Análisis de estrategias para clasificar contenidos en foros de discusión: Un caso de estudio

Valeria Zoratto[+], Gabriela N. Aranda[+], Sandra Roger[*], Alejandra Cechich[+]

Grupo GIISCo[+], Grupo GILIA[*]
Facultad de Informática, Universidad Nacional del Comahue
Buenos Aires 1400 (8300) Neuquén, Argentina
{vzoratto|gabriela.aranda|roger}@fi.uncoma.edu.ar

Resumen La información que generan las consultas realizadas en foros especializados puede ser de gran utilidad para otros usuarios que tengan problemas similares. Nuestra propuesta es capturar, mantener y analizar hilos de discusión existentes en foros técnicos para, dado un problema particular, sugerir un conjunto de soluciones exitosas en menos intentos que utilizando buscadores multipropósito tradicionales. En este trabajo se presenta una serie de casos de estudio enfocados en hilos de un foro técnico sobre el uso del lenguaje de programación Java y se analizan estrategias para clasificar dichos hilos y relacionarlos con las clases Java correspondientes.

1. Introducción

Los foros de discusión disponibles en la Web sobre temáticas relacionadas al desarrollo y mantenimiento de software, contienen un amplio conocimiento sobre diferentes problemáticas cotidianas, por lo que hacer un análisis de dicha información es algo deseable y valioso [1]. Los usuarios generalmente utilizan motores de búsqueda multipropósito para acceder a dicha información, y suelen recorrer varias páginas buscando un problema similar al suyo. Este proceso puede llevar al usuario a visitar distintas páginas antes de encontrar una propuesta de solución para su pregunta, y a veces es necesario probar varias de ellas hasta obtener la correcta. Para ello se propone avanzar en la línea presentada en [2] a fin de lograr el pre-procesamiento de la pregunta del usuario y, basado en el análisis previo de hilos de discusión clasificados por tema, ofrecerle un conjunto ordenado de soluciones con mayor probabilidad de éxito.

Como se muestra en la Figura 1, para reutilizar la información disponible en los foros de discusión, es necesaria una primera etapa durante la cual se clasifiquen los hilos de dichos foros. Para ello se propone utilizar un conjunto de entidades más reconocibles, llamados *documentos de referencia*. A modo de ejemplo, en este artículo se ha analizado una colección de hilos de discusión sobre uso del lenguaje Java, y como *documentos de referencia* un repositorio de especificación de clases de dicho lenguaje.

Teniendo en cuenta este objetivo, en la Sección 2 se describe el diseño de una familia de casos de estudio para evaluar estrategias que logren una mejora

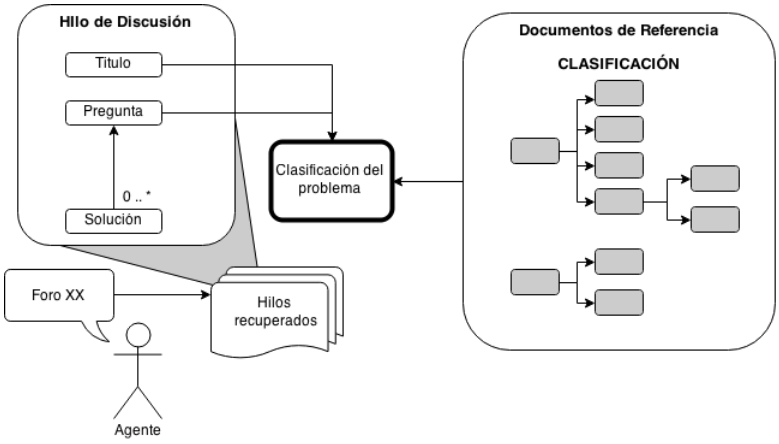

Figura 1. Proceso de clasificación de hilos de acuerdo a documentos de referencia

en la clasificación de hilos de discusión. Luego, en la Sección 3 se presenta el desarrollo de la primera etapa de dicha familia. Posteriormente, en la Sección 4 se analizan los resultados obtenidos y, finalmente, se presentan las conclusiones y líneas de trabajo futuro.

2. Diseño de una familia de casos de estudio

Un caso de estudio se desarrolla con el objetivo de investigar una entidad o fenómeno particular en el contexto de la vida real [3]. Los casos de estudio son estudios observacionales, es decir, se llevan a cabo mediante la observación de un proyecto o actividad que está en marcha, mientras que los experimentos son estudios en entornos controlados [4]. Dado que el objeto real de estudio es la información disponible en la Web, se ha diseñado una estrategia empírica basada en el estudio de casos. En la Tabla 1 se muestra la definición de dicha estrategia según la plantilla propuesta por el método GQM [5].

Tabla 1. Definición del caso de estudio según plantilla GQM

Analizar	Foros de discusión sobre aspectos técnicos disponibles en la Web
Con el propósito de	Clasificar los hilos de los foros para relacionarlos a una entidad más reconocible (documentos de referencia)
Con respecto a	Determinar el grado de relación entre distintos hilos de foros
Desde el punto de vista de	Usuario externo de un foro
En el contexto	Información disponible en la Web de forma pública

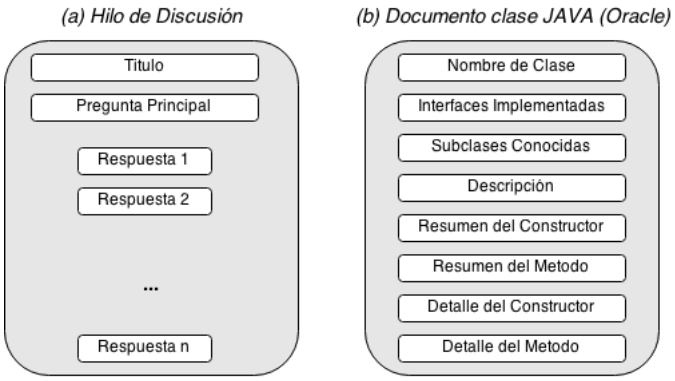

Figura 2. Estructura de los documentos utilizados en el caso de estudio

Como esta etapa se ha enfocado en el estudio de hilos de foros de discusión sobre uso del lenguaje Java, se ha seleccionado como conjunto de *documentos de referencia* la especificación de clases de Oracle (versión 5)[1].

En la Figura 2 se muestra un ejemplo de la estructura de los documentos involucrados en este estudio. Como se puede observar, un hilo de discusión tiene un título y una pregunta principal, que han sido los disparadores del hilo, y luego una serie de respuestas a dicha pregunta (pedidos de aclaración, propuesta de soluciones, etc). Por el otro lado, un documento Oracle tiene el nombre de la clase (*Class Name* en el documento Oracle) y a continuación se listan las interfaces que implementa (*Implemented Interfaces*), subclases conocidas (*Known Subclasses*), etc.

En base a la información contenida en las secciones presentes en cada tipo de documento (hilos de discusión y documentos Oracle), y a la tarea objeto de esta investigación de clasificar los hilos de foros de discusión de la manera más apropiada, se han definido las siguientes hipótesis:

- HIPÓTESIS A: Utilizar mayor cantidad de información sobre cada una de las clases Java documentadas en Oracle permite clasificar los hilos de discusión relacionados a ellas de forma más precisa.
- HIPÓTESIS B: Utilizar más información sobre el problema explicado en los hilos de discusión permite clasificarlos de forma más precisa respecto a los documentos Oracle de las clases Java.

En las siguientes secciones se explican los pasos realizados para implementar las distintas etapas de dicha estrategia.

3. Desarrollo de los casos de estudio

En base al diseño explicado anteriormente, se han definido una serie de pasos para comprobar las hipótesis propuestas mediante casos de estudio:

[1] http://docs.oracle.com/javase/1.5.0/docs/

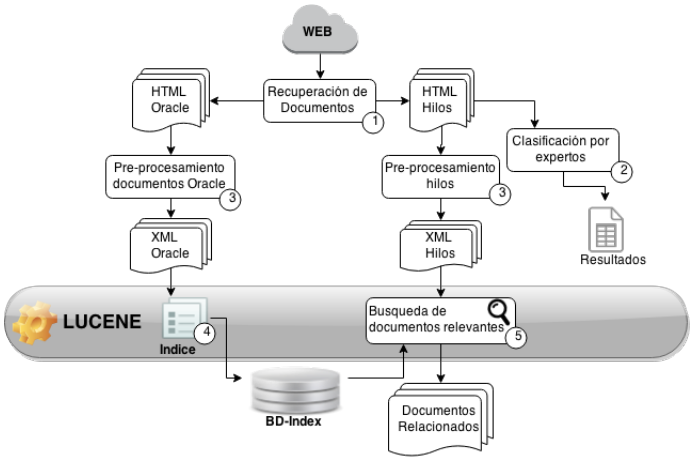

Figura 3. Fases del desarrollo de los casos de estudio

- Fase 1. Recuperación de documentos
- Fase 2. Clasificación por expertos
- Fase 3. Pre-procesamiento de documentos
- Fase 4. Indexación de documentos de referencia
- Fase 5. Búsqueda de documentos relevantes

En la Figura 3 se muestra una vista de dichas fases, así como de las entradas y salidas de cada una de ellas, que serán explicadas con más detalle en las secciones siguientes. Si bien las fases se han numerado y presentado de manera secuencial, se puede observar que algunas de ellas pueden realizarse de manera paralela. Por ejemplo, la Fase 2 (clasificación por expertos) puede realizarse a la vez que la Fase 3, 4 y 5, dado que su evolución y resultado no depende de otras fases o de resultados intermedios. En las secciones siguientes se irán explicando las características más importantes de cada fase y su implementación en los casos de estudio realizados.

3.1. Fase 1: Recuperación de documentos

En esta etapa se han recuperado de la Web, los documentos a utilizar durante los casos de estudio propuestos. Por tratarse de pruebas preliminares, se se ha restringido el estudio a un único foro de discusión, *Stack Overflow*[2], ampliamente utilizado por la comunidad de programadores. Haciendo uso de la funcionalidad del buscador de dicho foro, se ha realizado una consulta con la cadena: "*Integer class" AND Java*. Por el contrario, no se ha realizado ningún filtrado previo para los documentos del sitio Oracle, descargándose la especificación de todas las clases Java de la versión elegida. En la Tabla 3.1 se resumen las características

[2] http://stackoverflow.com/

Tabla 2. Características técnicas del caso de estudio

	Documentos de referencia	**Hilos de discusión**
Sitio	Oracle	Stack Overflow
Idioma	Inglés	Inglés
URL Sitio	http://docs.oracle.com/javase/1.5.0/docs/	http://stackoverflow.com/
Fecha recup.	27/11/2014	27/02/2015
Cadena búsq.	-	"Integer class" AND Java
Doc. analizados	2953 (todos los disponibles versión 1.5.0)	20 hilos

Tabla 3. Ejemplo de clasificación de los hilos realizadas por expertos

Hilo	**Muy alto**	**Alto**	**Medio**	**Bajo**	**Muy bajo**
3	Integer JTable			String Object Component	JScrollPane JCombobox JFrame
6	Integer		Long Boolean Character Double Short Float Byte		
8		Integer Character		String	BigInteger
12	Integer	Long	String		
16					String Integer
18	String	Integer			
19			Integer		Float Double

de la realización de esta fase, indicándose la fecha de descarga de los documentos utilizados, cantidad de documentos recuperados de cada tipo, etc., de acuerdo a las restricciones explicadas anteriormente.

3.2. Fase 2: Clasificación por expertos

Como primera instancia, los hilos de discusión recuperados han sido analizados para identificar, entre las clases Java mencionadas en el hilo, aquellas a las que el problema planteado por el usuario se encontraba más relacionado. A fin de conseguir una clasificación objetiva, se ha requerido el análisis consensuado de tres expertos. Dado que los hilos están escritos en lenguaje natural, la información que presentan contiene un alto grado de ambigüedad. Es por ello que el estudio se ha restringido a los primeros 20 hilos recuperados clasificados de acuerdo a la escala de relación: muy alto, alto, medio, bajo y muy bajo. Por ejemplo, para el hilo 8, se ha determinado que se encontraba relacionado en grado alto con las clases *Integer* y *Character*, con grado bajo con la clase *String*, y muy bajo con la clase *BigInteger*. En la Tabla 3 se muestra, a modo de ejemplo, el resultado de dicho análisis para algunos hilos recuperados.

Dado que las relaciones más bajas no son decisivas en la selección de documentos relevantes, se ha decidido enfocar el estudio a aquellos hilos que tengan al menos una clase Java en las categorías media, alta o muy alta. Como puede observarse en la Tabla 3, el hilo 16 no cumple con esta condición, razón por la cual el estudio posterior se ha restringido a los 19 hilos restantes.

3.3. Fase 3: Pre-procesamiento de documentos

El objetivo de esta fase ha sido preparar los documentos descargados de la Web (en formato *html*) para su posterior análisis con técnicas de recuperación de información. Para ello, se han generado nuevas versiones de dichos documentos, en formato *xml*, descartando código *html* irrelevante (por ejemplo, las etiquetas de comienzo y fin de estilo, separación de párrafos, etc.), y agregando etiquetas para diferenciar las secciones de interés en este estudio (tal como han sido detalladas en la Figura 2). Además, para evitar información innecesaria, se han descartado otras secciones con contenido irrelevante (por ejemplo, los banners de publicidad).

Finalmente, a fin de contar con información para evaluar las hipótesis planteadas, se han generado tres versiones diferentes de los documentos Oracle, considerando en cada una de ellas distintas partes del documento:

- O_a: conteniendo sólo el nombre de la clase
- O_b: conteniendo el nombre de la clase y los nombres de todos sus métodos
- O_c: contenido de todas las secciones del documento (excepto las secciones "Detalles de constructor" y "Detalle de métodos").

De manera similar, los hilos descargados del foro Stack Overflow se han pre-procesado y se han obtenido tres versiones diferentes de cada uno, considerando las siguientes secciones:

- F_1: Sólo el título del hilo
- F_2: El título del hilo y la pregunta principal
- F_3: El texto completo del hilo (título, pregunta principal, respuestas)

Dadas las distintas versiones obtenidas, tanto de los documentos Oracle (a, b y c), como de los hilos de discusión (1,2 y 3), ha sido posible establecer nueve combinaciones que forman la base de los casos de estudio a analizar. Dichas combinaciones se muestran en la Tabla 4.

3.4. Fase 4: Indexación de documentos de referencia

El objetivo de esta etapa ha sido indexar automáticamente los documentos de referencia (recuperados y pre-procesados en las etapas anteriores) y así, en la siguiente fase, obtener los documentos más relevantes asociados a una cadena de búsqueda. Para dicha indexación se ha utilizado *Lucene* [6], herramienta ampliamente utilizada para implementar estrategias de recuperación de información. Esta herramienta se destaca por simplicidad de uso y flexibilidad,

Tabla 4. Combinaciones de los tipos de documentos a analizar

Hilos / Docs Oracle	Sólo título (F_1)	Título + pregunta (F_2)	Texto completo (F_3)
Sólo el nombre de la clase (O_a)	O_aF_1	O_aF_2	O_aF_3
El nombre de la clase y el nombre de los métodos (O_b)	O_bF_1	O_bF_2	O_bF_3
Texto completo (O_c)	O_cF_1	O_cF_2	O_cF_3

Tabla 5. Lista de nombres de clase considerados *stopwords*

Clases consideradas stopwords									
Any	Byte	Class	Doc	Error	Exception	HTML	Key	Method	Object
Operation	Option	Package	Parameter	Point	Result	Set	Source	System	Text
Time	Type	Types	View	Void					

permitiendo indexar documentos y realizar búsquedas sobre ellos a través de una consulta (*query*), así como también definir el conjunto de palabras *stopwords*[3].

En nuestro caso de estudio, el conjunto de *stopwords* original de Lucene ha sido modificado para que no se consideren como tal las palabras reservadas del lenguaje Java (for, then, if, this) y se agregaron otras que han sido consideradas poco representativas en dicho contexto.

Además, durante la ejecución de la primera serie de casos (1 al 9) se ha detectado que algunos documentos Oracle aparecían relacionados a la mayoría de los hilos de discusión en el estudio, con un puntaje (*score*) alto. Por este motivo, se ha realizado un análisis de dichos documentos y se ha detectado que se trataba de nombres de clases que representaban vocablos de uso común en el lenguaje natural dentro del ambiente de programación. Por ejemplo, la palabra en inglés "*class*" es mencionada en diferentes contextos que no siempre se refieren a la clase Java *Class*. Algo similar ocurre con las clases *Parameter*, *Error*, etc. El listado de dichas clases, consideradas stopwords, se presenta en la Tabla 5.

Una vez detectada esta amenaza, se repitió la fase de indexación sin incluir los documentos de dichas clases en los documentos Oracle (casos 10 al 18). Al describir las características de los casos de estudio realizados (presentados en la Tabla 6), se han señalado dichas pruebas como "*filtrando clases stopwords*" o "*SW*".

3.5. Fase 5: Recuperación de documentos relevantes

Durante esta fase se ha utilizado la herramienta Lucene para el proceso de recuperación de información.

[3] Palabras que carecen de significado por sí solas y no brindan información acerca del contenido del texto

Tabla 6. Descripción de los casos de estudio realizados

#	Caso	Filtra SW	Descripción
1	O_aF_1	NO	Documentos Oracle conteniendo sólo el nombre de la clase y considerando sólo el título de los hilos, sin filtrar clases stopwords
2	O_aF_2	NO	Documentos Oracle conteniendo sólo el nombre de la clase y considerando título y pregunta principal de los hilos, sin filtrar clases stopwords
3	O_aF_3	NO	Documentos Oracle conteniendo sólo el nombre de la clase y considerando texto completo de los hilos, sin filtrar clases stopwords
4	O_bF_1	NO	Documentos Oracle conteniendo nombre de la clase y todos sus métodos, considerando sólo el título de los hilos, sin filtrar clases stopwords
5	O_bF_2	NO	Documentos Oracle conteniendo nombre de la clase y todos sus métodos, considerando título y pregunta principal de los hilos, sin filtrar clases stopwords
6	O_bF_3	NO	Documentos Oracle conteniendo nombre de la clase y todos sus métodos, considerando texto completo de los hilos, sin filtrar clases stopwords
7	O_cF_1	NO	Documentos Oracle completos, considerando sólo el título de los hilos, sin filtrar clases stopwords
8	O_cF_2	NO	Documentos Oracle completos, considerando título y pregunta principal de los hilos, sin filtrar clases stopwords
9	O_cF_3	NO	Documentos Oracle completos, considerando texto completo de los hilos, sin filtrar clases stopwords
10	O_aF_1SW	SI	Documentos Oracle conteniendo sólo el nombre de la clase y considerando sólo el título de los hilos, filtrando clases stopwords
11	O_aF_2SW	SI	Documentos Oracle conteniendo sólo el nombre de la clase y considerando título y pregunta principal de los hilos, filtrando clases stopwords
12	O_aF_3SW	SI	Documentos Oracle conteniendo sólo el nombre de la clase y considerando texto completo de los hilos, filtrando clases stopwords
13	O_bF_1SW	SI	Documentos Oracle conteniendo nombre de la clase y nombres de todos sus métodos, considerando sólo el título de los hilos, filtrando clases stopwords
14	O_bF_2SW	SI	Documentos Oracle conteniendo nombre de la clase y nombres de todos sus métodos, considerando título y pregunta principal de los hilos, filtrando clases stopwords
15	O_bF_3SW	SI	Documentos Oracle conteniendo nombre de la clase y nombres de todos sus métodos, considerando texto completo de los hilos, filtrando clases stopwords
16	O_cF_1SW	SI	Documentos Oracle completos, considerando sólo el título de los hilos, filtrando clases stopwords
17	O_cF_2SW	SI	Documentos Oracle completos, considerando título y pregunta principal de los hilos, filtrando clases stopwords
18	O_cF_3SW	SI	Documentos Oracle completos, considerando texto completo de los hilos, filtrando clases stopwords

Mediante el índice creado en la fase anterior, se ha realizado una serie de búsquedas utilizando el contenido de los hilos pre-procesados en la Fase 3. De esta manera, se ha obtenido un ranking de documentos Oracle relevantes para cada hilo, ordenados según la fórmula TF/IDF[7] que utiliza Lucene. En la sección siguiente se evaluará el ranking obtenido por Lucene para cada caso de estudio con respecto al ordenamiento sugerido por los expertos.

4. Evaluación

4.1. Medidas de evaluación utilizadas

Para poder evaluar y analizar los resultados obtenidos en cada uno de los casos de estudio, se ha elegido una serie de medidas utilizadas ampliamentes en técnicas de recuperación de información [8].

Sea D_{rlv} el conjunto de documentos relevantes en una búsqueda y $|D_{rlv}|$ el número de documentos de este conjunto. Asumiendo que la estrategia de recuperación, que está siendo evaluada, genera un conjunto de documentos respuestas D_{rcp}. Sea $|D_{rcp}|$ el número de documentos de este conjunto, se definen las siguientes medidas:

Precisión es la proporción de documentos recuperados (D_{rcp}) que son relevantes para la necesidad de información del usuario.

$$P = \frac{|D_{rlv} \bigcap D_{rcp}|}{|D_{rcp}|}$$

Recall también llamado "exhaustividad", es la fracción de documentos relevantes (D_{rlv}) recuperados para una consulta

$$R = \frac{|D_{rlv} \bigcap D_{rcp}|}{|D_{rlv}|}$$

Medida-F es una medida de evaluación que combina las medidas anteriores (*precisión* y *recall*) en un solo valor.

$$F = \frac{2 * P * R}{P + R}$$

Respecto a dichas medidas, la *precisión* proporciona información de cuantos documentos válidos (o relevantes) son recuperados, pero no permite saber cuántos documentos válidos no lo fueron. Por ejemplo, si hay 10 documentos relevantes y el sistema recupera uno de ellos, entonces se obtiene una precisión P=1 que representa el 100 %, ya que todos los documentos recuperados fueron válidos. Sin embargo, esta medida no brinda información sobre la cantidad de documentos válidos no recuperados (que en este caso son 9). Por otro lado, la medida *recall* informa la fracción de documentos relevantes que han sido recuperados sobre la totalidad de documentos recuperados. Para el ejemplo

anterior, la medida de recall es R=0,10 ya que se ha recuperado un solo documento relevante de los 10 esperados. Dado que el objetivo de los sistemas de recuperación de información es tratar de maximizar la cantidad de documentos recuperados que sean relevantes, la medida-F combina ambas medidas en una sola para poder evaluar los documentos recuperados. Para el caso del ejemplo anterior, se obtiene un valor $F = \frac{2*P*R}{P+R} = \frac{2*1*0,1}{1+0,1} = 0,18$; el cual brinda mayor información que considerar sólo la precisión. Esto no indica que la medida precisión sea poco útil, sino que se necesita de todas ellas para realizar una buena evaluación.

Un enfoque adicional es calcular estas medidas con valores de cortes (*cutoff*) sobre la cantidad de respuestas válidas devueltas por el recuperador de información utilizado. Es decir, en el caso de corte N se analizan cuantos documentos relevantes, no relevantes, precisión, recall y medida-F se obtienen al considerar solamente los primeros N documentos obtenidos en el proceso de recuperación.

4.2. Análisis de los resultados

La arquitectura y comportamiento de los casos de estudio realizados han sido descriptos en las secciones precedentes. A continuación se presentan los resultados obtenidos y se discute su *perfomance*.

Como se ha mencionado en la Sección 4.1, es importante evaluar los resultados uilizando valores de cortes sobre la cantidad de respuestas válidas. Para establecer dichos valores de corte se ha analizado la Tabla 3, donde se puede observar que la cantidad máxima de respuestas relevantes para un hilo es de 8 clases (hilo 6). Además, se ha calculado el promedio de respuestas válidas para cada hilo, obteniéndose un valor de 1.94, por lo que se ha tomado N=2 como corte inicial. Aunque tradicionalmente en la literatura relacionada a la recuperación de información se establecen los cortes siguiendo la escala 5, 10, 20, 30, 50, 100, etc., por el análisis expuesto anteriormente, se ha considerado apropiado reducir la escala y establecer los cortes (N) de 2, 3, 4, 5 y 10. Esto posibilita realizar la evaluación especialmente sobre los primeros valores retornados por Lucene. Esta decisión ha sido tomada debido a que las medidas de evaluación utilizadas no consideran el orden entre las respuestas devueltas.

Como se ha explicado anteriormente (Sección 3.4), se han llevado a cabo 18 casos de estudio, donde en los primeros 9 se ha indexado el total de los documentos Oracle recuperados y en los restantes casos (10 al 18) se ha filtrado el conjunto de clases consideradas *stopwords* (listadas en la Tabla 5).

La Tabla 7 muestra los resultados de la evaluación realizada a estos 18 casos considerando los cortes establecidos (2, 3, 4, 5 y 10) con respecto a la clasificación realizada por los expertos. Como se puede observar en esta tabla, para los casos 8, 17 y 18 se ha obtenido un valor F = 0 para el corte 2. Esto se debe a que no se ha recuperado ningún resultado válido para dicho corte, lo cual produce una

Tabla 7. Medidas obtenidas para los distintos casos de estudio

		2			3			4			5			10		
#	Caso	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
1	O_aF_1	0.32	0.39	0.35	0.41	0.61	0.49	0.43	0.66	0.52	0.43	0.68	0.53	0.44	0.68	0.54
2	O_aF_2	0.37	0.52	0.43	0.32	0.65	0.42	0.29	0.73	0.41	0.25	0.73	0.37	0.22	0.80	0.34
3	O_aF_3	0.50	0.69	0.56	0.39	0.75	0.51	0.33	0.82	0.47	0.31	0.93	0.46	0.18	0.95	0.30
4	O_bF_1	0.32	0.40	0.34	0.24	0.40	0.30	0.28	0.56	0.37	0.25	0.56	0.35	0.24	0.68	0.36
5	O_bF_2	0.24	0.36	0.29	0.19	0.44	0.27	0.16	0.46	0.24	0.17	0.54	0.26	0.11	0.80	0.18
6	O_bF_3	0.26	0.38	0.31	0.25	0.48	0.33	0.21	0.54	0.30	0.20	0.63	0.30	0.11	0.69	0.19
7	O_cF_1	0.08	0.11	0.09	0.07	0.14	0.09	0.05	0.14	0.08	0.04	0.14	0.06	0.03	0.19	0.05
8	O_cF_2	0.00	0.00	0.00	0.02	0.03	0.02	0.01	0.03	0.02	0.01	0.03	0.02	0.01	0.03	0.01
9	O_cF_3	0.03	0.03	0.03	0.04	0.05	0.04	0.03	0.03	0.04	0.02	0.05	0.03	0.02	0.08	0.03
10	O_aF_1*	**0.82**	**0.64**	**0.72**	**0.84**	**0.68**	**0.75**	**0.84**	**0.68**	**0.75**	**0.84**	**0.68**	**0.75**	**0.84**	**0.68**	**0.75**
11	O_aF_2*	0.47	0.59	0.53	0.47	0.70	0.56	0.45	0.73	0.55	0.43	0.73	0.54	0.42	0.75	0.54
12	O_aF_3*	0.53	0.70	0.60	0.46	0.83	0.59	0.37	0.84	0.51	0.34	0.87	0.49	0.25	0.90	0.40
13	O_bF_1*	0.42	0.45	0.44	0.49	0.51	0.44	0.43	0.56	0.48	0.43	0.56	0.48	0.43	0.68	0.53
14	O_bF_2*	0.24	0.36	0.29	0.19	0.44	0.27	0.18	0.49	0.27	0.16	0.52	0.24	0.10	0.67	0.18
15	O_bF_3*	0.29	0.40	0.34	0.25	0.48	0.33	0.21	0.54	0.30	0.21	0.64	0.32	0.11	0.69	0.19
16	O_cF_1*	0.08	0.11	0.09	0.07	0.14	0.09	0.05	0.14	0.08	0.04	0.14	0.06	0.03	0.19	0.05
17	O_cF_2*	0.00	0.00	0.00	0.02	0.03	0.02	0.01	0.03	0.02	0.01	0.03	0.02	0.01	0.03	0.01
18	O_cF_3*	0.00	0.00	0.00	0.02	0.03	0.02	0.03	0.05	0.04	0.02	0.05	0.03	0.01	0.05	0.02

precisión P=0, y esto se propaga en las fórmulas del recall y de la medida-F, haciendo que también se anulen. [4]

Si se observan gráficamente los promedios de las medida-F en la Figura4, el rendimiento de la recuperación mejora sustancialmente cuando se filtran las clases *stopword*. Esta mejora se produce en la mayoría de los casos y se debe a que las clases *stopword*, al presentar un alto grado de ambigüedad, producen el recupero de muchos documentos considerados irrelevantes. Además, se puede observar el crecimiento del rendimiento en los cuatro primeros casos del segundo grupo (casos 10 al 13) con respecto a la misma combinación de documentos sin filtrar dichas clases (casos 1 al 4).

Continuando con el análisis anterior, se observa que el caso 10 es el de mayor perfomance a partir del corte de 3 respuestas válidas; obteniéndose una precisión del 84 % y una medida-F del 75 %. Ampliando el análisis a los casos 11, 12 y 13 se puede apreciar que la perfomance tiende a disminuir a medida que se considera más información (ver descripción de los casos en la Tabla6).

Respecto a las hipótesis planteadas en la Sección 2 se puede apreciar que:

- HIPÓTESIS A: Utilizar mayor cantidad de información sobre cada una de las clases Java documentadas en Oracle permite clasificar los hilos de discusión relacionados a ellas de forma más precisa.

[4] Cabe aclarar que en estos 3 casos existen documentos relevantes como muestran los resultados de las evaluaciones para otros valores de corte.

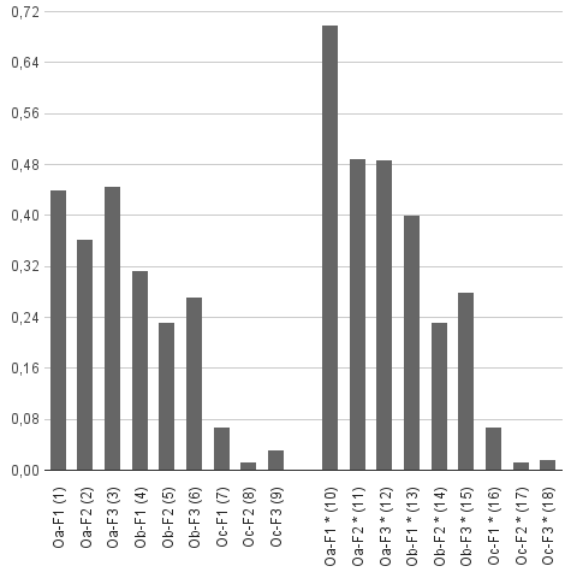

Figura 4. Comparación del promedio de la medida F para todos los casos de estudio

- Al contar con mayor información en los documentos Oracle (Casos O_c) se observa que la performance es considerablemente más baja que en los documentos que cuentan con menos información (Casos O_ay O_b) como puede observarse en la Figura 5(a). A su vez, la performance de los casos que consideran sólo el nombre de la clase (Casos O_a), es mayor que los que consideran más información en dicho tipo de documento (Casos O_b y O_c). Esto puede deberse a que el nombre de las clases relacionadas a la pregunta del hilo, suele estar mencionado directamente y con mayor frecuencia que los métodos u otra información que puede estar expresada en la descripción de cada clase.

- Hipótesis B: Utilizar más información sobre el problema explicado en los hilos de discusión permite clasificarlos de forma más precisa respecto a los documentos Oracle de las clases Java.
  - Aunque los casos que han considerado sólo el título del hilo (Casos F_1) son los que han obtenido la mejor performance en general, se puede apreciar que al utilizar la información del hilo completo (Casos F_3) mejora los resultados obtenidos en comparación a los casos que utilizan título y pregunta (Casos F_2), como puede observarse en la Figura 5(b). En este caso no se contradice completamente la hipótesis planteada, sino que es necesario replicar los casos de estudio incluyendo más hilos de discusión y de diferente tipo para rectificar o ratificar la validez de la misma.

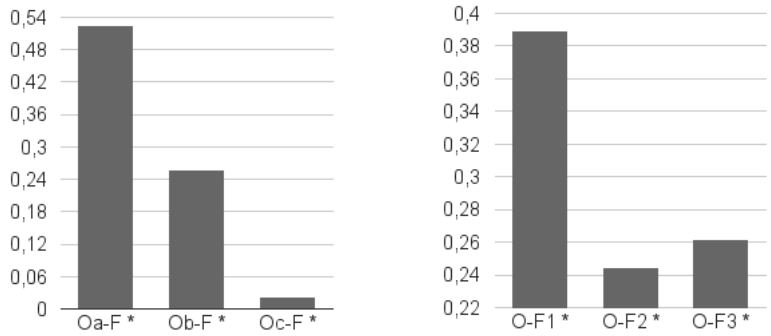

Figura 5. Análisis de performance según las hipótesis planteadas

Respecto a las amenazas a la validez del experimento, se debe mencionar que, los documentos Oracle utilizados para clasificar los hilos en estos casos de estudio se restringieron a la versión Java 1.5, por lo que podría producirse alguna diferencia en los resultados al usar documentos de otra versión del lenguaje (por ejemplo por la inclusión o eliminación de un método de una versión a otra). Otra amenaza detectada es haber centrado la experimentación sobre una clase de uso genérico, como *Integer*, que puede dar resultados más dispersos que al utilizar una clase de uso más especifico (por ejemplo *AbtractButton*). A futuro se planea incluir corpus de hilos más amplios y que se enfoquen en otros tipos de clases.

5. Trabajos relacionados

Existen varias propuestas de reúso de conocimiento en foros de discusión, como por ejemplo [9], que analiza automáticamente los hilos de un foro de discusión de un curso de Inteligencia Artificial y propone otros hilos con contenido similar a los anteriores, los cuales fueron realizados por estudiantes del mismo curso en dictados anteriores. Otra propuesta relacionada es la de Helic y Scerbakov [10], que clasifica los mensajes de un foro de discusión de acuerdo a una jerarquía de temas predefinida. Ambas propuestas están pensadas para un dominio de aprendizaje colaborativo, mientras que nuestro recomendador apunta a un contexto más amplio, involucrando usuarios con distinto conocimiento previo del tema (*background*). Finalmente, y más importante, en dichos trabajos el foro utilizado es único, por lo tanto se puede asegurar que la información a analizar se encuentra en un formato estándar, mientras que nuestra propuesta apunta a recolectar información de distintos foros, por lo tanto la heterogeneidad de formatos de la información a capturar es un desafío extra.

En cuanto a clasificación de texto contenido en foros de discusión utilizando herramientas de recuperación de información, podemos mencionar el trabajo de [11] que estudia la clasificación de mensajes recuperados de foros de la web de acuerdo a una jerarquía de temas obtenido de Wikipedia. En este caso, se

diferencia de nuestro trabajo al enfocarse a nivel de mensaje en lugar de hilos y al conjunto de documentos de referencia.

6. Conclusiones y trabajo futuro

En este trabajo se presenta una serie de casos de estudio realizados con el fin de analizar estrategias para clasificar hilos de conversaciones recuperadas de un foro de discusión técnico. En esta etapa se restringió el análisis a los hilos de un foro de discusión en particular (Stack Overflow) y se tomó un subconjunto de ellos relacionados a una cadena de búsqueda concreta, tomando como base la clase *Integer* de Java. La clasificación se realizó considerando la información contenida en distintas secciones de ambos tipos de documentos y también se consideró el filtrado de las clases Java cuyo nombre es de uso común en lenguaje natural.

Los resultados obtenidos en esta etapa indicarían que, al contrario de las hipótesis planteadas que suponían que a mayor información mayor la posibilidad de contar con una clasificación exitosa, sería más apropiado que la relación se restrinja al nombre de la clase y al título del hilo. Sin embargo, para confirmar el resultado de la hipótesis B, habría que ampliar los casos de prueba para verificar la tendencia observada de que los hilos completos mejoran la performance con respecto a los casos que consideraron el título y la pregunta principal, y cómo varía la performance en comparación a los que sólo consideran el título.

Dado que este resultado proviene de un conjunto de hilos restringido, nuestro trabajo a futuro se enfocará en replicarlo con mayor cantidad de hilos así como con otras técnicas aplicadas en la fase de la recuperación de información, para asegurar la generalidad de estos resultados. También se preve realizar nuevos casos de estudio centrados en otros factores que podrían incidir en la precisión de la clasificación, y que no tienen que ver con la "cantidad" de información, sino con la forma de manipularla.

Finalmente, en este trabajo las medidas utilizadas trabajan sobre un conjunto de documentos relacionados sin tener en cuenta su orden de relevancia. A futuro se planea ampliar el análisis a otros tipos de métricas que consideren el ranking de documentos relevantes retornado por Lucene.

Agradecimientos

Este trabajo está parcialmente soportado por el subproyecto "*Reuso de Conocimiento en Foros de Discusión Técnicos*", correspondiente al Programa de Investigación 04/F001 "*Desarrollo Orientado a Reuso*", y por el proyecto 04/F006 "*Agentes inteligentes en ambientes dinámicos*", de la Universidad Nacional del Comahue (Neuquén, Argentina).

Referencias

1. S. Gottipati, D. Lo, and J. Jiang, "Finding relevant answers in software forums," in *26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), Lawrence, KS, USA, November 6-10, 2011*, pp. 323–332, 2011.
2. Aranda, Gabriela, Martínez Carod, Nadina, Roger, Sandra, Faraci, Pamela, and Cechich, Alejandra, "Una herramienta para el análisis de hilos de discusión técnicos," in *CACIC 2014, XX Congreso Argentino de Ciencias de la Computación*, (San Justo, Argentina), pp. 803 – 812, Oct. 2014.
3. C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Computer Science, Springer, 2012.
4. M. V. Zelkowitz, D. R. Wallace, and D. W. Binkley, "Lecture notes on empirical software engineering," ch. Experimental Validation of New Software Technology, pp. 229–263, River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2003.
5. R. van Solingen and E. Berghout, *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill, 1999.
6. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.
7. T. Roelleke and J. Wang, "Tf-idf uncovered: A study of theories and probabilities," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, (New York, NY, USA), pp. 435–442, ACM, 2008.
8. R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.
9. W. Chen and R. Persen, "A recommender system for collaborative knowledge," in *2009 Conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, (Amsterdam, The Netherlands, The Netherlands), pp. 309–316, IOS Press, 2009.
10. D.Helic and N. Scerbakov, "Reusing discussion forums as learning resources in wbt systems," in *IASTED International Conference Computers and Advanced Technology in Education*, (Rhodes, Greece), pp. 223 – 228, 2003.
11. M. Nicoletti, S. Schiaffino, and D. Godoy, "Mining interests for user profiling in electronic conversations," *Expert Syst. Appl.*, vol. 40, pp. 638–645, Feb. 2013.