

formar una cadena en otra.

Trabajos más recientes interpretan la corrección ortográfica como una cuestión estadística, donde una consulta con errores es vista como una degeneración probabilística de una correcta (Brill y Moore, 2000). Esta aproximación, conocida como *modelo de canal ruidoso* (Kernighan, Church, y Gale, 1990),² también proporciona formas de incorporar información de pronunciación para mejorar el rendimiento por medio de la captura de similitudes en la pronunciación de las palabras (Toutanova y Moore, 2002).

Sin embargo, en este trabajo proponemos una estrategia basada en n -gramas de caracteres como alternativa para el tratamiento de consultas degradadas en español, buscando, además, una metodología simple y que pueda ser utilizada independientemente de la base de datos documental considerada y de los recursos lingüísticos disponibles. Presentaremos, también, dos aproximaciones basadas en corrección ortográfica no interactiva.

Este artículo se estructura como sigue. En primer lugar, en la Sección 2 describimos brevemente nuestra propuesta basada en n -gramas de caracteres. A continuación, en la Sección 3, se presentan las dos aproximaciones de corrección ortográfica que han sido comparadas con nuestra propuesta. En la Sección 4 se describe nuestra metodología de evaluación y los experimentos realizados. Finalmente, la Sección 5 contiene nuestras conclusiones y propuestas de trabajo futuro.

2. Recuperación de Texto mediante N -Gramas de Caracteres

Formalmente, un n -grama es una subsecuencia de longitud n de una secuencia dada. Así, por ejemplo, podemos dividir la palabra "patata" en los 3-gramas de caracteres superpuestos -pat-, -ata-, -tat- y -ata-. Este simple concepto ha sido redescubierto recientemente por el *Johns Hopkins University Applied Physics Lab* (JHU/APL) (McNamee y Mayfield, 2004a) de cara a la indexación de documentos, y nosotros lo recuperamos ahora para nuestra propuesta.

Al tratar con RI monolingüe, la adaptación resulta sencilla ya que tanto las consultas como los documentos son simplemente tokenizados en n -gramas superpuestos en

lugar de palabras. Los n -gramas resultantes son entonces procesados como lo haría cualquier motor de recuperación. Su interés viene dado por las posibilidades que ofrecen, especialmente en lengua no inglesa, al facilitar un modo alternativo para la normalización de formas de palabras y permitir tratar lenguas muy diferentes sin procesamiento específico al idioma y aún cuando los recursos lingüísticos disponibles son escasos o inexistentes.

Estaríamos, pues, ante un prometedor punto de partida sobre el cual desarrollar una estrategia de indexación y recuperación efectiva para el tratamiento de consultas degradadas. Además, la utilización de índices basados en n -gramas desmonta el principal argumento que justifica la integración de métodos de corrección ortográfica en aplicaciones de RI robustas: la necesidad de una coincidencia exacta con los términos almacenados en los índices. De este modo, con el empleo de n -gramas en lugar de palabras completas, sólo se requeriría la coincidencia en subcadenas de éstas. En la práctica, esto elimina la necesidad de normalizar los términos, minimizando además el impacto de los errores ortográficos, a los que no se les prestaría especial atención. En general debería, además, reducir de forma considerable la incapacidad del sistema para manejar las palabras desconocidas.

3. Corrección Ortográfica

Con el fin de justificar el interés práctico de nuestra propuesta de RI robusta basada en n -gramas de caracteres, introducimos también una aproximación más clásica asociada a un corrector ortográfico contextual (Otero, Graña, y Vilares, 2007), lo que nos permite definir un marco de pruebas comparativo. En un principio aplicaremos un algoritmo global de corrección ortográfica sobre autómatas finitos, propuesto por Savary (Savary, 2002), que encuentra todas las palabras cuya distancia de edición con la palabra errónea sea mínima.

Desafortunadamente, esta técnica puede devolver varias reparaciones candidatas posibles que, desde un punto de vista morfológico, tengan una calidad similar, es decir, cuando existan varias palabras cuya distancia de edición con la palabra errónea es la misma.

Sin embargo, es posible ir más allá de la propuesta de Savary aprovechando la información lingüística contextual embebida en un proceso de etiquetación con el fin de

²Noisy channel model en inglés.

ordenar las correcciones candidatas. Hablamos entonces de *corrección ortográfica contextual*, cuyo núcleo, en nuestro caso, es un etiquetador morfosintáctico estocástico basado en una extensión dinámica del algoritmo de Viterbi sobre *Modelos Ocultos de Markov* (Graña, Alonso, y Vilares, 2002) de segundo orden. Esta extensión del algoritmo de Viterbi original se aplica sobre retículas en lugar de enrejados (ver Figura 1) ya que éstas son mucho más flexibles al ser representadas las palabras en los arcos en lugar de en los nodos. En el contexto de la corrección ortográfica, nos permite representar un par *palabra/etiqueta* en cada arco, y luego calcular la probabilidad de cada uno de los caminos por medio de una adaptación de las ecuaciones del algoritmo de Viterbi.

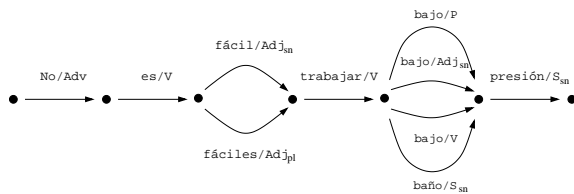


Figura 1: Alternativas de corrección ortográfica representadas en una retícula.

Para ilustrar el proceso con un ejemplo, consideremos la frase “No es fácil trabajar baio presión”, cuya corrección esperada sería “No es fácil trabajar bajo presión”, donde las palabras “fácil” y “baio” son erróneas. Asumamos ahora que nuestro corrector ortográfico nos ofrece “fácil”/Adjetivo singular y “fáciles”/Adjetivo plural como posibles correcciones para “fácil”; y “bajo”/Adjetivo singular, “bajo”/Preposición, “bajo”/Verbo y “baño”/Sustantivo singular para “baio”. La ejecución del algoritmo de Viterbi dinámico sobre la retícula asociada, mostrada en la Figura 1, nos ofrecería tanto las etiquetas de las palabras como las correcciones más probables en el contexto de esa frase concreta, lo que nos permitiría obtener una lista ordenada de correcciones candidatas. De este modo obtendríamos, para nuestro ejemplo, que las correcciones deseadas, “fácil”/Adjetivo singular y “bajo”/Preposición, serían las primeras opciones, ya que se corresponderían con la secuencia de etiquetas correcta.

4. Evaluación

Nuestra propuesta ha sido inicialmente testada para el español. Este idioma puede

ser considerado un ejemplo significativo dado que muestra una gran variedad de procesos morfológicos, lo que lo convierte en una lengua difícil para la corrección ortográfica (Vilares, Otero, y Graña, 2004). Las características más diferenciadoras se encuentran en los verbos, con un paradigma de conjugación altamente complejo. En el caso de sustantivos y adjetivos esta complejidad se extiende al número y al género, con hasta 10 y 20 grupos de variación respectivamente.

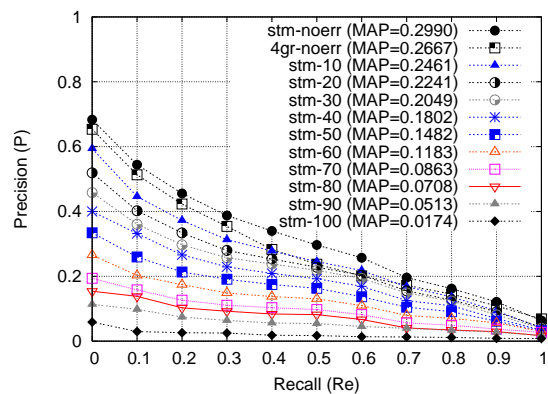


Figura 2: Precisión vs. Cobertura para las consultas sin corregir (empleando *stemming*).

4.1. Procesamiento de Errores

La primera fase en el proceso de evaluación consiste en introducir errores ortográficos en el conjunto de consultas de prueba. Estos errores son introducidos de forma aleatoria por un generador de errores automático de acuerdo con un ratio de error dado. Inicialmente se genera un *fichero maestro de errores* como sigue. Para cada palabra de más de 3 caracteres de la consulta, se introduce en una posición aleatoria uno de los cuatro errores de edición descritos por Damerau (Damerau, 1964). De este modo, los errores introducidos son similares a aquellos que cometería un ser humano o un dispositivo OCR. Al mismo tiempo se genera un valor aleatorio entre 0 y 100 que representa la probabilidad de que la palabra no contenga ningún error ortográfico. De este modo obtenemos un fichero maestro de errores que contiene, para cada palabra, su forma errónea correspondiente, y un valor de probabilidad.

Todos estos datos hacen posible generar de una forma sencilla conjuntos de prueba diferentes para distintos ratios de error, permitiéndonos así valorar el impacto de esta variable en los resultados. El procedimiento

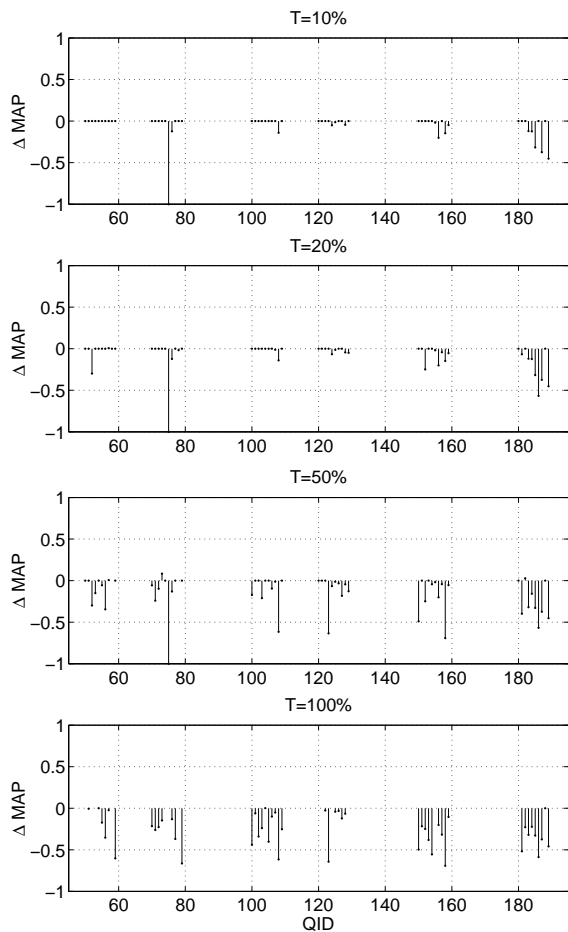


Figura 3: Diferencias de MAP por consulta: consultas sin corregir vs. consultas originales (empleando *stemming*).

consiste en recorrer el fichero maestro de errores y seleccionar, para cada palabra, la forma original en el caso de que su probabilidad sea mayor que el ratio de error fijado, o la forma errónea en caso contrario. Así, dado un ratio de error T , sólo el $T\%$ de las palabras de las consultas contendrán un error. Una característica interesante de esta solución es que los errores son incrementales, ya que las formas erróneas que están presentes para un ratio de error determinado continuarán estando presentes para ratios de error mayores, evitando así cualquier distorsión en los resultados.

El siguiente paso consiste en procesar las consultas con errores y lanzarlas contra el sistema de RI. En el caso de nuestra propuesta basada en n -gramas no se precisan recursos extra, ya que el único procesamiento necesario consiste en tokenizar las consultas en n -gramas. Sin embargo, para las aproximaciones de corrección ortográfica se necesita un

lexicón y, en el caso de la corrección contextual, también un corpus de entrenamiento etiquetado manualmente para entrenar con él el etiquetador. En nuestros experimentos hemos trabajado con el corpus de español MULTEX-JOC (Véronis, 1999), que consta de alrededor de 200.000 palabras etiquetadas morfo-sintácticamente, y con su lexicón asociado, de 15.548 palabras.

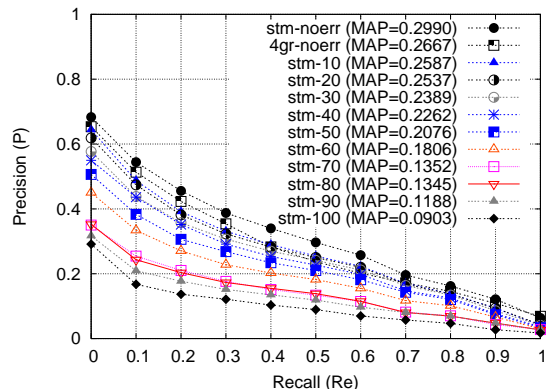


Figura 4: Precisión vs. Cobertura para las consultas corregidas mediante el algoritmo de Savary (empleando *stemming*).

4.2. Marco de Evaluación

En nuestros experimentos se ha empleado el corpus de español de la *robust task* del CLEF 2006 (Nardi, Peters, y Vicedo, 2006),³ formado por 454.045 documentos (1,06 GB) y 160 *topics* —a partir de los cuales generar las consultas— de los que hemos empleado únicamente un subconjunto del mismo (*training topics*) formado por 60 *topics* proporcionados por el CLEF específicamente para tareas de entrenamiento y puesta a punto.⁴ Dichos *topics* están formados por 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. En cualquier caso únicamente hemos empleado el campo de *título* para así simular el caso de las consultas cortas utilizadas en motores comerciales.

Partiendo de dicha colección de documentos se han generado dos índices diferentes. Primeramente, para probar las propuestas basadas en corrección ortográfica, se ha usa-

³Estos experimentos han de considerarse no oficiales, ya que los resultados no han sido evaluados por la organización.

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

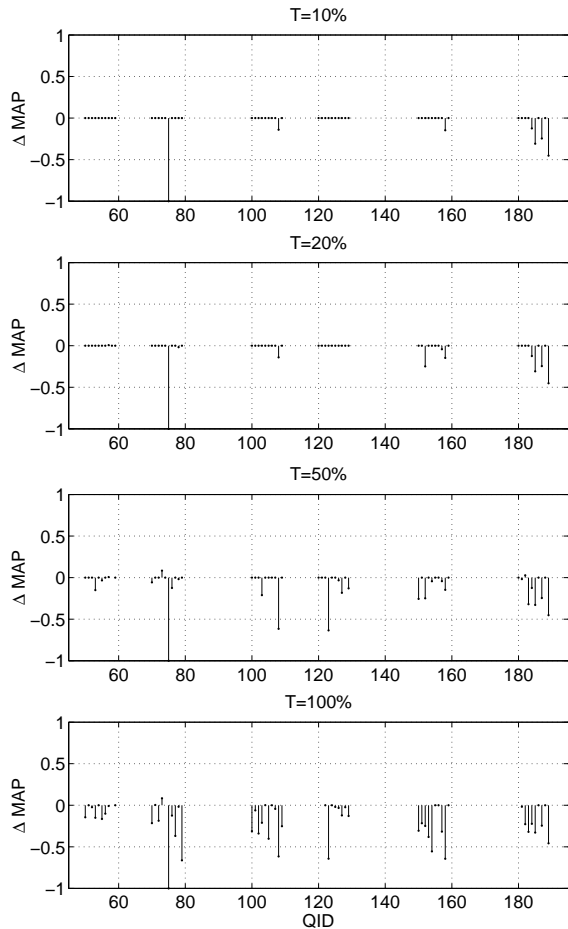


Figura 5: Diferencias de MAP por consulta: consultas corregidas mediante el algoritmo de Savary vs. consultas originales (empleando *stemming*).

do una aproximación clásica basada en *stemming* empleando SNOWBALL,⁵ basado en el algoritmo de Porter (Porter, 1980), y la lista de *stopwords* de la Universidad de Neuchatel.⁶ Ambos recursos son de uso amplio entre la comunidad de IR. Asimismo, en el caso de las consultas, se ha utilizado una segunda lista de *meta-stopwords* (Mittendorfer y Winiwarter, 2001; Mittendorfer y Winiwarter, 2002). Dichas *stopwords* corresponden a metacontenido, es decir, expresiones de formulación de la consulta que no aportan ninguna información útil para la búsqueda, como en el caso de la expresión “encuentre aquellos documentos que describan ...”.

En segundo lugar, a la hora de probar nuestra solución basada en *n*-gramas, los documentos han sido convertidos a minúsculas y se han eliminado los signos de puntuación,

⁵<http://snowball.tartarus.org>

⁶<http://www.unine.ch/info/clef/>

aunque no los signos ortográficos. El texto resultante ha sido tokenizado e indexado utilizando 4-gramas como longitud de compromiso tras estudiar los resultados previos del JHU/APL (McNamee y Mayfield, 2004b). En este caso no se han empleado *stopwords*.

Finalmente, ya a nivel de implementación, nuestro sistema emplea como motor de recuperación la plataforma de código abierto TERRIER (Ounis et al., 2006) con un modelo InL2 (Amati y van Rijsbergen, 2002).⁷

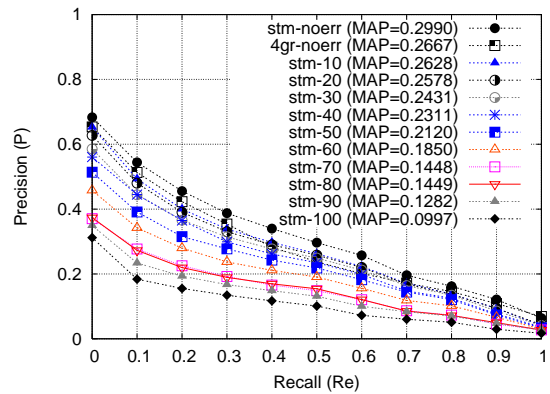


Figura 6: Precisión vs. Cobertura para las consultas corregidas mediante el algoritmo de corrección contextual (empleando *stemming*).

4.3. Resultados Experimentales

Nuestra propuesta ha sido probada para un amplio rango de ratios de error T con el fin de estudiar el comportamiento del sistema no sólo para densidades de error bajas, sino también para los elevados ratios de error propios de entornos ruidosos como aquellos en que la entrada se obtiene de dispositivos móviles o basados en escritura a mano —PDAs y tabletas digitalizadoras, por ejemplo. De este modo se ha trabajado con:

$$T \in \{0\%, 10\%, 20\%, 30\%, \dots, 100\%\}$$

donde $T=0\%$ significa que no se han introducido errores.

En el primer conjunto de experimentos realizados se utilizaron las consultas sin corregir aplicando una aproximación clásica basada en *stemming*. Los resultados obtenidos para cada ratio de error T se muestran en las gráficas de la Figura 2 tomando como referencia tanto los resultados obtenidos para las

⁷Inverse Document Frequency model with Laplace after-effect and normalization 2.

consultas originales aplicando *stemming* —es decir, para $T=0\%$ — (*stm-noerr*), como los obtenidos aplicando la aproximación basada en n -gramas (*4gr-noerr*). También se dan los valores de precisión media (MAP).⁸ Estos primeros resultados muestran que el *stemming* es sensible a los errores ortográficos. Como se puede apreciar, aún un ratio de error bajo como $T=10\%$ tiene un impacto significativo sobre el rendimiento⁹ —la MAP decrece el 18%—, empeorando conforme aumenta el número de errores introducidos: pérdida del 25% para $T=20\%$, 50% para $T=50\%$ (con 2 consultas que ya no recuperan ningún documento) y 94% para $T=100\%$ (con 13 consultas sin documentos), por ejemplo. Tales variaciones, ya a nivel de consulta, se muestran en la Figura 3. Esto se debe al hecho de que con el tipo de consultas que estamos utilizando aquí —con unas 4 palabras de media—, cada palabra es de vital importancia, ya que la información perdida cuando un término ya no encuentra correspondencia debido a un error ortográfico no puede ser recuperada a partir de ningún otro término.

En nuestra segunda ronda de experimentos se estudió el comportamiento del sistema al usar la primera de las aproximaciones de corrección consideradas en este trabajo, esto es, cuando lanzamos las consultas con errores tras ser procesadas con el algoritmo de Savary. En este caso el módulo de corrección toma como entrada la consulta con errores, obteniendo como salida una versión corregida donde cada palabra incorrecta ha sido substituida por el término más cercano del lexicón de acuerdo a la distancia de edición. En caso de empate —es decir, cuando existen varias palabras en el lexicón a la misma distancia—, la consulta es expandida con todas las correcciones empatadas. Por ejemplo, tomando como entrada la oración considerada en la Sección 3, “*No es f^ácil trabajar baio presión*”, la salida sería “*No es f^ácil f^áciles trabajar bajo ba^ño presión*”. Analizando los resultados obtenidos, mostrados en la Figura 4, vemos que la corrección tiene un efecto general significativamente positivo sobre el rendimiento, disminuyendo en gran medida —aunque no eliminando— el impacto de los errores ortográficos, no sólo para ratios de error bajos (la pérdida de MAP disminuye del

18% al 13% para $T=10\%$ y del 25% al 15% para $T=20\%$), sino también para ratios de error altos y muy altos (del 50% al 31% para $T=50\%$ y del 94% al 70% para $T=100\%$), reduciéndose también el número de consultas que no devuelven documentos (ahora sólo 1 para $T=50\%$ y 5 para $T=100\%$). Las diferencias de MAP a nivel de consulta se muestran en la Figura 5. Asimismo, el análisis de los datos muestra que la efectividad relativa de la corrección aumenta con el ratio de error.

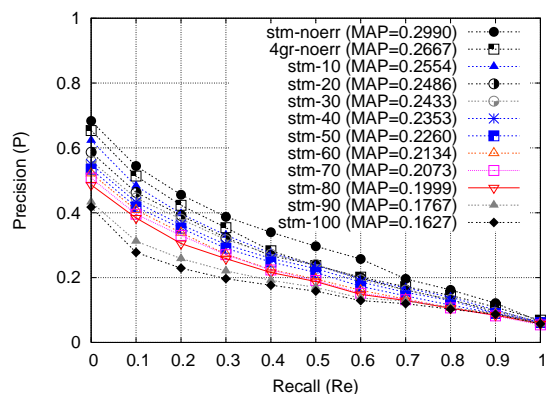


Figura 7: Precisión vs. Cobertura para las consultas sin corregir (empleando n -gramas).

Con el fin de eliminar el ruido introducido por los empates al emplear el algoritmo de Savary, se ha realizado un tercer conjunto de pruebas usando nuestro corrector ortográfico contextual. Dichos resultados se muestran en la Figura 6 y, como era de esperar, éstos mejoran consistentemente con respecto a la aproximación original, si bien la mejora obtenida mediante este procesamiento extra no llega a ser significativa: un 2% de pérdida de MAP recuperado para $10\% \leq T \leq 60\%$ y un 7–10% para $T > 60\%$.

Finalmente, hemos probado nuestra propuesta basada en n -gramas. La Figura 7 muestra los resultados obtenidos cuando las consultas sin corregir son lanzadas contra nuestro sistema de RI basado en n -gramas. Aunque el *stemming* funciona significativamente mejor que los n -gramas para las consultas originales, no ocurre lo mismo cuando hay errores ortográficos, superando claramente el segundo método al primero no sólo cuando no se aplica ningún tipo de corrección, siendo la mejora significativa para $T \geq 40\%$, sino también cuando se aplica cualquiera de los dos métodos basados en corrección ortográfica —salvo para ratios de error muy bajos—, si bien la diferencia no es sig-

⁸ Mean average precision en inglés.

⁹ A lo largo de este trabajo se han empleado tests- t bilaterales sobre las MAP con $\alpha=0.05$.

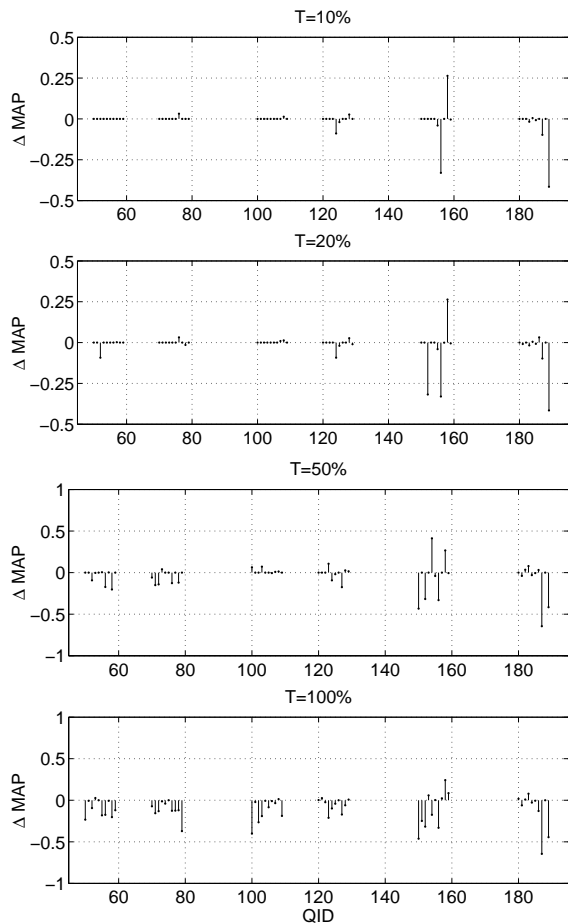


Figura 8: Diferencias de MAP por consulta: consultas sin corregir vs. consultas originales (empleando n -gramas).

nificativa hasta $T \geq 70\%$. Además, la robustez de nuestra propuesta basada en n -gramas en presencia de errores ortográficos demuestra ser claramente superior a cualquiera de las aproximaciones previas basadas en *stemming*. Como ejemplo, la pérdida de MAP para *stemming* —como se dijo previamente— era significativa incluso para $T=10\%$, con una reducción del 18% para $T=10\%$, 25% para $T=20\%$, 50% para $T=50\%$ y 94% para $T=100\%$. Para los mismos valores de T , la aplicación de nuestro corrector ortográfico contextual —ligeramente superior a la propuesta de Savary— reducía dichas pérdidas a 12%, 14%, 29% y 67%, respectivamente, con lo que dichas caídas ya no eran significativas hasta $T=20\%$. Sin embargo, los n -gramas superan a ambos de forma clara, siendo la pérdida de MAP significativa sólo a partir de $T=40\%$, y casi reduciendo a la mitad la cuantía de dichas pérdidas: 4%, 7%, 15% y 39%, respectivamente. Además, ya no

hay consultas que no devuelven documentos, ni siquiera para $T=100\%$. El rendimiento a nivel de consulta se muestra en la Figura 8.

5. Conclusiones y Trabajo Futuro

Este trabajo es un primer paso hacia el diseño de técnicas de consulta para su empleo en aplicaciones de base lingüística para dominios genéricos no especializados. Nuestro objetivo es el tratamiento eficiente de las consultas degradadas en español, evitando métodos clásicos de corrección ortográfica que requieran una implementación compleja, no sólo desde el punto de vista computacional sino también desde el lingüístico. En este sentido, se proponen aquí dos aproximaciones diferentes. En primer lugar, se presenta un corrector ortográfico contextual desarrollado a partir de una técnica de corrección global previa ampliada para incluir información contextual obtenida mediante etiquetación morfosintáctica. Nuestra segunda propuesta consiste en trabajar directamente con las consultas con errores ortográficos, pero utilizando un sistema de RI basado en n -gramas en lugar de uno clásico basado en *stemming*.

Las pruebas realizadas han mostrado que las aproximaciones clásicas basadas en *stemming* son sensibles a los errores ortográficos, aunque el uso de mecanismos de corrección permiten reducir el impacto negativo de éstos. Por su parte, los n -gramas de caracteres han mostrado ser altamente robustos, superando claramente a las técnicas basadas en corrección ortográfica, especialmente para ratios de error medios o altos. Además, dado que no se precisa procesamiento específico al idioma, nuestra aproximación basada en n -gramas puede ser utilizada con lenguas de naturaleza diferente aún cuando los recursos lingüísticos disponibles sean escasos o inexistentes.

Con respecto a nuestro trabajo futuro, tenemos la intención de ampliar el concepto de *stopword* al caso de n -gramas de caracteres con el fin de incrementar el rendimiento del sistema así como reducir sus requerimientos computacionales y de almacenamiento. Sin embargo, con el fin de mantener la independencia respecto al idioma, tales "*stop-n-gramas*" deberían ser generados de forma automática a partir de los propios textos de entrada (Lo, He, y Ounis, 2005). Finalmente, se están preparando nuevos experimentos para otros idiomas.

Bibliografía

- Amati, G. y C. J. van Rijsbergen. 2002. Probabilistic models of Information Retrieval based on measuring divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389.
- Brill, E. y R. C. Moore. 2000. An improved error model for noisy channel spelling correction. En *Proc. of the ACL'00*, pág. 286–293.
- Damerau, F. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176.
- Graña, J., M. A. Alonso, y M. Vilares. 2002. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. *Lecture Notes in Computer Science*, 2448:3–10.
- Kernighan, M. D., K. W. Church, y W. A. Gale. 1990. A spelling correction program based on a noisy channel model. En *Proc. of the COLING'90*, pág. 205–210.
- Levenshtein, V.I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady*, 6:707–710.
- Lo, R.T.W., B. He, y I. Ounis. 2005. Automatically building a stopword list for an information retrieval system. En *Proc. of the 5th Dutch-Belgian Information Retrieval Workshop (DIR'05)*.
- McNamee, P. y J. Mayfield. 2004a. Character N-gram tokenization for European language text retrieval. *Information Retrieval*, 7(1-2):73–97.
- McNamee, P. y J. Mayfield. 2004b. JHU/APL experiments in tokenization and non-word translation. *Lecture Notes in Computer Science*, 3237:85–97.
- Mittendorfer, M. y W. Winiwarter. 2001. A simple way of improving traditional IR methods by structuring queries. En *Proc. of the 2001 IEEE International Workshop on Natural Language Processing and Knowledge Engineering (NLPKE 2001)*.
- Mittendorfer, M. y W. Winiwarter. 2002. Exploiting syntactic analysis of queries for information retrieval. *Data & Knowledge Engineering*, 42(3):315–325.
- Nardi, A., C. Peters, y J.L. Vicedo, eds. 2006. En *Working Notes of the CLEF 2006 Workshop*. Disponible en <http://www.clef-campaign.org> (visitada en octubre 2008).
- Otero, J., J. Graña, y M. Vilares. 2007. Contextual spelling correction. *Lecture Notes in Computer Science*, 4739:290–296.
- Ounis, I., G. Amati, V. Plachouras, B. He, C. Macdonald, y C. Lioma. 2006. TERRIER: A high performance and scalable Information Retrieval platform. En *Proc. of the ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, pág. 18–25. Herramienta disponible en <http://ir.dcs.gla.ac.uk/terrier/> (visitada en octubre 2008).
- Porter, M.F. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.
- Savary, A. 2002. Typographical nearest-neighbor search in a finite-state lexicon and its application to spelling correction. *Lecture Notes in Computer Science*, 2494:251–260.
- Toutanova, K. y R.C. Moore. 2002. Pronunciation modeling for improved spelling correction. En *Proc. of the ACL'02*, pág. 144–151.
- Vilares, M., J. Otero, y J. Graña. 2004. On asymptotic finite-state error repair. *Lecture Notes in Computer Science*, 3246:271–272.
- Véronis, J. 1999. MULTEXT-corpora: An annotated corpus for five European languages. CD-ROM. Distributed by ELRA/ELDA.