RI con *n*-gramas: tolerancia a errores y multilingüismo*

Jesús Vilares, Miguel A. Alonso, Carlos Gómez-Rodríguez, Jorge Graña

Departamento de Computación, Universidade da Coruña
Campus de Elviña s/n, 15071 – A Coruña
{jvilares,alonso,cgomezr,grana}@udc.es

Resumen En este artículo presentamos el trabajo que en el Grupo LYS (Lengua y Sociedad de la Información) hemos venido desarrollando en fechas recientes en las áreas de recuperación de información tolerante a errores y recuperación de información multilingüe. El nexo común entre ambas líneas de investigación es el empleo de *n*-gramas de caracteres como unidad de procesamiento, en detrimento de soluciones más convencionales basadas en palabras o frases. El empleo de *n*-gramas nos permite beneficiarnos de su simplicidad, independencia del idioma y robustez. En el caso de recuperación tolerante a errores presentamos los resultados de un estudio que hemos realizado acerca de la robustez de nuestra solución frente a errores ortográficos y de escritura presentes en la consulta, así como la metodología que hemos diseñado para generar conjuntos de test empleando errores humanos reales. Finalmente, en el caso de la recuperación multilingüe, presentamos nuestro algoritmo de alineamiento a nivel de *n*-gramas para corpus paralelos, base de nuestra propuesta, y los resultados obtenidos en nuestros experimentos.

Key words: *n*-Gramas de caracteres; Recuperación de Información Tolerante a Errores; Recuperación de Información Multilingüe

1. Introducción

Formalmente, un *n*-grama de caracteres es una secuencia de caracteres dentro de una palabra [16]. Podemos, por ejemplo, dividir la palabra `"patata"` en cuatro 3-gramas superpuestos: `pat`, `ata`, `tat` y `ata`. La utilización de *n*-gramas tiene como principales ventajas su simplicidad, eficiencia, robustez, completitud e independencia del dominio. Estas ventajas no han pasado desapercibidas para la comunidad investigadora en Recuperación de Información (RI) y, por ejemplo, a día de hoy el empleo términos índice basados en *n*-gramas en entornos de RI es una práctica relativamente usual [10,16].

* Este trabajo ha sido parcialmente subvencionado por el Ministerio de Educación y Ciencia y FEDER (a través del proyecto HUM2007-66607-C04-03) y por la Xunta de Galicia (a través de los proyectos INCITE08-PXIB302179PR y PGIDIT07-SIN005206PR; y a través de la *"Red Gallega de Procesamiento del Lenguaje y Recuperación de Información"* y la *"Red Gallega de Lingüística de Corpus"*).

Mientras que los sistemas clásicos de RI suelen integrar conocimiento lingüístico y recursos tales como listas de *stopwords*, *stemmers*, lexicones, tesauros, etiquetadores [10], etc., la *tokenización* en n-gramas no precisa de ningún tipo de procesamiento o conocimiento más allá del propio texto: tanto consultas como documentos son simplemente *tokenizados* en n-gramas superpuestos en lugar de en palabras [11], para luego ser procesados como cualquier otro término por el motor de recuperación. Por la misma razón, esta aproximación es independiente del idioma y dominio, ya que no integra ningún tipo de recurso o conocimiento concreto sobre ellos, bien al contrario, el empleo de correspondencias a nivel de n-gramas constituye en sí mismo un mecanismo de normalización de términos que permite además trabajar con una gran diversidad de lenguas sin procesamiento alguno a mayores [11,16,10]. Finalmente está su propia robustez, consecuencia de la redundancia introducida por el propio proceso de *tokenización*.

Este trabajo presenta dos líneas de investigación en las que nuestro grupo, el Grupo LYS (Lengua y Sociedad de la Información)[1] de la Universidade da Coruña, ha venido trabajando recientemente. Ambas están basadas en el empleo de n-gramas de caracteres como unidad de procesamiento en dos contextos diferentes: la Recuperación de Información Tolerante a Errores (concretamente errores presentes en la consulta) y la Recuperación de Información Multilingüe.

El resto del artículo se estructura como sigue. En primer lugar presentamos en la Sección 2 nuestra propuesta para recuperación tolerante a errores. El diseño del sistema y de los experimentos realizados a tal efecto se describen en la Sección 3, analizando a continuación en la Sección 4 los resultados obtenidos. Nuestro trabajo en recuperación multilingüe se describe en la Sección 5, analizando seguidamente los resultados de nuestros experimentos en la Sección 6. Finalmente, presentamos nuestras conclusiones y trabajo futuro en la Sección 7.

2. Aplicación a tolerancia a errores

Si bien los modelos formales de RI están diseñados inicialmente para operar sobre textos sin errores, deberían ser capaces también de operar sobre textos con *errores ortográficos*[2], los cuales pueden reducir el rendimiento del sistema [6]. Actualmente, debido a ello y al fenómeno de globalización de la red, existe un creciente interés en la *Recuperación de Información Tolerante a Errores (RITE)*. Básicamente, se trata de sustituir los mecanismos de establecimiento de correspondencia exacta por otros más flexibles que permitan correspondencias aproximadas. En este contexto, el estado del arte recoge dos aproximaciones genéricas al problema [8]. La primera contempla la palabra como unidad de trabajo; la segunda, sin embargo, opera a nivel de subpalabra.

[1] http://www.grupolys.org

[2] Entendiendo como tales aquellos errores resultado del desconocimiento de la ortografía, errores tipográficos durante la escritura y errores derivados de la presencia de ruido en el proceso de generación (ej. OCR).

En nuestro caso estudiaremos el uso de n-gramas como términos índice para poder así aprovecharnos de su robustez al operar sobre consultas con errores. Como ya hemos comentado, el texto (tanto consultas como documentos) es meramente *tokenizado* en n-gramas superpuestos antes de enviárselo al sistema de recuperación de información. De esta forma, cualquier error ortográfico presente afectará únicamente a parte de ellos, por lo que los n-gramas restantes de esa palabra harán posible el establecimiento de correspondencias. De este modo el sistema estará mejor dotado para trabajar en ambientes con ruido, siendo no sólo capaz de hacer frente a errores ortográficos, sino también a palabras desconocidas, e incluso variantes morfológicas o de escritura, en contraposición a otros métodos clásicos de normalización como el *stemming*, la lematización o el análisis morfológico, que se ven todos ellos afectados negativamente por estos fenómenos.

3. Diseño de experimentos sobre tolerancia a errores

3.1. Marco de evaluación

Los trabajos previos en este área se centran en el inglés [6,2], un idioma de estructura léxica relativamente simple. En nuestro caso hemos optado por el español, una lengua mucho más compleja desde el punto de vista morfológico [17]. Para la realización de experimentos en este ámbito disponemos del corpus para español de la *robust task* del CLEF 2006 [12], formada por 454.045 teletipos de noticias (1,06 GB) de la agencia EFE[3]. En concreto, el conjunto de prueba está formado por los 60 *topics* del denominado *conjunto de entrenamiento* proporcionado para dicha *task*[4]. Éstos constan de tres campos: *título*, un breve título como su nombre indica; *descripción*, una somera frase de descripción; y *narrativa*, un pequeño texto especificando los criterios de relevancia. Siguiendo la política del CLEF, nuestros experimentos se han realizado empleando únicamente los campos *título* y *descripción*.

Asimismo, en todos los experimentos de este trabajo se ha empleado la plataforma de código abierto Terrier [15] como motor de recuperación, usando un modelo de ordenación InL2[5].

Como se ha apuntado, nuestra propuesta (que denominaremos *4gr*) se basa en la utilización de n-gramas de caracteres como términos de indexación en lugar de palabras. Para ello el texto es pasado a minúscula y se eliminan los signos de puntuación, pero no los ortográficos. El texto resultante es *tokenizado* en 4-grams [11], para ser finalmente procesado por el motor de indexación.

La *línea de base* con la que comparar nuestra propuesta será una aproximación clásica basada en *stemming* (que denominaremos *stm*) que emplea el *stemmer* Snowball[6], basado en el algoritmo de Porter, y la lista de *stopwords*

[3] Debemos precisar que los experimentos que aquí se muestran deben ser considerados *no oficiales* en tanto que no han sido validados por la organización del CLEF.

[4] *Topics* C050-C059, C070-C079, C100-C109, C120-C129, C150-159 y C180-189.

[5] Frecuencia inversa de documento con normalización 2 de Laplace.

[6] `http://snowball.tartarus.org`

proporcionada por la Universidad de Neuchâtel[7]. Ambos recursos son bien conocidos y de amplio uso entre la comunidad investigadora.

3.2. La generación de errores

Ambas aproximaciones han sido evaluadas mediante la introducción de errores ortográficos en los *topics* para poder así analizar su impacto en los resultados. Se ha probado un amplio rango de *tasas de error* T:

$$T \in \{0\,\%, 10\,\%, 20\,\%, 30\,\%, \ldots, 60\,\%\}$$

donde una tasa T implica que el $T\,\%$ de las palabras del *topic* incluyen errores[8], permitiéndonos así estudiar el comportamiento del sistema incluso para altas tasas de error propias de entornos ruidosos como aquellos en que la entrada se obtiene de dispositivos móviles o basados en escritura a mano (PDAs, bolígrafos digitales y tabletas digitalizadoras, por ejemplo), o incluso interfaces por habla.

Al contrario que en otras aproximaciones [14], los errores no han sido generados artificialmente, sino que hemos desarrollado una metodología que permite trabajar con errores humanos reales, mucho más complejos de generar y controlar, pero también mucho más próximos a la realidad. En una primera fase, se le pidió a un grupo de personas externo a nuestro trabajo que teclearan varias copias de los *topics* originales, preferiblemente tecleando rápido o en ambientes con distracciones (viendo la televisión, por ejemplo) y sin corregir los errores que pudiesen cometer. De esta forma se obtuvo un corpus base formado por 27 copias de los *topics*, donde una media del 7,70 % de sus palabras contenía errores de copia (i.e. T=7,70 %). En una segunda fase se incrementó dicho número de errores. Para ello se alinearon a nivel de palabra todas las copias, permitiendo así acceder, para cada posición, a todas las formas en que dicho término había sido tecleado por los copistas. De este modo se pudo identificar, si lo hubiese, el error más frecuente cometido al teclear aquella palabra en aquella posición, pudiendo así identificar errores de copia para un 65,62 % de los términos (i.e. T=65,62 %, un 60 % en la práctica). Finalmente, en una tercera fase, se generaron los conjuntos de prueba de forma que la tasa de error fuese incremental y acumulativa. Es decir, si un error dado aparece para T=20 %, deberá seguir existiendo para T=30 %, T=40 % y así sucesivamente, evitando de esta forma cualquier distorsión en los resultados. Para ello, todas las palabras que hubiesen sido tecleadas incorrectamente alguna vez se distribuyeron de forma aleatoria pero uniforme en 66 grupos[9]. De esta forma, si queremos generar un conjunto de prueba con una tasa de error T dada, basta con recorrer el texto del *topic* y quedarnos con la versión con errores si y sólo si dicha palabra pertenece a alguno de los T primeros grupos.

[7] `http://www.unine.ch/info/clef/`

[8] T=0 % corresponde al *topic* original. Asimismo, como veremos, el límite de T=60 % ha venido impuesto por el conjunto de prueba.

[9] La tasa máxima era T=65,62 %, lo que supone un grupo por cada 1 % de variación.

Cuadro 1. Resultados de nuestra propuesta para RITE.

	stm		***4gr***	
T	MAP	***%loss***	MAP	***%loss***
0	0,3427	–	0,3075	–
10	0,3289	-4,03	0,2908	-5,43
20	0,3049	-11,03	0,2767	-10,02
30	0,2804	-18,18	0,2642	-14,08
40	0,2194	-35,98	0,2430	-20,98
50	0,1789	-47,80	0,2254	-26,70
60	0,1374	-59,91	0,2061	-32,98
avg.	–	-29,49	–	-18,36

4. Resultados para tolerancia a errores

Los resultados obtenidos se muestran en la Tabla 1. Cada fila corresponde a una tasa de error T dada. Para cada configuración se muestra, por una parte, la precisión media o *mean average precision* obtenida (columna MAP), y por otra, la caída porcentual del rendimiento respecto a MAP para los *topics* originales (para T=0 %, columna *%loss*). La media de tales pérdidas se indica al final de la tabla en la fila *avg.*

En el caso de nuestra línea de base basada en *stemming* (*stm*), las cifras muestran un claro impacto negativo de los errores en el comportamiento del sistema incluso para las tasas de error más bajas, que se vuelve estadísticamente significativo[10] a partir de T=30 %. La pérdida de MAP media (fila *avg.*) es 29,49 %.

Respecto a los n-gramas (*4gr*), si bien el *stemming* obtiene unos mejores resultados iniciales en cuanto a MAP, nuestro planteamiento es menos sensible a los errores, de tal forma que al ir aumentando progresivamente la tasa de error, dicha diferencia se va reduciendo, llegando incluso a superar al *stemming* para $T \geq 40$ %. La pérdida de MAP, además, es ahora claramente menor, con una media de 18,36 % (*avg.*) frente al 29,49 % del *stemming*, poniéndose así en evidencia la robustez de la solución propuesta.

5. Aplicación a multilingüismo

La *Recuperación de Información Multilingüe (RIM)* es un caso particular de RI en el cual consultas y documentos están escritos en idiomas diferentes, por lo que se hace necesario algún tipo de fase de traducción intermedia empleando técnicas de *Traducción Automática (TA)* y así permitir el establecimiento de

[10] A lo largo de este trabajo se han empleado tests-t bilaterales sobre las MAP con α=0,05.

correspondencias. Sin embargo, al contrario que en los sistemas clásicos de TA, en las aplicaciones de RIM no es necesario respetar ciertas restricciones como la de devolver una única traducción o que ésta sea sintácticamente correcta [3].

Es por ello que muchos de estos sistemas emplean métodos de traducción palabra a palabra más sencillos. En esta dirección, el Johns Hopkins University Applied Physics Lab (JHU/APL) propuso ir todavía más lejos y relajar todavía más tales restricciones [10,11]. De esta forma, optan ya no por traducir palabras completas, sino que les basta traducir partes de ella, en concreto los n-gramas que la integran. Para ello emplean un algoritmo de traducción directa de n-gramas que permite traducir a nivel de n-grama de caracteres en lugar de a nivel de palabra, y que está basado en técnicas estadísticas para el alineamiento a nivel de n-grama de corpus paralelos en idiomas diferentes. Aunque desde un punto de vista lingüístico no estamos ante una traducción propiamente dicha, esta aproximación nos permite, en el caso de la RIM, extender las ventajas propias de la utilización de n-gramas como unidad de procesamiento también al proceso de traducción, permitiéndonos así evitar algunas de las limitaciones de las técnicas clásicas de traducción basadas en diccionarios, tales como la necesidad de normalizar las palabras o la imposibilidad de traducir palabras desconocidas. Además, como se trata de una solución que no emplea ningún tipo de procesamiento particular dependiente del idioma, puede ser empleada cuando la disponibilidad de recursos lingüísticos es reducida. Sin embargo, la propuesta original del JHU/APL adolecía de ser lenta, lo que constituía un problema a la hora de ensayar nuevas soluciones o modificaciones, además de integrar numerosas herramientas y recursos *ad-hoc*.

5.1. Descripción del sistema

Tomando como modelo el sistema propuesto por el JHU/APL [11], hemos desarrollado una solución propia intentando preservar las ventajas de la propuesta original pero evitando sus principales desventajas. Primeramente, en lugar de recursos *ad-hoc*, hemos optado por emplear recursos de libre distribución para así minimizar el coste de desarrollo y hacer el sistema más transparente. De este modo emplearemos, por ejemplo, la plataforma Terrier [15] como motor de recuperación y el bien conocido corpus paralelo Europarl [4] para el alineamiento.

Sin embargo, la principal diferencia la constituye el algoritmo de alineamiento de n-gramas, corazón de la solución, y que ahora consta de dos fases. Primeramente el corpus paralelo de entrada es alineado a nivel de palabra empleando la bien conocida herramienta estadística GIZA++ [13], obteniendo como salida las probabilidades de traducción entre las palabras de ambos idiomas. Este primer paso actúa como filtro, ya que sólo aquellos pares de n-gramas correspondientes a palabras alineadas serán considerados para su posterior procesamiento, en contraste con la aproximación original del JHU/APL, de granularidad mucho mayor al partir del corpus alineado a nivel de párrafo para luego procesar todos los n-gramas que contenían dos párrafos alineados. Además hemos introducido varios mecanismos de

filtrado con objeto de reducir el número de alineamientos ambiguos y así reducir el ruido introducido en el sistema. Por una parte, el alineamiento a nivel de palabra es bidireccional [5], es decir, aceptaremos un alineamiento *idiomaOrigen*→*idiomaDestino* (w_s, w_t), donde w_s y w_t denotan respectivamente la palabra origen y su traducción candidata, sólo si existe también el correspondiente alineamiento *idiomaDestino*→*idiomaOrigen* (w_t, w_s). Asimismo serán también desechados aquellos alineamientos cuya probabilidad sea menor que un umbral W=0,15. Esto permite, además, reducir notablemente el consumo de recursos del sistema, al disminuir drásticamente el número de pares de palabras a procesar: hasta un 70 % en el caso del alineamiento bidireccional y un 95 % en el caso del umbral.

Partiendo de los alineamientos de palabras obtenidos, en la segunda fase del algoritmo se procede a calcular las medidas de alineamiento entre n-gramas empleando para ello medidas estadísticas de asociación [9]. Esta solución permite acelerar el proceso de entrenamiento, al concentrar la complejidad en la fase de alineamiento a nivel de palabra y de este modo facilitar el poder probar nuevas medidas de asociación u otros procedimientos en la fase final de alineamiento a nivel de n-gramas propiamente dicha. A la hora de calcular las medidas de asociación estudiaremos las coocurrencias de los diferentes n-gramas que componen dos palabras previamente alineadas por GIZA++. Así, dado el par de n-gramas (g_s, g_t), donde g_s denota el n-grama en la lengua origen y g_t su traducción candidata, sus frecuencias de coocurrencia pueden organizarse en una *tabla de contingencia* resultante de clasificar sus coocurrencias entre ellos y con otros n-gramas presentes en la lista de entrada de palabras alineadas:

	$T = g_t$	$T \neq g_t$	
$S = g_s$	O_{11}	O_{12}	$= R_1$
$S \neq g_s$	O_{21}	O_{22}	$= R_2$
	$= C_1$	$= C_2$	$= N$

La primera fila corresponde a las observaciones donde la palabra en la lengua origen contiene el n-grama g_s, y la segunda fila a aquéllas en las que dicha palabra no lo contiene. Lo mismo ocurre para las columnas para la palabra en la lengua destino y g_t. Las cuentas de estas celdas se denominan *frecuencias observadas*[11]. R_1 y R_2 son las sumas parciales por fila de dichas frecuencias, y C_1 and C_2 las por columna. El número total de pares considerados, N, es la suma total de las frecuencias observadas.

Sin embargo, en este caso deberemos además ponderar la frecuencia de las observaciones en base a la probabilidad asociada a dichos alineamientos. Esto se debe a que no nos encontramos ante observaciones de coocurrencias *reales*, sino únicamente *probables*, ya que GIZA++ emplea un modelo de alineamiento estadístico que calcula la probabilidad de traducción para cada par de palabras

[11] O_{11}, por ejemplo, corresponde al número de alineamientos donde la palabra origen contiene g_s y su corrección candidata contiene g_t, mientras que O_{12} corresponde al número de alineamientos donde la palabra origen contiene g_s pero la corrección candidata no contiene g_t, etcétera.

que coocurran [13]. Por consiguiente, una misma palabra origen puede estar alineada con varias traducciones candidatas, con una probabilidad diferente para cada una. Tomemos como ejemplo el caso de las palabras en inglés `milk` y `milky`, y las españolas `leche`, `lechoso` y `tomate`; un posible alineamiento a nivel de palabra, con sus correspondientes probabilidades y n-gramas componente, podría ser:

source term	**candidate translation**	**prob.**
milk = {milk}	leche = {lech, eche}	0,98
milky = {milk, ilky}	lechoso = {lech, echo, chos, hoso}	0,92
milk = {milk}	tomate = {toma, omat, mate}	0,15

con lo que la tabla de contingencia resultante correspondiente a los pares de n-gramas *(milk, lech)* y *(milk, toma)* sería de la forma:

	$T =$ lech	$T \neq$ lech		$T =$ toma	$T \neq$ toma	
$S =$ milk	O_{11} =**1,90**	O_{12} =**4,19**	R_1 =**6,09**	O_{11} =**0,15**	O_{12} =**5,94**	R_1 =**6,09**
$S \neq$ milk	O_{21} =**0,92**	O_{22} =**2,76**	R_2 =**3,68**	O_{21} =**0**	O_{22} =**3,68**	R_2 =**3,68**
	C_1 =**2,82**	C_2 =**6,95**	N =**9,77**	C_1 =**0,15**	C_2 =**9,62**	N =**9,77**

Obsérvese que, por ejemplo, la frecuencia O_{11} correspondiente a *(milk, lech)* no es 2 como hubiera cabido esperar en otro caso, sino 1,90: el par aparece en dos alineamientos, `milk`–`leche` y `milky`–`lechoso`, pero dichas coocurrencias son ponderadas de acuerdo a la probabilidad de sus alineamientos:

$$O_{11} = 0{,}98 \text{ (de milk–leche)} + 0{,}92 \text{ (de milky–lechoso)} = \mathbf{1{,}90}$$

Una vez generada la tabla, podemos proceder a calcular las medidas de asociación entre cada par de n-gramas. En contraste con la aproximación original del JHU/APL [11], que emplea una medida *ad-hoc*, en nuestro caso hemos optado por medidas estándar: el *coeficiente de Dice* (*Dice*), la *información mutua* (*IM*) y el *log-likelihood* (*logl*), calculadas de la siguiente manera [9]:

$$Dice = \frac{2\,O_{11}}{R_1 + C_1} \qquad IM = log\frac{NO_{11}}{R_1 C_1} \qquad logl = 2\sum_{i,j} O_{ij}\, log\frac{NO_{ij}}{R_i C_j}$$

Retomando el ejemplo anterior, obsérvese que para el caso del coeficiente de Dice, por ejemplo, nos encontramos que la asociación para el par *(milk, lech)*, que es el correcto, es mucho mayor que para el par *(milk, toma)*, que es incorrecto:

$$Dice(\texttt{milk}, \texttt{lech}) = \tfrac{2*1{,}90}{6{,}09+2{,}82} = \mathbf{0{,}43} \quad Dice(\texttt{milk}, \texttt{toma}) = \tfrac{2*0{,}15}{6{,}09+0{,}15} = \mathbf{0{,}05}$$

6. Resultados para multilingüismo

Nuestra aproximación ha sido probada para el caso inglés→español empleando los *topics* en inglés y los documentos en español de la *robust task* del CLEF 2006 empleados en los experimentos de la Sección 3.1, salvo que en

Cuadro 2. Resultados de nuestra propuesta para RIM.

	stm	***4gr***$_{ES}$	***4gr***$_{EN}$	***Dice***$_N$	***Dice***$_T$	***IM***$_N$	***IM***$_T$	***logl***$_N$	***logl***$_T$
MAP	0,3427	0,3075	0,1314	0,2096	0,1589	0,1497	0,1487	0,2231	0,0893
0,00	0,7671	0,6831	0,2845	0,4957	0,4437	0,3782	0,4027	0,5489	0,2171
0,10	0,5974	0,5557	0,2181	0,3782	0,3053	0,3050	0,3000	0,4248	0,1394
0,20	0,5222	0,4722	0,1929	0,3310	0,2606	0,2315	0,2374	0,3620	0,1290
0,30	0,4447	0,4087	0,1610	0,2788	0,2033	0,1927	0,1877	0,2875	0,1082
0,40	0,3922	0,3365	0,1482	0,2320	0,1783	0,1737	0,1632	0,2425	0,0946
0,50	0,3364	0,2800	0,1387	0,2021	0,1535	0,1485	0,1458	0,2124	0,0855
0,60	0,2770	0,2466	0,1226	0,1729	0,1295	0,1269	0,1188	0,1769	0,0731
0,70	0,2241	0,2043	0,0958	0,1302	0,0896	0,0993	0,0980	0,1356	0,0655
0,80	0,1903	0,1722	0,0802	0,1013	0,0666	0,0718	0,0721	0,1015	0,0570
0,90	0,1435	0,1314	0,0704	0,0728	0,0476	0,0497	0,0476	0,0757	0,0540
1,00	0,0795	0,0695	0,0487	0,0452	0,0315	0,0179	0,0181	0,0445	0,0434

este caso los *topics* son los de lengua inglesa. Los parámetros del experimento son también los mismos que los descritos en la Sección 3.1.

El proceso de indexación es también el mismo pero el proceso de consulta sí varía, aunque siempre empleando 4-gramas. La consulta a traducir se descompone primero en n-grams, para luego sustituir dichos n-gramas por sus traducciones de acuerdo a un algoritmo de selección. La consulta traducida resultante es lanzada contra el sistema de recuperación. Actualmente existen dos *algoritmos de selección*: *(a) por rango*, que devuelve los N n-gramas traducción con las medidas de asociación más altas, para $N \in \{1, 2, 3, 5, 10, 20, 30, 40, 50, 75, 100\}$; *(b) por umbral*, que devuelve los n-gramas traducción cuya medida de asociación es superior o igual a un umbral T dado.

6.1. Resultados para el coeficiente de Dice

La Tabla 2 recoge un resumen de los resultados obtenidos en nuestros experimentos, mostrando las precisiones medias (MAP) obtenidas así como los resultados de *precisión respecto cobertura*.

En el caso del algoritmo de selección por rango empleando el coeficiente de Dice, los mejores resultados se obtuvieron con un número limitado de traducciones, siendo los mejores aquéllos para N=1, como se muestra en la columna *Dice*$_N$ de la Tabla 2.

A continuación estudiamos el caso del algoritmo de selección por umbral. En este caso, dado que el coeficiente de Dice toma valores en el rango $[0..+1]$, hemos optado por emplear una serie de umbrales T de la forma:

$$T \in \{0;\ 0,001;\ 0,01;\ 0,05;\ 0,10;\ 0,2;\ 0,3;\ 0,4;\ 0,5;\ 0,6;\ 0,7;\ 0,8;\ 0,9;\ 1\}$$

El mejor resultado, obtenido para T=0,40, se muestra en la columna *Dice*$_T$. Como se puede apreciar, los resultados no fueron tan buenos (significativamente) como los anteriores.

6.2. Resultados para la información mutua

En este caso, los resultados obtenidos para la selección por rango no fueron tan positivos como con el coeficiente de Dice. Los mejores, con N=10, se muestran en la columna IM_N.

En el caso de la selección por umbral, debemos tener ahora en cuenta que la medida IM puede tomar cualquier valor en el rango $(-\infty .. +\infty)$, donde además los valores negativos corresponden a pares de términos que se evitan mutuamente. De este modo, y para homogeneizar los test, los umbrales a emplear se calcularon de acuerdo a la siguiente fórmula:

$$T_i = \mu + 0{,}5\ i\ \sigma$$

donde T_i representa el i-ésimo umbral (con $i \in \mathbb{Z}^+$), μ representa la *media* de los valores de asociación obtenidos y σ representa su *desviación típica*. Nuestras pruebas demuestran que en el caso de selección por umbral, no hay diferencias significativas con los resultados de la selección por rango. Los mejores resultados, para $T = \mu$, se muestran en la columna IM_T.

6.3. Resultados para el *log-likelihood*

Para el algoritmo de selección por rango, los mejores resultados se obtuvieron limitando el número de candidatos, siendo aquéllos con N=1 los mejores, los cuales se muestran en la columna $logl_N$.

En el caso del algoritmo por umbral, como en el caso de la IM, no existe un rango fijo de valores posibles. Tras estudiar la distribución de los alineamientos de n-gramas obtenidos con respecto a sus valores de log-likelihood, se decidió emplear esta vez granularidades variables:

$$T_i = \begin{cases} \mu + 0{,}05\ i\ \sigma & -\infty < i \leq 2 \\ \mu + 0{,}50\ (i-2)\ \sigma & 2 < i < +\infty \end{cases}$$

donde T_i representa el i-ésimo umbral (con $i \in \mathbb{Z}$), μ representa la *media* de los valores de log-likelihood obtenidos y σ representa su *desviación típica*. Los resultados que obtuvimos fueron los más bajos de los todas las configuraciones estudiadas. Los mejores de ellos, para $T = \mu + 3\sigma$, se muestran en la columna $logl_T$.

Finalmente y con intención de completar nuestra evaluación, hemos comparado los resultados anteriores con varias líneas de base: una ejecución monolingüe en español empleando *stemming* (stm_{ES}); otra ejecución monolingüe en español empleando 4-gramas como términos ($4gr_{ES}$) que constituiría nuestro rendimiento objetivo deseado; y una última ejecución empleando 4-gramas lanzando directamente las consultas en inglés sin traducción alguna contra el índice en español ($4gr_{EN}$) que nos permite medir el impacto de las correspondencias casuales. Como podemos apreciar los mejores resultados fueron los obtenidos para el *log-likelihood* empleando el algoritmo de selección por rango, si bien la diferencia con aquéllos obtenidos con el coeficiente de Dice no

resultó ser significativa. Por otra parte, ambas aproximaciones se comportaron significativamente mejor que la información mutua. Los resultados obtenidos, pues, han sido muy positivos y alentadores, si bien necesitan todavía ser mejorados para seguir acercándose a nuestro rendimiento objetivo.

7. Conclusiones y trabajo futuro

Este trabajo plantea la utilización de n-gramas de caracteres como unidad de procesamiento en dos contextos diferentes: la Recuperación de Información Tolerante a Errores y la Recuperación de Información Multilingüe. El objetivo en ambos casos es beneficiarse de su simplicidad de uso e integración, su robustez y su independencia respecto al idioma y al dominio.

En el caso de la recuperación tolerante a errores ortográficos en las consultas, el empleo de n-gramas permite trabajar directamente sobre el texto con errores sin procesamiento alguno a mayores, ya que las correspondencias no se establecen ya a nivel de palabra completa, sino a nivel de sus subcadenas, posibilitando las correspondencias parciales y, por tanto, aumentando la robustez. De este modo los experimentos realizados mostraron que si bien las aproximaciones clásicas basadas en *stemming* son altamente sensibles a los errores ortográficos, los n-gramas confirmaron tener una robustez considerablemente mayor.

Asimismo, para realizar dichos experimentos se desarrolló una metodología que permite introducir de forma sencilla errores humanos reales en el conjunto de *topics* de entrada, posibilitando además elegir la tasa de error deseada, para así poder analizar el impacto de los mismos en el rendimiento del sistema.

En el caso de la recuperación multilingüe, este trabajo propone emplear n-grams no sólo como términos de indexación, como en el caso anterior, sino también como unidades de traducción. Para ello se describe un algoritmo para el alineamiento a nivel de n-grama de textos paralelos. Los alineamientos resultantes son empleados para la traducción de la consulta, también a nivel de n-grama. Tal aproximación permite evitar algunas de las limitaciones propias de las técnicas clásicas de traducción basadas en diccionarios, tales como la necesidad de normalizar las palabras o la imposibilidad de traducir palabras desconocidas. Además, como se trata de una solución que no emplea ningún tipo de procesamiento particular dependiente del idioma, puede ser empleada cuando la disponibilidad de recursos lingüísticos es reducida. Los resultados obtenidos han sido muy positivos y alentadores.

Con respecto al trabajo futuro, el principal problema de utilizar n-gramas como términos índice es el gran tamaño de los índices resultantes. Para reducir su tamaño queremos estudiar el empleo de técnicas de *pruning* [1] así como extender el concepto de *stopword* al caso de n-gramas para eliminar aquéllos más frecuentes y menos discriminantes. Tales *stop-n-grams* deberían ser generados automáticamente a partir de los textos de entrada [7] para así preservar su independencia respecto al idioma y el dominio. Pretendemos, además, ampliar nuestros experimentos a una mayor número de idiomas y tipos de consulta.

Referencias

1. D. Carmel, D. Cohen, R. Fagin, E. Farchi, M. Herscovici, Y.S. Maarek, and A. Soffer. Static index pruning for information retrieval systems. In *Proc. of SIGIR'01*, pages 43–50, 2001. ACM.
2. B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesley, 2009.
3. G. Grefenstette, editor. *Cross-Language Information Retrieval*, volume 2 of *The Kluwer International Series on Information Retrieval*. Kluwer Academic Publishers, 1998.
4. P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of the 10th Machine Translation Summit*, pages 79–86, 2005. Corpus disponible en http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/europarl/ (visitada en abril 2010).
5. P. Koehn, F.J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. of the 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03)*, pages 48–54, 2003. ACL.
6. K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4):377–439, 1992.
7. R.T.W. Lo, B. He, and I. Ounis. Automatically building a stopword list for an information retrieval system. In *Proc. of the 5th Dutch-Belgian Information Retrieval Workshop (DIR'05)*, 2005.
8. C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
9. C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.
10. P. McNamee and J. Mayfield. Character N-gram Tokenization for European Language Text Retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.
11. P. McNamee and J. Mayfield. JHU/APL experiments in tokenization and non-word translation. volume 3237 of *Lecture Notes in Computer Science*, pages 85–97. 2004.
12. A. Nardi, C. Peters, and J.L. Vicedo, editors. *Working Notes of the CLEF 2006 Workshop*, 2006. Disponible en http://www.clef-campaign.org (visitada en abril 2010).
13. F.J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003. Herramienta GIZA++ disponible en http://www.fjoch.com/GIZA++.html (visitada en abril 2010).
14. J. Otero, J. Vilares, and M. Vilares. Corrupted queries in Spanish text retrieval: error correction vs. n-grams. In *Proc. of ACM CIKM 2008 Workshop on Improving Non-English Web Searching (iNEWS'08)*, pages 39–46, 2008. ACM.
15. I. Ounis, C. Lioma, C. Macdonald, and V. Plachouras. Research directions in Terrier: a search engine for advanced retrieval on the web. *Novática/UPGRADE Special Issue on Web Information Access*, 8(1):49–56, 2007. Herramienta Terrier disponible en http://www.terrier.org (visitada en abril 2010).
16. A.M. Robertson and P. Willett. Applications of n-grams in textual information systems. *Journal of Documentation*, 54(1):48–69, January 1998.
17. M. Vilares, J. Graña, and P. Alvariño. Finite-state morphology and formal verification. *Journal of Natural Language Engineering, special issue on Extended Finite State Models of Language*, 3(4):303–304, 1997.