Recuperación de noticias públicas ordenadas temporalmente y por zonas utilizando funciones combinadas de Apache Solr

Luis Ignacio Aita[1], Damián Barry[1], Carlos Buckle[1], Claudio Delrieux[12]

LINVI, Departamento de Informática, Facultad de Ingeniería, UNPSJB, Puerto Madryn 9120, Argentina,
IMAGLAB, Departamento de Ingeniería Eléctrica y Computadoras, UNS, 8000 Bahía Blanca, Argentina,
I.ignacioaita@sursoftware.com.ar, II.damian_barry@unpata.edu.ar,
WWW home page:http://madryn.unp.edu.ar

Resumen El presente trabajo se enfoca en investigar y mejorar las técnicas de recuperación de información sobre una base de datos existente de noticias obtenidas en redes sociales y diarios en línea. Dicha información ya se encuentra clasificada e indexada por diversos atributos, entre los cuales vale destacar la pertenencia geográfica de la noticia, tanto en forma semántica (teniendo en cuenta localidad, partido, provincia, etc. en forma jerárquica) como geográfica (latitud, longitud). Este desarrollo propone comprobar que es posible mejorar la combinación de funciones de búsqueda y recuperación de la información almacenada en grandes volúmenes de información mediante la utilización índices existentes, principalmente en relación con la pertenencia geográfica y la antigüedad de la información buscada.

Keywords: Big Data, Boosting, Information Retrieval, Ecuaciones de Búsqueda, Apache Solr

1. Introducción

Este trabajo describe el desarrollo y resultados obtenidos en el proyecto *"Zonales"*[1], el cual consiste en un gestor de contenidos web (www.zonales.com) que brinda un conjunto de servicios de información reunidos en un mismo lugar que poseen un factor en común: la georreferenciación de la información y de servicios desde una perspectiva local para cada comunidad. Dentro del portal, cada usuario puede consultar y publicar contenidos relacionados con su localidad y zonas

[1] El proyecto se realizó en el marco de un convenio de colaboración entre la empresa Mediabit S.A., especializada en comunicación, asesoramiento e integración tecnológica orientada a los negocios en Internet (Internet Bussines Provider), y el equipo de vinculación y transferencia tecnológica de la Sede Puerto Madryn de la Universidad Nacional de la Patagonia San Juan Bosco, conformado con la finalidad de mejorar la calidad tecnológica de las organizaciones destinatarias, brindando cursos y realizando convenios con empresas y proyectos de extensión con empresas.

de interés, generando información desde la periferia hacia el centro. El desarrollo permitió que la información clasificada por zona y por fuente se recupere utilizando una combinación de funciones espec cuyos resultados se presentan en el presente trabajo.

La motivación principal de este trabajo consiste en la necesidad de diseñar, investigar y probar diversas técnicas para combinar funciones de recuperación y filtros de información, para que los datos recuperados sean de interés para el usuario en el marco geográfico-temporal, teniendo en cuenta la gran cantidad de información mayormente irrelevante que se puede generar en la base de datos. Dichas técnicas permitirán establecer las capacidades necesarias con las que debería contar una base de datos de información masiva, tanto desde la perspectiva de almacenamiento y técnicas de indexación, como de distribución de las consultas, escalabilidad y rendimiento en ambientes heterogéneos.

2. Componentes del proyecto Zonales

2.1. Motor de extracción de información

Para la extracción de contenido público de Internet se desarrollo un motor de extracción de contenido público de Internet denominado *"ZCrawler"*[4]. El mismo se compone de un lenguaje de extracción que le permite a los usuarios mediante una especificación formal comenzar a extraer información siguiendo múltiples criterios y permitiendo la clasificación de esta información según a quien se está evaluando. Para ello se desarrollaron servicios de extracción para las redes sociales Facebook y Twitter y para medios digitales que contaran con RSS estándar. Además se han adaptado algunos parsers específicos para blogs y diarios en línea que no contaban con una implementación estándar de RSS. En particular se implementaron conectores para los diarios Jornada y Chubut, ambos de la provincia de Chubut. Adicionalmente para los grandes medios nacionales se adaptaron extractores ya que sus definiciones de RSS no accedían ni a los comentarios ni a las referencias (links) externas ni a las imágenes.

A partir de poder extraer imágenes de las noticias y comentarios se mejoró la herramienta permitiendo realizar un tratamiento común a todo el material multimedia asociado a una publicación realizando una gestión uniforme tanto para Facebook, Twitter y las páginas sindicadas. La última versión generada de Zonales estuvo en linea aproximadamente seis meses en forma totalmente funcional. El volumen de datos que se generó durante ese tiempo fue de aproximadamente un millón y medio de documentos. El importante volumen de noticias extraídas (aproximadamente 8.500 noticias diarias y clasificadas de acuerdo a su pertenecía geográfica, ya sea por su emisor o por auto-clasificación según criterios de contenido) provocó el desafío de recuperar la información almacenada de forma efectiva y eficaz. Esto es especialmente importante, debido a que se estaban extrayendo noticias e información de solamente 15 municipios de los 2112 que existen en la República Argentina, lo que demandaba la construcción de una cuidadosa ecuación que garantizara los requerimientos de pertenencia geográfica

y ordenamiento temporal, además de las búsquedas tradicionales por términos que se desearan realizar.

Proceso de Extracción El proceso de extracción de la información para almacenar en el índice se divide en dos etapas claramente diferenciadas: configuración y extracción.

Como se puede apreciar el proceso consta de 3 fases bien definidas:

1. Definición de ecuaciones de extracción.
2. Ejecución de las extracciones de acuerdo a sus parámetros de planificación.
3. Clasificación e Indexación de las noticias extraídas. A su vez esta fase se divide en las siguientes tareas:
   a) Extraer de la noticia por fuente.
   b) Expandir la información no provista por la fuente.
   c) Clasificar la información.
   d) Indexar y almacenar la información en Apache Solr.

2.2. Recuperación de la Información

El desafío planteado de extracción de múltiples fuentes de información impone otro desafío que el proyecto debía resolver: la escalabilidad, debido al gran procesamiento de información extraída. En resultados previos (ver[6]) se pudo arribar a un conjunto de conclusiones relevantes en este trabajo, entre las cuales mencionamos las siguientes:

- La búsqueda secuencial de cualquier tipo de información presenta varios problemas, siendo el principal la falta de escalabilidad. Una solución a este inconveniente es el uso de estructuras de datos que permitan ser rápidamente consultadas.
- El indexado transforma los datos desde su forma original en una estructura que facilita la búsqueda y recuperación de los mismos en forma rápida y precisa, por ejemplo un índice invertido[5], un índice de citas, una matriz o un árbol.
- El proceso de indexado generalmente requiere un análisis y procesamiento de los documentos a incluir en el índice: lematización, tokenización, análisis fonético, etc. Estos pasos introducen problemas y desafíos importantes al momento del procesamiento[5], que no son alcance del presente trabajo.
- Dentro del estudio del presente trabajo, debido a la facilidad de implementación en un ambiente heterogéneo, se llegó a la conclusión que una solución óptima requeriría el uso del concepto de base de datos de partición horizontal o sharding [2].
- Apache Solr[7], que utiliza una base de datos no convencional para almacenar el índice de búsqueda (listas invertidas), provee múltiples capacidades para el escalado horizontal, permitiendo dividir la carga de trabajo entre múltiples instancias. Esto permite una fragmentación horizontal de los mismos entre múltiples servidores Apache Solr, a los cuales se les denomina shards[5]. Las

búsquedas son luego redirigidas a cada shard, y finalmente una respuesta única es construida en base a los resultados obtenidos de cada uno. Esta técnica es utilizada especialmente cuando se cuenta con un gran volumen de datos sobre el cual realizar consultas."

Adicionalmente al desafío de escalabilidad y rendimiento de la recuperación de información en grandes volúmenes de datos sobre Apache Solr, se platea la necesidad de filtrado por criterios y parámetros de búsqueda como de boosting (ordenamiento) de la misma.

Para resolver este último desafío el mismo se encara en 2 etapas bien definidas:

1. Diseño de filtros y boosting (ordenamiento) en un entorno de pruebas.
2. Implementación en un entorno real sobre la información extraída (gran volumen de datos)

3. Diseño e implementación de las Ecuaciones de Búsqueda

El alcance del entorno de pruebas para el presente trabajo está limitado al índice Solr construido durante el proyecto Zonales.

3.1. Diseño

Al decidir la utilización de la herramienta Apache Solr[7] para el indexado y la búsqueda de información se tomaron dos decisiones de arquitectura clave que afectaban al resto de los componentes de la solución:

- El diseño del esquema de documentos que se utilizaría para indexar la información.
- La configuración de la instancia Solr y el diseño de las consultas para recuperar la información buscada.

En el proyecto Zonales, el diseño del índice y la configuración de la instancia Solr para almacenar la información se realizó en dos etapas claramente diferenciadas por la arquitectura pensada para la solución en cada una de ellas.

En ambas etapas la unidad de información o documento para Zonales era el post. Un post es un artículo re-publicado en Zonales, proveniente de una red social como Facebook o Twitter, de un medio digital o de cualquier otra fuente que se indexaba y clasificaba para luego mostrarlo "zonificado" y ecualizado[3] según los intereses del usuario en la web. En la segunda etapa se comenzó a considerar los comentarios de los usuarios sobre las publicaciones extraídas (crawleadas) y también se indexaron como posts vinculados con otro post preexistente en el índice.

Indexación de documentos (posts): La extracción de información heterogénea de diversas fuentes y el proceso de normalización y homogenización de la información para ajustarla a una estructura única fue el mayor desafío. El caso más emblemático fue el de la extracción de información a partir de fuentes RSS, donde gran parte de la misma no respetaba los estándares para comentarios, enlaces (links) y contenido multimedia. Para poder lograrlo se desarrolló una herramienta de configuración de selectos CSS que, en conjunto con una serie de funciones de parsing incorporadas al extractor correspondiente, permitieron normalizar esta información. Este esfuerzo de homogenización de la información derivó en la creación de dos formatos normalizados para estructurar las noticias que permitieron contener tanto los posts extraídos de las diversas fuentes como los datos de clasificación asociados. Se decidió utilizar los formatos estándares XML y JSON para contener la estructura de datos diseñada y se la bautizó como XZone y JZone respectivamente.

3.2. Recuperación de posts

Partiendo de la premisa de mostrar al usuario la información ordenada en base a su contexto geográfico y sus preferencias, se diseñó el portal web considerando cuatro aspectos principales:

- **Ecualización**[3]: Se le permitía a los usuarios del portal definir un set de preferencias para el filtrado de la información.
- **Relevancia**: se le permitía a los lectores de zonales elevar o disminuir la relevancia de una noticia desde la propia web. Se utilizó además como parámetros la cantidad de "me gusta", "retweets" y "comentarios" de cada post para calcular la relevancia en base a un fórmula de peso para cada caso.
- **Tipo de fuente**: los lectores del portal pueden seleccionar el tipo de fuente de la cual recuperar las noticias. En principio se dividió la recuperación en dos conceptos: "En la red" para englobar las noticias de las redes sociales Facebook y Twitter, y "Noticias" para la información recuperada desde medios digitales a través de RSS.
- **Temporalidad de la noticia**: considerando que para el común de los los lectores las noticias pierden relevancia con el paso del tiempo este es un aspecto de peso en la recuperación de la información.
- **Zona**: el usuario debía ver principalmente noticias de su zona y aledaños.

Las pruebas narradas en la siguiente sección se realizaron haciendo foco en la temporalidad y la localización de la información, aspectos por los cuales se trabajó con las funciones de boosting de Solr y que son objeto del presente trabajo.

4. Pruebas

La búsqueda de la consulta Solr que permitiera recuperar la información teniendo en cuenta los aspectos de temporalidad y zona se realizó a través de

métodos de prueba sucesivas, refinando las consultas en cada iteración, analizando en cada uno de ellas los resultados obtenidos en comparación con los esperados y volviendo a ajustar los parámetros. Se utilizaron para tal fin boosting querys (bq) y boosting functions (bf), parámetros que permitieron especificar un factor que aumentara o "impulsara" la importancia de determinados campos, valores o combinación de ambos en la consulta.

Debido a la problemática planteada se decidió la utilización de técnicas de TDD (Test Driven Development o Desarrollo Guiado por Pruebas)[1] como opción válida para las pruebas de recuperación y boosting realizadas durante el proceso, brindando una opción repetible que permitiera garantizar el cumplimiento del objetivo de las pruebas a alcanzar para los distintos escenarios y sus variantes. Se diseñaron distintos escenarios de prueba, cada uno de los cuales suponía la ejecución de cambios en el contexto por parte del usuario, planteando los resultados esperados en relación al orden de las noticias, abarcando las combinaciones que se consideraron de mayor importancia.

Es importante considerar el razonamiento seguido para el planteo de los resultados esperados cuando la consulta involucraba estos aspectos. Es decir, cuando un usuario deseaba ver las noticias de su zona ordenadas desde la más reciente a la más antigua.

Partiendo de la suposición de la existencia de un conjunto de datos que recibe noticias extraídas de diversas fuentes en forma constante, tanto para la zona del usuario, por ejemplo su localidad, como para el resto de los elementos de la jerarquía geográfica (barrios, provincia, país, etc.), tanto ascendente como descendente, se consideró que, para un usuario de una determinada localidad, serían de mayor relevancia las noticias de su provincia dentro de las ultimas cuarenta y ocho horas que las de su localidad con mayor antigüedad, por ejemplo de cinco días.

Siguiendo este razonamiento se dividió la temporalidad en cuatro franjas que, combinadas con las jerarquías geográficas, definían el siguiente orden esperado en los resultados:

1. Noticias de la zona seleccionada dentro de las 48 horas
2. Noticias de los hijos de la zona seleccionada dentro de las 48 horas
3. Noticias de los padres de la zona seleccionada dentro de las 48 horas
4. Noticias hermanas de la zona seleccionada dentro de las 48 horas
5. Noticias de la zona seleccionada desde 48 horas hasta 1 semana
6. Noticias de los hijos de la zona seleccionada desde 48 horas hasta 1 semana
7. Noticias de los padres de la zona seleccionada desde 48 horas hasta 1 semana
8. Noticias hermanas de la zona seleccionada desde 48 horas hasta 1 semana
9. Noticias de la zona seleccionada desde 1 semana hasta 1 mes
10. Noticias de los hijos de la zona seleccionada desde 1 semana hasta 1 mes
11. Noticias de los padres de la zona seleccionada desde 1 semana hasta 1 mes
12. Noticias hermanas de la zona seleccionada desde 1 semana hasta 1 mes
13. Resto de las noticias respetando la temporalidad.

4.1. Escenarios

Los distintos escenarios de prueba se construyeron combinando conjuntos de datos y criterios de búsqueda donde para cada combinación se especificaron los resultados esperados.

Para facilitar la creación de los conjuntos de datos, la sistematización de las pruebas y el análisis de lo resultados, se utilizaron valores descriptivos de la noticia dentro de la prueba en el campo *verbatim*. Por ejemplo, para una noticia extraída de Facebook con una antigüedad de tres días, una relevancia con valor 20 para la zona Puerto Madryn en el campo verbatim se grabó la siguiente información: "Noticia 3 días - Puerto Madryn, Chubut, Argentina – Facebook – R20".

El primer conjunto de datos D1 se compone de treinta y cinco noticias, organizadas en sub-conjuntos de cinco (5) noticias de la siguiente forma:

- Sub-conjunto de noticias de barrio Centro, localidad Puerto Madryn, Chubut
- Sub-conjunto de noticias de localidad Puerto Madryn, Chubut
- Sub-conjunto de noticias de la provincia de Chubut
- Sub-conjunto de noticias de otra localidad de la provincia (hermana): Trelew, Chubut
- Sub-conjunto de noticias de una localidad de otra provincia: Rosario, Santa Fe
- Sub-conjunto de noticias de otra provincia: Mendoza
- Sub-conjunto de noticias a nivel país: Argentina

Cada uno de estos sub-conjuntos se compone con noticias de distinto valor temporal:

- Una noticia de hoy
- Una noticia de ayer
- Una noticia con 3 días de antigüedad
- Una noticia con 5 días de antigüedad
- Una noticia con 20 días de antigüedad

Para definir los criterios de búsqueda se abarcaron distintos niveles de jerarquía geográfica, considerando como "Nivel 1" a la zona de mayor escala, es decir Argentina. Se utilizaron los siguientes criterios:

- C1-Recuperar las noticias de Puerto Madryn, Chubut, Argentina – Nivel 3
- C2-Recuperar las noticias del barrio Centro, Puerto Madryn, Chubut, Argentina – Nivel 4
- C3-Recuperar las noticias de Trelew, Chubut, Argentina – Nivel 3
- C4-Recuperar las noticias de la provincia de Chubut, Argentina – Nivel 2

El segundo conjunto de datos D2 contenía cinco noticias adicionales, que tienen por objeto principalmente simular la dinámica de ingreso de nueva información. Es decir, la extracción de noticias en Zonales se realiza en forma constante de acuerdo a la programación del scheduler.

Los datos adicionales agregados son:

- Una noticia de Puerto Madryn, Chubut de ahora (más reciente)
- Una noticia de Puerto Madryn, Chubut de 3 días de antigüedad (más de 48 horas)
- Una noticia de Sarmiento, un barrio de Trelew, Chubut de ahora (nivel 4 en localidad hermana, temporalidad más reciente)
- Una noticia de Comodoro Rivadavia de ahora (localidad hermana, temporalidad más reciente)
- Una noticia de Formosa, Formosa de ahora (localidad de otra provincia, temporalidad más reciente)

La combinación de ambos conjuntos de datos con los criterios definidos dieron origen a un total de ocho escenarios donde reflejamos alguno de ellos en la siguiente tabla:

Escenario	**Resultado**	**Orden esperado**
E1	RE1	Se espera recuperar en primer lugar las noticias en el rango de 48 hs. para Puerto Madryn, luego las del barrio Centro de Puerto Madryn, Chubut, Trelew y finalmente las de Argentina y las de localidades de otras provincias. A continuación las noticias del rango 48hs. a 1 semana y 1 semana a 1 mes, ambos en el mismo orden de zona descripto anteriormente.
E2	RE2	Se espera recuperar en primer lugar las noticias en el rango de 48 hs. para barrio Centro de Puerto Madryn, luego las de Puerto Madryn, Chubut, Trelew y finalmente las de Argentina y las de localidades de otras provincias. A continuación las noticias del rango 48hs. a 1 semana y 1 semana a 1 mes, ambos en el mismo orden de zona descripto anteriormente.
...	...	...
E8	RE8	El orden esperado es el mismo del escenario E4 con las siguientes alteraciones: Dentro del rango de 48 horas se espera recuperar las noticias más recientes de Puerto Madryn, barrio Sarmiento de Trelew y Comodoro Rivadavia a continuación de las noticias de Chubut y la de Formosa al final del rango. La noticia insertada para Puerto Madryn de más de 48 horas debe aparecer a continuación de las noticias de Chubut dentro del rango correspondiente.

Cuadro 1: Resultados esperados de cada escenario

4.2. Sistematización de las pruebas

Las pruebas se realizaron ejecutando las consultas en forma directa sobre la herramienta Solr a través de request HTTP sobre su API pública y solicitando los resultados en formato XML.

Se estructuraron los resultados esperados respetando este formato y en cada ciclo de prueba y refinamiento se automatizó la comparación de resultados utilizando herramientas de testing mediante asertos[1].

5. Resultados

Luego de reiteradas modificaciones y adaptaciones sobre las funciones de boosting (ordenamiento) para lograr obtener los resultados esperados en cada escenario se logró una función estable, probada mediante la regeneración de los datos y la repetición de las consultas. Una vez estabilizada la función objetivo se repitieron las consultas veinte veces, logrando obtener siempre los mismos resultados para cada escenario. La efectividad lograda en función de los resultados obtenidos versus resultados esperados es cercana un 90 %, aunque la misma aumenta hasta un valor cercano al 100 % si se consideran solo la primer franja temporal, es decir las noticias dentro de las 48 horas.

Para la obtención de la función de boosting se creó un servicio javascript que, en función del contexto, construye la URL Solr. En particular la porción de la URL relacionada con las pruebas y las funciones de boosting se construyó utilizando la función getSolrBoosting(zCtx) que se explica a continuación. Todas las dimensiones definidas fueron divididas en escala en función de la cantidad de niveles jerárquicos de la zona geográfica. Es decir, Nivel 1: Argentina, Nivel 2: Provincias, Nivel 3: localidades, Nivel 4: barrios, y así sucesivamente. El campo selZone del contexto que contiene la zona en formato extendido permite calcular este nivel. Para el peso de las zonas en las consultas se utilizó boosting querys (bq) construidas alternando los campos zonePartialExtendedString y zoneExtendedString con el siguiente criterio de boosting:

- zonePartialExtendedString = (zona de nivel 1)$\hat{1}0^{3}$
- zoneExtendedString = (zona de nivel 1)$\hat{1}0^{4}$
- zonePartialExtendedString = (zona de nivel 2)$\hat{1}0^{5}$
- zoneExtendedString = (zona de nivel 2)$\hat{1}0^{6}$
- ...
- zonePartialExtendedString = (zona de nivel n)$\hat{1}0^{n+3}$
- zoneExtendedString=(zona de nivel n)$\hat{1}0^{n+4}$

Esta distribución de peso permite dar mayor importancia a las zonas más cercanas a las del nivel buscado en una relación 10^{n}. Las franjas temporales cobran una relevancia mayor que la zona en relación directa al nivel. Por ejemplo, en una búsqueda de zona nivel 4 el peso de la franja temporal debe ser mucho mayor que para un búsqueda de una zona nivel 2. Para el peso de las franjas temporales también se utilizó boosting querys (bq) construidas en base a campo modified de la noticia y la cantidad de niveles utilizando el siguiente criterio:

- modified:[NOW-48HOURS TO *$\hat{1}0^{\#\text{nivel}*4}$
- modified:[NOW-7DAYS TO NOW-48HOURS]$\hat{1}0^{\#\text{nivel}*(4\text{-}5)}$
- modified:[NOW-30DAYS TO NOW-7DAYS]$\hat{1}0^{\#\text{nivel}*(4\text{-}10)}$

El factor de corrimiento utilizado en el multiplicador del exponente, definido en una escala 1, 5, 10, es el que permitió aumentar o disminuir el peso de la franja temporal en función de los niveles de la zona. Otro requisito utilizado en las consulta fue recuperar en orden de fecha las noticias dentro de una misma

franja temporal. Para lograrlo se utilizó una boosting function aplicada sobre el campo modified y utilizando el nivel para el factor de boosting de la siguiente manera: bf=ord(modified)$\hat{1}0^{\#\text{nivel}*2}$. La consultas obtenidas se construyen en función de los criterios. A continuación se detalla la query resultante para cada uno de ellos.

5.1. Ejemplos de algunos criterios

Criterio 2, Escenarios 2 y 6 *Centro, Puerto Madryn, Chubut, Argentina*

```
/solr/select?indent=on&version=2.2&start=0&fl=*%2Cscore&rows=40&qt=zonalesConten
&sort=&wt=xml&explainOther=&hl.fl=&q=docType:post&bf=ord(modified)^100000000
&bq=+
        zoneExtendedString:"Centro,+ Puerto+Madryn,+Chubut,+Argentina"^10000000000
        zonePartialExtendedString:"Centro,+ Puerto+Madryn,+ Chubut,+Argentina"^100
        zoneExtendedString:"Puerto+Madryn,+ Chubut,+Argentina"^100000000+
         zonePartialExtendedString:"Puerto+Madryn,+ Chubut,+Argentina"^10000000+
         zoneExtendedString:"Chubut,+   Argentina"^1000000+
        zonePartialExtendedString:"Chubut,+Argentina"^100000+
        zoneExtendedString:"Argentina"^10000+
        zonePartialExtendedString:"Argentina"^1000+
        modified:[NOW-48HOURS+TO+*]^10000000000000000+
        modified:[NOW-7DAYS+TO+NOW-48HOURS]^100000000000+
        modified:[NOW-30DAYS+TO+NOW-7DAYS]^1000000
```

Criterio 4, Escenarios 4 y 8 *Chubut, Argentina*

```
/solr/select?indent=on&version=2.2&start=0&fl=*%2Cscore&rows=40&qt=zonalesConten
    zoneExtendedString:"Chubut,+Argentina"^1000000+
    zonePartialExtendedString:"Chubut,+Argentina"^100000+
    zoneExtendedString:"Argentina"^10000+
    zonePartialExtendedString:"Argentina"^1000+
    modified:[NOW-48HOURS+TO+*]^100000000+
    modified:[NOW-7DAYS+TO+NOW-48HOURS]^1000+
    modified:[NOW-30DAYS+TO+NOW-7DAYS]^0.01
```

No se hicieron pruebas de estrés sobre este conjunto de datos pues las pruebas de escalabilidad se realizaron en el trabajo ya mencionado "Distributed Search on Large NoSQL Databases" [6].

6. Conclusiones y Trabajo Futuro

Las funciones y consultas de boosting utilizadas en el presente trabajo son solo una pequeña parte de un gran conjunto de herramientas que ofrece Apache Solr para la recuperación ordenada de información. El campo de Big Data es tan amplio que incluso muchos expertos consideran a esta época como "la era del

Big Data" y en este contexto dominar una herramienta como Solr que permita indexar, clasificar y recuperar en forma ordenada gran cantidad de información valiosa es un punto fuerte de la herramienta. El resultado principal de este trabajo consistió en poder implementar los requerimientos propuestos en un entorno de producción real, con datos reales de la empresa Mediabit, para su proyecto Zonales. Si bien la aproximación ha sido empírica a las funciones de boosting utilizadas, las mismas han cumplido satisfactoriamente las premisas requeridas.

Como trabajo futuro podemos mencionar que las nuevas versiones de Apache Solr ofrecen la posibilidad de indexar datos geográficos asociados a la información e incorporar nuevas herramientas de búsquedas para recuperar dicha información por conceptos tales como distancias, pertenencia a una polígono geográfico, etc. Esto permitiría mejorar la solución Zonales a tal punto que un usuario podría recuperar las noticias ordenadas por cercanía geográfica a partir del punto donde se encuentre con niveles de precisión de metros.

En [6] se indexaron artículos de wikipedia utilizando como criterio de particionamiento horizontal de los datos la paridad del id numérico de los documentos. Repetir la experiencia utilizando la información de Zonales y tomando como método de partición de los datos el campo de zona geográfica en su formato semántico es otro desafío de interés y permitiría incluso probar la escalabilidad utilizando las funciones de boosting del presente trabajo. Otro desafío es mejorar el modelo matemático para refinar los resultados del presente trabajo. Se considera que las relaciones de pesos encontradas tanto para la zona como para la temporalidad pueden mejorarse a través de este tipo de modelos.

Referencias

1. Kent Beck. *Test-driven development: by example.* Addison-Wesley Professional, 2003.
2. Henderson Cal. *Building Scalable Web Sites.* 2006.
3. Barry Damián, Juan Manuel Cortez, and Francisco Páez. Construcción de un ecualizador de interés mediante el uso de lucene-solr. *IEEE Intercon*, 2010.
4. Barry Damián, Aita Luis Ignacio, and Cortez Juan Manuel. Zcrawler: Extracción, clasificación y publicación de información pública desde su perspectiva geográfica. *JAIIO 2014.*
5. Hatcher Erik and Otis Gospodnetić. *Lucene in Action.* Manning Publications Co., 2nd. ed edition, 2004.
6. Tinetti Fernando, Damián Barry, Ignacio Aita, and Franciasco Páez. Distributed search on large nosql databases. *PDPTA2011*, 2011.
7. Apache Fundation. Apache solr reference guide.