clase errónea vemos:

- En 2 de las ocasiones (3.50%) el sistema ofrece una respuesta más precisa que la ofrecida por los operadores.
- En 3 ocasiones (5,26%) la respuesta del sistema es diferente pero podría ser una alternativa perfectamente válida.
- En 29 (50,88%) ocasiones la respuesta podría ser válida aunque menos precisa.
- Del resto de errores en 2 ocasiones el preprocesado del mensaje no es correcto lo que puede ser la causa del error.

Si analizamos el caso de las 56 veces que el sistema responde cuando no debería hacerlo, vemos que en más del 50% (29 ocasiones) el sistema opta por retornar la clase mayoritaria como respuesta, que aunque puede no ser precisa es la mejor de las alternativas posibles.

Este último comportamiento nos hace plantearnos si el conjunto y estructura de plantillas creado por el centro de atención es el idóneo para ofrecer una respuesta automática. Este aspecto es algo a lo que no hemos prestado atención asumiendo que el conjunto de plantillas disponible era el más adecuado para la tarea que realizan diariamente los operadores. Sin embargo, el hecho de que la clase 3, clase mayoritaria en el dataset, sea una especie de cajón de sastre que puede ajustarse con menos o más precisión a más respuestas que las de su clase, crea una línea difusa de cuando es más adecuada una respuesta u otra. Mostraremos este problema con un ejemplo, los dos casos donde la respuesta del sistema es más precisa que la de los operadores se corresponde en concreto a dos casos donde los operadores respondieron con la respuesta general de “cómo comprar” (clase 3), mientras que el sistema eligió adecuadamente la clase 33 “reservas agotadas”. Estos son los mensajes recibidos:

ID 27578: “Buenos días: Organizo un viaje con mis alumnos en Madrid en marzo de 2016, somos 49 alumnos + 4 profesores y no puedo reservar las entradas para todo el grupo ya que sólo les quedan 30 entradas disponibles el 07 de marzo de 2016 y ninguna el 10 de marzo de 2016 por la mañana, ya que son los 2 únicos días durante los cuales estamos en Madrid. Gracias por su comprensión y ayuda para que los alumnos pueden visitar el Palacio Real. Espero su pronta confirmación para confirmar las 30 entradas porque si no es posible no hace falta que las tengan. Quedamos en contacto. Un cordial saludo.”

ID 25693: “Buenos días, estoy intentando reservar para una visita de grupo educativo. La fecha que nos viene bien es el 19 de Febrero del 2016. No puedo hacerlo.”

En resumen, por un lado disponer de una clase que pueda ser una alternativa viable pero imprecisa para muchas respuestas se convierte en una red de seguridad que atenúa (no elimina) los errores del clasificador, mientras que por otro dificulta el entrenamiento al no ofrecer límites de separación claros sobre qué muestras caen en qué clases. En cualquier caso, una revisión detallada y reestructuración de las plantillas disponibles, si necesaria, debería ser una etapa previa al abordar estos trabajos ya que no se puede garantizar que los operadores hayan abordado la construcción de las mismas de una forma adecuada o sistemática.

7 Conclusiones

Un análisis frío y sin perspectiva de los datos obtenidos en la clasificación global del sistema nos puede hacer ver que la utilización de técnicas de clasificación, únicamente sobre el contenido textual de los mensajes, no es viable para la construcción de un sistema de respuesta automática de correo. La utilización de un mecanismo de seguridad ante respuestas erróneas, para que los clientes puedan reencolar sus peticiones, no es una alternativa aceptable con tasas de error superiores al 5%, y seguramente muchos centros de atención al cliente necesiten tasas aún menores. Sin embargo, la curva PR del clasificador compuesto muestra tasas de precisión del 90% para una cobertura del 45% de los mensajes recibidos. Estos datos nos hacen pensar que un mejor preprocesado de los mensajes, la utilización de otros tipos de kernels, ajustes más finos de parámetros, variaciones en la combinación de clasificadores, utilización de umbrales u otro tipo de técnicas puede hacer viable la respuesta con alta precisión manteniendo una cobertura entre el 30%-40%. En conclusión, en este punto, no podemos descartar ni aceptar por completo nuestra hipótesis de lograr una respuesta automática de email basada exclusivamente en el contenido textual mediante la composición de clasificadores binarios. Por otro lado, nuestra restricción de minimizar la cantidad de información de dominio utilizada para facilitar la generalización del problema a otros escenarios es autoimpuesta. Si eliminamos esta restricción y utilizamos en el proceso información de dominio directamente en los clasificadores estamos seguros que las tasas de precisión y cobertura se dispararán. Así, información como si el cliente está registrado o no, tipo de cliente, antigüedad, número de ventas, etc. complementarían la base actual basada únicamente en contenido textual a costa de perder capacidad de generalización. No obstante es posible que muchos de estos atributos, si son correctamente elegidos, sean también opciones naturales en otros dominios.

Entrando en detalle en las problemáticas abordadas, quizás el resultado más sorprendente es relativo al balanceo de datos. A priori, el balanceo de datos parecía un elemento crucial que debía ser abordado dada la influencia que tiene en la capacidad de clasificación. Sin embargo, los resultados muestran que para nuestro conjunto de datos la utilización de técnicas de oversampling tiene una utilidad marginal. La utilización de SMOTE sobre la clase positiva no arroja ninguna mejora en la capacidad de los clasificadores SVM e incluso se puede decir que la empeora muy ligeramente. Es posible que esta irrelevancia del oversampling sea provocada por el punto donde hemos decidido aplicarla, después de las transformaciones de los datos, y no en origen sobre el contenido textual. La interpolación para generar nuevos mensajes sintéticos quizás podría haber generado nuevas muestras más significativas si se hubiese realizado añadiendo o quitando palabras y no sumando incrementos aleatorios tras las transformaciones de los mensajes a un espacio continuo.

De las tres técnicas estudiadas SVM es con un margen notable la mejor herramienta para abordar una respuesta automática de email para el tamaño de corpus evaluado. En todos los casos con un pequeño ajuste en la variante lineal se ha mostrado superior a las otras dos técnicas. Sólo MNB para tasas de cobertura muy bajas puede ofrecer resultados similares de precisión. Se podría discutir si con un número menor de muestras el comportamiento de SVM sería el mismo, pero en un escenario donde el crecimiento continuo de información es inherente al problema, plantearse otras opciones a medio y largo plazo no parece razonable.

Los resultados obtenidos para el caso de estudio también muestran que sólo usar clasificación lineal no es suficiente para abordar la tarea que nos ocupa. Los kernels polinomiales de grado 2 han mostrado una clara ventaja a través de la mayoría de clases. Es posible que otro tipo de kernels más sofisticados puedan proporcionar mejoras en la precisión. Adicionalmente, la utilización de calibradores para obtener una probabilidad de salida, combinada con un umbral que garantiza un mayor nivel de certeza al determinar una respuesta positiva, aporta poco en el incremento de la precisión. Cuando la separación espacial entre clases es pequeña estos umbrales generan fuertes reducciones en la cobertura con pequeñas ganancias de precisión.

En el lado opuesto de SVM está KNN con distancia Jaccard. A priori la intuición podría indicar que retornar la clase del mensaje más similar como resultado de la clasificación puede ser una buena opción, la realidad de los resultados muestra que esto está muy lejos de ser verdad. Los motivos para estos resultados pueden ser dispares, el primero y más obvio es que la propia naturaleza y número de los mensajes de la clase positiva puede no ser adecuada para la simplicidad de Jaccard. Otro factor que hemos visto que puede provocar este comportamiento es el fuerte preprocesado de los mensajes que reduce considerablemente su longitud y homogeniza su contenido haciendo que mensajes más cortos se solapen total o parcialmente sobre mensajes más largos de otras categorías.

La capacidad de clasificación mostrada en algunas clases es muy buena, pero no deja de ser un resultado sobre una foto fija en un punto del tiempo. A esta foto fija hay que añadir que algunas de las transformaciones aplicadas a los datos necesitan de información de todo el dataset. En concreto las transformaciones de longitud y TF-IDF necesitan de las informaciones de frecuencia de todos los mensajes. Cada nuevo mensaje recibido debe ser transformado de acuerdo a esta información, y la información que el nuevo email aporta al dataset modifica la transformación que se hubiese aplicado al resto de datos generando una posible desviación respecto del modelo calculado. Creemos que para nuestro dominio, en pequeños periodos de tiempo, el posible error introducido no supone un problema. Por ejemplo, asumiendo un flujo constante de recepción de mensajes, en una semana se habría incrementado el tamaño total del dataset en un 2% en el peor de los casos. En este contexto una

actualización batch de los modelos de cada clasificador es la mejor opción. Una actualización en periodos de tiempo más pequeños tiene un coste computacional alto en relación a la ganancia que se puede obtener en los modelos. Plantearse un proceso de online learning no parece razonable para nuestro escenario de aplicación y volumen de datos.

8 Futura mejoras

Creemos que la etapa de preprocesado de datos para la construcción del corpus debe recibir mucha más atención. En concreto al menos estas áreas deberían ser revisadas:

- Identificación de entidades con nombre. Aunque la aplicación del lexicón y la eliminación de términos de baja frecuencia palián parcialmente esta problemática, al menos la identificación de nombre propios, nombre de instituciones o empresas y lugares debería estar cubierta.
- Eliminación de despedidas, firmas y citados. Hemos visto que aunque esta eliminación por norma general hace un buen trabajo, en ocasiones puede equivocarse y eliminar parte del contenido del mensaje generando duplicados o no aplicarse conservando mensajes excesivamente largos. La inclusión de criterios de control de ubicación dentro del texto antes de realizar el borrado o una búsqueda de despedidas más flexible podría reducir o completamente eliminar estas problemáticas.
- Eliminación de mensajes outliers. Aparte de que este punto está estrechamente ligado al anterior, creemos que la eliminación de mensajes excesivamente cortos u excesivamente largos tras el preprocesado, bien basándose en longitudes promedios de clase o distancia Jaccard podría conducir a la generación de mejores modelos de clasificación.
- Filtros de idiomas. Estos filtros en ocasiones dejan pasar mensajes de otros idiomas que entran al proceso de construcción del corpus. Seguramente de nuevo la eliminación de términos de baja frecuencia resuelva en gran medida este problema, pero sería deseable una mejor precisión en la selección del idioma español.

Una evolución natural al problema aquí presentado sería utilizar técnicas de extracción de información y construir la clasificación sobre la salida del sistema de extracción en vez de directamente sobre el contenido textual. Este planteamiento tendría numerosas ventajas, aunque en contrapartida el coste final de construcción y mantenimiento se dispararía en relación al enfoque aquí presentado. Entre las ventajas podemos citar:

- Menor dependencia del volumen de datos disponible.

- Los procesos de clasificación estarían alineados con acciones relevantes del dominio y no únicamente a nivel superficial con palabras. Aunque actualmente un clasificador puede llegar a inferir esa “acción de dominio” a partir de la entrada, la dependencia del dataset disponible es absoluta.
- La extracción de información favorecería no sólo una respuesta automática más precisa si no también la posibilidad de ejecutar acciones no supervisadas en el sistema.
- Aunque no se pudiese responder automáticamente o ejecutar acciones, los operadores podrían recibir un email “aumentado” con posibles acciones asociadas accesibles mediante hiperenlaces que reducirían los costes de gestión de los casos.

9 Bibliografía

- [1] William W. Cohen. Fast effective rule induction. In Machine Learning: Proceedings of the Twelfth International Conference, Lake Tahoe, California, 1995. Morgan Kaufmann
- [2] W. Cohen, Learning rules that classify e-Mail, in: Proc. AAAI Symposium on Machine Learning in Information Access, 1996, pp. 18– 25
- [3] J. Helfman and C. Isbell. Ishmail: Immediate identification of important information. AT&T 1995
- [4] Mario Lenz, Hans-Dieter Burkhard. CBR for Document Retrieval: The FALLQ Project. ICCBR '97 Proceedings of the Second International Conference on Case-Based Reasoning Research and Development Pages 84-93
- [5] F. Ciravegna, A. Lavelli, N. Mana, J. Matiassek, L. Gilardoni, S. Mazza, M. Ferraro, W. Black, F. Rinaldi, D. Mowatt, Facile: classifying text integrating pattern matching and information extraction, in: Proceedings of IJCAI-99, Sotckholm, Sweden, 1999, pp. 890-895
- [6] Philip J. Hayes, Steven P. Weinstein. CONSTRUE/TIS: A system for content-based indexing of a database of news stories. IAAI '90 Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence Pag. 49-64
- [7] Jefferson Provost. Naïve-Bayes vs. Rule-Learning in Classification of Email. 1999
- [8] Mehran Sahami, Susan Dumais, David Heckerman y Eric Horvitz. A Bayesian Approach to Filtering Junk E-Mail. 1998
- [9] Paul Graham. A plan for spam. <http://www.paulgraham.com/spam.html>. 2002
- [10] Paul Graham. Better Bayesian filtering. <http://www.paulgraham.com/spam.htm4> 2003
- [11] Jason D. M. Rennie, IFILE: Application of Machine Learning to E-Mail Filtering. In: Proc. KDD 2000 Workshop on Text Mining, Boston
- [12] P. Pantel, D. Lin. SpamCop: a spam classification and organization program, in: Proc. AAAI Workshop on Learning for Text Categorization 1998
- [13] Banko, Michele, and Eric Brill. Scaling to very very large corpora for natural language disambiguation. 2001. In Proc. ACL.
- [14] H. Drucker, D. Wu, and V. N. Vapnik. Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10(5), 1999.

- [15] Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine D. Spyropoulos and Panagiotis Stamatopoulos. Learning to Filter Spam Email - A comparison of a Naive Bayesian and a Memory-Based Approach 2000
- [16] Gulsen Eryigit , A. Cuneyd Tantug. A comparison of support vector machines, memory based and naïve bayes techniques on spam recognition. Proceedings of the 32rd IASTED Interntational Multi-Conference Artificial Intelligence and Applications. 2005 February 14-16, Innsbruck, Austria.
- [17] Jake D. Brutlag, Christopher Meek. Challenges of the Email Domain for Text Classification. Proceeding ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning Pages 103-110
- [18] Richard B. Segal, Jeffrey O. Kephart. MailCat: an intelligent assistant for organizing e-mail. AGENTS '99 Proceedings of the third annual conference on Autonomous Agents. Pages 276-282
- [19] Patrick Pantel, Dekang Lin. SpamCop: A Spam Classification & Organization Program. In Learning for Text Categorization: Papers from the 1998 Workshop.
- [20] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, David R. Karger, Tackling the Poor Assumptions of Naive Bayes Text Classifiers. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.
- [21] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze. Introduction to Information Retrieval (book). Cap. Choosing what kind of classifier to use. <http://nlp.stanford.edu/IR-book/html/htmledition/choosing-what-kind-of-classifier-to-use-1.html>
- [22] Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001), L. Lee and D. Harman (Eds.), pp. 44–50, Carnegie Mellon University, Pittsburgh, PA, USA, 2001.
- [23] Kjersti Aas and Line Eikvil. Text Categorisation: A Survey. 1999
- [24] Yiming Yang. An Evaluation of Statistical Approaches to Text Categorization. Journal Information Retrieval archive Volume 1 Issue 1-2, 1999 Pages 69-90
- [25] Yiming Yang, Jan O. Pedersen, A Comparative Study on Feature Selection in Text Categorization 1997

- [26] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceeding ECML '98 Proceedings of the 10th European Conference on Machine Learning Pages 137-142
- [27] Pedro Domingos., Michael Pazzani. On the Optimality of the Simple Bayesian Classifier under ZeroOne Loss. Published in Journal Machine Learning - Special issue on learning with probabilistic representations archive Volume 29 Issue 2-3, Nov./Dec. 1997 Pages 103 - 130
- [28] Celso Antonio Alves Kestner. Support Vector Machines and Kernel Functions for Text Processing
- [29] SMOTE: synthetic minority over-sampling technique. Published in Journal of Artificial Intelligence Research archive Volume 16 Issue 1, January 2002 Pages 321-357
- [30] John Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. 1998
- [31] A. M. Kibriya, E. Frank, B. Pfahringer, G. Holmes. Multinomial naive Bayes for text categorization revisited. In G.I. Webb & Xinghuo Yu(Eds.), Proceedings of 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004.(pp. 488-499). Berlin: Springer.
- [32] Gaspar P1, Carbonell J, Oliveira JL. On the parameter optimization of Support Vector Machines for binary classification. 2012
- [33] Seongwook Youn, Dennis McLeod. A Comparative Study for Email Classification. Advances and Innovations in Systems, Computing Sciences and Software Engineering pp 387-391. 2007
- [34] Rahul Malik, L.Venkate Subramaniam, Saroj Kaushik. Automatically Selecting Answer Templates to Respond to Customer Emails. Proceeding IJCAI'07 Proceedings of the 20th international joint conference on Artificial intelligence Pages 1659-1664. Hyderabad, India — January 06 - 12, 2007

10 Anexos

10.1 Lexicón

Cada entrada del lexicón tiene 3 líneas, según la siguiente estructura:

- Línea 1. Descripción de la entrada.
- Línea 2. Expresión regular a aplicar.
- Línea 3. Etiqueta de reemplazo.

Email

```
(&#92;\w\.-]+)@(&#92;\w\.-+)((\.-(\w){2,3})+)  
<TFMEmail>
```

Urls

```
((([A-Za-z]{3,9}:(?://\?|(?::[-;=&#92;\+\$, \w]+@)?[A-Za-z0-9.-]+|(?::www.|[-;=&#92;\+\$, \w]+@)[A-Za-z0-9.-]+)((?://[\+\~%\/\.-_])*)?\?(\?::[-\+=&#92;\+\$, \w_]*)#?(?::[\w]*))?)  
<Url>
```

1.Palacio real de madrid

```
(&#92;\s*(palacio)?\s*(real)?\s*(de)?\s*madrid|\s*palacio\s+real)  
<TFMMuseo>
```

2.Palacio de Aranjuez

```
(&#92;\s*(palacio)?\s*(real)?\s*(de)?\s*aranjuez)  
<TFMMuseo>
```

3.Palacio de La Granja de San Ildefonso

```
(&#92;\s*(palacio)?\s*(real)?\s*(de)?\s*(la\s*granja\s*(de)?\s*(san|s\.-|s)\s*ildefonso|\s*la\s*granja|(san|s\.-|s)\s*ildefonso))  
<TFMMuseo>
```

4.Palacio del Pardo

```
(&#92;\s*(palacio)?\s*(real)?\s*(del|de\s*el)?\s*pardo)  
<TFMMuseo>
```

5.Palacio de la Almudaina

```
(&#92;\s*(palacio)?\s*(real)?\s*(de)?\s*(la)?\s*almudaina)  
<TFMMuseo>
```

6.Palacio de Riofrio

```
(&#92;\s*(palacio)?\s*(real)?\s*(de)?\s*riofr[í]o)  
<TFMMuseo>
```

7.El Escorial

```
(&#92;\s*(real)?\s*(monasterio|convento)?\s*(de)?\s*((san|s\.-|s)\s*lorenzo|s\.-l\.-)?\s*(de\s*el|del|el)?\s*escorial)  
<TFMMuseo>
```

8.Monasterio Yuste

```
(&#92;\s*(monasterio|convento)?\s*(de)?\s*((san|s\.-|s)\s*j[er]o[ón]nimo)?\s*(de)?\s*yuste)  
<TFMMuseo>
```

9.Monasterio Las Huelgas

```
(&#92;\s*(monasterio|convento)?\s*(de)?\s*((santa|sta\.-|sta)\s*mar[í]a)?\s*(la)?\s*(real)?\s*(de)?\s*(las)?\s*huelgas)  
<TFMMuseo>
```

10.Monasterio Santa Clara

```
(&#92;\s*(real)?\s*(monasterio|convento)?\s*(de)?\s*((santa|sta\.-|sta)\s*clara|(de)?\s*tordesillas)|\s*(de)?\s*tordesillas)  
<TFMMuseo>
```

Monasterio Descalzas

```
(&#92;\s*(monasterio|convento)?\s*(de)?\s*(las)?\s*descalzas\s*(reales)?|\s*descalzas)  
<TFMMuseo>
```

Monasterio Encarnación

```
(&#92;\s*(real)?\s*(monasterio|convento)?\s*(de)?\s*(la)?\s*encarnaci[ón]  
<TFMMuseo>
```

Valle de los Caidos

```
(\s*(abad[íi]a)?\s*(benedictina)?\s*(de)?\s*(la)?\s*(santa)?\s*(cruz)?\s*(del)?\s*
*valle\s*(de)?\s*(los)?\s*ca[íi]dos)
<TFMMuseo>
Panteon
\s*pante[óó]n\s*(de)?\s*(los)?\s*(hombres)?\s*(ilustres)?
<TFMMuseo>
Museo Reina Sofia
\s*(museo?)\s*(reina)?\s*(sof[íi]a)
<TFMMuseo>
Casa de Labrador o principe
\s*(casa|casita)(\s*(del|de))?(labrador|pr[íi]ncipe|infante)
<TFMMuseo>
Albaran
\s*(vn|VN)\d+
<TFMAlbaran>
Localizador
\s*(00|[a-zA-Z]0)\d+
<TFMLocalizador>
Fecha larga
\s*\d+\s*(y\s*\d+)?\s*(de)?\s*(enero|febrero|marzo|abril|mayo|junio|julio|agosto|
septiembre|octubre|noviembre|diciembre)\s*(de)?\s*(\d+)?
<TFMFecha>
Fecha corta
\s*\d+[-/\._]\d+([-/\._]\d+)?
<TFMFecha>
Mes
\s*(en|de)\s*(enero|febrero|marzo|abril|mayo|junio|julio|agosto|septiembre|octubr
e|noviembre|diciembre)
<TFMFecha>
Día
\s*((lunes|martes|mi[eé]rcoles|jueves|viernes|s[aá]bado|domingo)|fin\s*de\s*seman
a)
<TFMFecha>
Hora larga
\s*((antes|despu[eé]s)\s*de\s*las\s*\d+|por\s*la\s*mañana|por\s*la\s*tarde)
<TFMHora>
Hora larga 2
\s*(\d{1,2}|una|dos|tres|cuatro|cinco|seis|siete|ocho|nueve|diez|once|doce)(\s*de
)?(\s*la)?\s*(ma[ññ]ana(s)?|tarde(s)?)
<TFMHora>
Hora corta
\s*(\d+\s*(horas|hrs|hrs\.|hs|am|pm|h\.|h)|\d+[:\.,]\d+([:\.,]\d+)?)\s*(horas|hrs|
hrs\.|hs|am|pm|h\.|h)?)
<TFMHora>
NumeroIntegrantes
\s*\d+(\s*-
\s*\d+)?\s*(persona(s)?|entrada(s)?|billete(s)?|estudiante(s)?|ticket(s)?|alumn(o
s|as|o|a)?|profesor(es)?|niñ(os|as|o|a)|adult(os|as|o|a)?|pax|senior(s)?|jubilad
os|as|o|a)?)
<TFMNumeroIntegrantes>
Importe
\s*(€\d+|\d+\s*(€|euro(s)?))
<TFMImporte>
NIF
\s*([a-zA-Z]\d+|\d+[a-zA-Z])
<TFMNIF>
Telefono
\s*(\+\s*\d+)?\s*(\d{2}\s*\d{3}\s*\d{2}\s*\d{2}|\d{3}\s*\d{3}\s*\d{3}|\d{2}\s*\d{
7}|\d{9})
<TFMTelefono>
Codigo postal
\b[0-5]\d{4}\b
```

```
<TFMCodigoPostal>
Tarifa Basica y Basica+Expo (+expo es opcional)
\s*(b[aa]sica(s)?((\s*\+\s*|\s*y\s*)(e[sx]posici[óo]n|e[sx]po)))?)
<TFMTarifa>
Tarifa Coleg. y Coleg.Pago
\s*(coleg.pago|colegios)
<TFMTarifa>
Tarifa 5-16 años + Expo (+expo es opcional)
\s*(5-16\s*a[ñn]o(s)?((\s*\+\s*|\s*y\s*)(e[sx]posici[óo]n|e[sx]po)))?)
<TFMTarifa>
Tarifa Estudiantes + Expo (+expo es obligatorio)
\s*(estudiante(s)?((\s*\+\s*|\s*y\s*)(e[sx]posici[óo]n|e[sx]po)))
<TFMTarifa>
Tarifa Mayor 65 + Expo
\s*(mayor\s*65(\s*a[ñn]o(s)?)((\s*\+\s*|\s*y\s*)(e[sx]posici[óo]n|e[sx]po)))?)
<TFMTarifa>
Edad
\s*\d+(\s*-\s*\d+)?\s*año(s)?
<TFMEdad>
Bono
\s*bono(s)?
<TFMBono>
Gratuidad
\s*(gratis|gratuidad|gratuit(os|as|o|a))
<TFMGratuidad>
Desempleados
\s*desemplead(os|as|o|a)
<TFMDesempleado>
Jubilados
\s*jubilad(os|as|o|a)
<TFMJubilado>
Discapacitado
\s*discapacitad(os|as|o|a)
<TFMDiscapacitado>
Abadia, monasterio, museo
\s*(abad[ií]a(s)?|monasterio(s)?|biblioteca|cocina(s)?|casa(s)?|casita(s)?|palaci
o(s)?)
<TFMMuseo>
Cursos escolares
\s*(1|2|3|4|1º|2º|3º|4º)(\s*de)?(\s*la)?\s*[Ee][Ss][Oo]
<TFMCursoEscolar>
Online
\s*on[- ]line
<TfmOnline>
Entradas
\s*(entrada(s)?|ticket(s)?|billete(s)?|boleto(s)?)
<TFMEntrada>
Numero
\s*\d+
<TFMNumero>
```

10.2 Librerías y herramientas

En este trabajo hemos utilizado las siguientes herramientas:

Microsoft Visual Studio 2012 Professional, .net Framework 4.5 y lenguaje C#

- Para la extracción de mensajes del sistema de tickets.
- Preprocesado de los mensajes apoyándose en la utilización de la librería OpenNLP mediante IKVM.
- Construcción del dataset de entrada a WEKA
- En conjunción con la librería Accord .net v3.3 para la implementación de KNN sobre distancia Jaccard.

WEKA v3.8.0

- Para la generación de los conjuntos de test y entrenamiento.
- Para la vectorización de los mensajes.
- Para las transformaciones TF-IDF y normalización.
- Para la realización del sobremuestreo.
- Para la evaluación de las clasificaciones basadas en SVM y MNB y ajuste de parámetros.

JetBrains IntelliJ IDEA 2016.3 y Java 8 con las librerías de WEKA

- Para los ajustes de la razón de probabilidad de salida de SVM.
- Para evaluar la capacidad del clasificador final combinando los clasificadores binarios individuales.