# Presentación sinóptica de textos bilingües mediante distancias de edición

Sergio Ortiz Rojas, Rafael C. Carrasco
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante

**Resumen** Aunque la alineación de textos multilingües mediante métodos de traducción estadística consigue buenos resultados, su implementación es compleja y su fundamentación teórica es, a menudo, intrincada. En el caso de textos escritos en idiomas emparentados (por ejemplo, el latín y el castellano) hemos conseguido resultados satisfactorios usando programas muy simples basados en la distancia de edición carácter a carácter y modificados para reducir su coste temporal. Este procedimiento permite incluso detectar omisiones y otras divergencias locales entre los textos. Para idiomas muy diferentes, es posible utilizar un traductor automático para generar un texto intermedio que facilite el alineamiento de los textos originales.

## 1 Introducción

En aquellas bibliotecas digitales que contienen *bitextos*, esto es, textos paralelos en lenguas diferentes, interesa la presentación sinóptica de los textos de forma que el lector pueda consultar de manera sencilla una frase y su correspondiente traducción en el texto paralelo. El alineamiento manual de bitextos es costoso y requiere la intervención de personas con conocimiento de las dos lenguas involucradas, por lo que conviene automatizar en la medida de lo posible la tarea. Aunque a primera vista esto parece sencillo (por ejemplo, contando el numero de frases o párrafos en cada texto), en la practica se dan divergencias en la traducción (las omisiones, las inserciones, las reiteraciones y las reubicaciones de fragmentos son las más frecuentes) que dificultan la obtención de resultados aceptables.

Una solución a este problema consiste en emplear algoritmos de traducción estadística (Brown et al., 1990). Este tipo de algoritmos utiliza modelos probabilísticos, obtenidos automáticamente a partir de los mismos bitextos, para la generación de traducciones. El modelo asigna aleatoriamente a cada palabra del texto origen una *fertilidad* (el número de palabras en el texto meta que se derivan de la palabra original) y un conjunto de palabras derivadas. Además incluye probabilidades de desplazamiento para reordenar la traducción obtenida. Por tanto, cada modelo contiene, implícitamente, un modelo probabilístico de alineamiento entre la lengua origen y la lengua meta, lo que permite la aplicación de la traducción estadística al alineamiento de bitextos. Si bien esta técnica puede aplicarse a cualquier par de idiomas, no está exenta de ciertas dificultades:

1. Por su carácter estadístico requiere trabajar con textos de una longitud suficiente y, aún así, puede tener dificultades para encontrar la traducción exacta (o el alineamiento) de las palabras menos frecuentes.
2. Los algoritmos no son fáciles de entender, son muy complejos de implementar eficientemente y están protegidos, en parte, por una patente (US 5510981).

Sin embargo, en el caso de lenguas con raíz común, es posible realizar una alineación basándose únicamente en la similitud de las palabras o secuencias de caracteres que aparecen en cada texto. Por ejemplo, `char_align` (Church, 1993) busca secuencias de caracteres idénticas (debidas a la existencia de palabras *cognatas*) en ambos textos. Para intentar evitar que el coste sea proporcional al cuadrado de la longitud de los textos, el método realiza una búsqueda subóptima mediante un procedimiento de poda de los candidatos a alineamiento óptimo. Sin embargo, esta reducción es incompatible con la presencia de divergencias notables entre los textos.

El procedimiento de Owen et al. (2000) permite alinear textos en la misma lengua (*alineamientos intralingua*) buscando el alineamiento óptimo según una función de valoración que mide tanto la similitud entre las palabras —usando para el inglés la base de datos léxica WordNet (Miller, 1995)— como la distancia entre la palabra y su pareja (teniendo en cuenta, en su caso, la diferente longitud de los textos).

En este trabajo presentamos un método que se encuentra en un punto intermedio entre los procedimientos descritos: aunque la alineación se realiza con granularidad de caracteres, se aplica a idiomas relativamente próximos, lo que permite utilizar algoritmos sencillos y bien conocidos y modificarlos de forma simple para tener en cuenta grandes divergencias entre un texto y su traducción. Efectivamente, existen algoritmos eficientes (Wagner and Fischer, 1974) que construyen una secuencia mínima de operaciones de edición (borrados, inserciones y sustituciones) precisas para transformar un texto en otro.

El mayor problema de dichos algoritmos es que requieren un tiempo que crece cuadráticamente con la longitud de los textos que se quiere alinear, lo que hace inviable, salvo que se realice alguna aproximación, alinear más de unos pocos párrafos o, a lo sumo, páginas. La aproximación más usual consiste en restringir la búsqueda de posibles alineamientos a un cierto entorno: sólo pueden alinearse palabras que ocupan posiciones similares en sus textos respectivos. Dado que los textos son sistemáticamente más largos en algunos idiomas y que, en algunas traducciones, se producen desviaciones sistemáticas (por ejemplo, comentarios insertados por el traductor), por posición similar debe entenderse aquella que yace en el entorno de la diagonal de la tabla que se obtiene al representar cada texto en uno de los ejes, tal y como en la figura 1 se muestra de forma simplificada (textos de una sola palabra).

En este trabajo, hemos implementado un algoritmo que utiliza una ventana móvil y que se describe en la sección siguiente. Este procedimiento permite, en principio, tratar desviaciones arbitrariamente grandes respecto a la diagonal. Sin embargo, como veremos, es preciso realizar alguna modificación adicional para resolver las divergencias más grandes.

## 2 Un alineador con coste lineal basado en la distancia de edición

El cálculo del conjunto mínimo de operaciones de edición necesario para transformar un texto *x* en otro *y* es equivalente a la búsqueda de un camino de longitud mínima en

una tabla como la representada en la figura 1. Cada movimiento vertical puede interpretarse como una inserción, cada movimiento horizontal como un borrado y cada movimiento diagonal puede interpretarse como una sustitución (si los caracteres son distintos) o un alineamiento (si los caracteres son iguales). Por ello, en la tabla, cada *I* representa una inserción, cada *B* un borrado, cada *S* una sustitución de un carácter por otro y cada *A* el alineamiento de dos caracteres idénticos. El camino, que puede describirse mediante la secuencia de operaciones *AASSIABBBABBAA*, contiene 6

| | a | r | c | h | i | e | p | i | s | c | o | p | o |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | A | | | | | | | | | | | | |
| r | | A | | | | | | | | | | | |
| z | | | S | | | | | | | | | | |
| o | | | | S | | | | | | | | | |
| b | | | | I | | | | | | | | | |
| i | | | | | A | B | B | B | | | | | |
| s | | | | | | | | | A | B | B | | |
| p | | | | | | | | | | | | A | |
| o | | | | | | | | | | | | | A |

**Figura 1**: Un camino de distancia mínima para la transformación de la palabra "archiepiscopo" en "arzobispo" mediante 8 operaciones de edición. Los alineamientos generados están marcados con A.

alineamientos y 8 operaciones de edición. La búsqueda de este camino puede realizarse de forma bastante eficiente mediante algoritmos de programación dinámica (Cormen et al., 1992). Estos algoritmos se basan en que el camino óptimo para llegar a la posición (*i*, *j*) puede deducirse de los caminos óptimos para llegar a (*i*-1, *j*), (*i*, *j* -1) y (*i*-1, *j*-1) y construyen el camino mínimo para llegar a cada posición de la tabla por lo que el tiempo que se requiere para el cómputo es proporcional a $|x||y|$ siendo $|x|$ e $|y|$ los tamaños respectivos de los textos. Este crecimiento cuadrático es incompatible con la ejecución del algoritmo en un tiempo razonable si los textos que se quiere comparar son medianamente extensos (por ejemplo, varias páginas de texto). Por ello, es habitual restringir la búsqueda a un cierto entorno de la diagonal de la tabla, de forma que el coste temporal es entonces proporcional a $\max\{|x|,|y|\}$ y factible en la práctica para libros completos (Casacuberta and Vidal, 1987, página 74).

En nuestro caso hemos implementado una variante de esta idea que permite, en principio, alinear textos correctamente incluso si la distribución de las omisiones e inserciones no es homogénea a lo largo del bitexto (lo que puede alejar demasiado el camino óptimo de la diagonal como para que el procedimiento anterior lo encuentre). El procedimiento consiste en calcular el camino óptimo para un fragmento inicial del bitexto de *n* caracteres de cada texto (llamaremos *tamaño del buffer* al valor *n*) y considerar como válida sólo la primera parte de este camino (por ejemplo, sin sobrepasar los *m* primeros caracteres del buffer). La justificación intuitiva de esta decisión es que los alineamientos del final de este fragmento pueden ser incorrectos porque no se ha tenido en cuenta cómo continúa el bitexto. Supongamos que las operaciones dadas por válidas permiten transformar los *m* primeros caracteres del texto *x* en los *k* primeros caracteres de *y*; a continuación, se repite el proceso empezando por las posiciones *m*+1 y *k*+1 del bitexto. El algoritmo aparece representado gráficamente

en la figura 2. En la figura 3 puede verse el resultado de la aplicación del algoritmo a un fragmento de un bitexto latino-castellano.

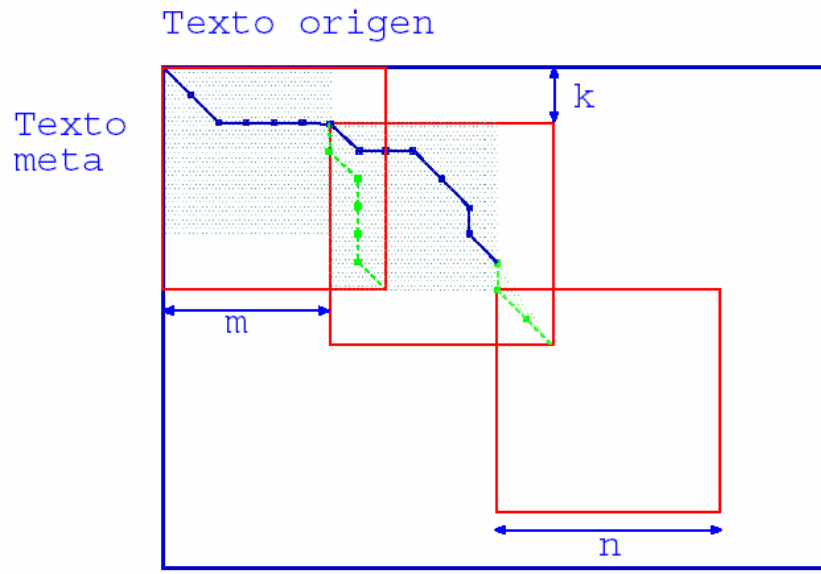


**Figura 2**: Representación gráfica del procedimiento de la ventana móvil para reducir el coste del cálculo de las operaciones de edición. Cada cuadrado de tamaño *n* representa una ventana de cómputo y los cuadrados sombreados de tamaño *m* contienen la parte del camino validada (en color más oscuro).

| |
|---|
| **Nihil** ita humano generi **nocere** consuevit -sicut **magnus**<br>*Nada tanto solió dañar a los hombres -según dice el gran* |
| inquit Chrysostomus-, **ut** est amicitiam contempnere, nec eam<br>*Crisóstomo- como el menospreciar la* |
| magno cum studio et tota observatione servare,<br>*amistad y no guardarla con gran afán y dedicación completa* |
| sicuti, e contra, nihil est **quod ita** res humana<br>*así como, por el contrario, no hay nada que tan bién* |
| **moderatur ac dirigat**, ut hanc omnibus viribus prosequi.<br>*modere y dirija los asuntos humanos como el promoverla* |
| Quod profecto Christus **insinuans**<br>*con todas las fuerzas. Es exactamente lo mismo que Cristo insinuaba* |
| **aiebat**: **Si duo** ex vobis consenserit in unum, quidquid<br>*diciendo: Si dos de vosotros se ponen de acuerdo* |
| petierint accipient.<br>*para pedir algo, lo conseguirán* |

**Figura 3**: Fragmento de la salida del algoritmo de alineamiento. En negrita aparecen las palabras alienadas perfectamente.

# 3 Resultados

El algoritmo descrito en la sección anterior ha sido utilizado para alinear un texto en latín medieval (*Lumen ad revelationem gentium*, escrito por Alonso de Oropesa en 1465 y disponible en la Biblioteca Virtual Miguel de Cervantes, http://cervantesvirtual.com/) con su correspondiente traducción al castellano actual (Luis A. Díaz y Díaz). El único preprocesamiento realizado en los textos consistió en reducir los espacios en blanco múltiples a uno sólo y tratar todos los caracteres como si hubieran sido escritos con minúsculas, ya que el traductor no siempre ha respetado el aspecto original. El texto contiene unas 15000 líneas, más de 160000 palabras y más de un millón de caracteres.

Para evaluar la calidad del alineamiento sin tener que recurrir a expertos, hemos representado en una gráfica (figura 4) la distancia de edición entre los fragmentos alineados (normalizada con la suma de sus longitudes $m + k$) a lo largo del libro. El tamaño del buffer utilizado en este ejemplo es de 1000 caracteres y los fragmentos validados contienen los primeros 500 caracteres. Es destacable que la distancia de edición es muy variable pero presenta zonas de valor significativamente más alto que otras.

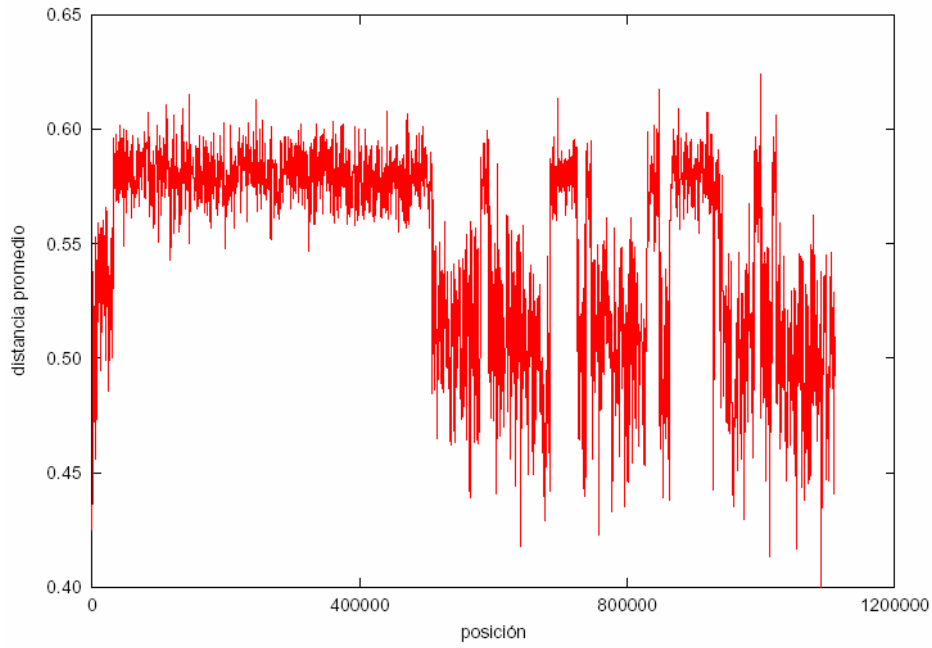


**Figura 4**: Distancia de edición normalizada entre los fragmentos alineados en función de su posición en el bitexto ($n$ = 1000 y $m$ = 500).

Para entender el significado de estas variaciones hemos calculado la distancia promedio entre fragmentos en dos casos extremos:

1. un fragmento del texto latino y otro en castellano sin ninguna relación entre sí (el resultado es un alineamiento aleatorio);
2. un fragmento del texto latino y su correspondiente traducción al castellano (con tamaño inferior a *n* con lo que el alineamiento es el óptimo que puede obtenerse usando distancias de edición).

En el primer caso, la distancia promedio es aproximadamente 0,58 y en el segundo oscila entre 0,5 y 0,52; estos valores se corresponden claramente con los observados en la gráfica 4 y apoyan la interpretación de que las zonas de valor bajos se corresponden con fragmentos correctamente alineados y las mesetas con fragmentos mal alineados. Tras examinar los textos y el alineamiento obtenido se observa que las zonas de distancia elevada contienen divergencias notables entre el texto original y la traducción. Por ejemplo, la primera meseta corresponde a la reiteración en la versión latina del prefacio (aproximadamente 30000 caracteres), repetición ausente en la versión castellana que contiene, en cambio, algunos comentarios del traductor al respecto (unos 7000 caracteres). La meseta se extiende más allá en el bitexto debido a que la diferencia entre los textos (más de 20000 caracteres) es superior al tamaño del buffer empleado y los alineamientos generados a partir de ese momento son aleatorios sin que el algoritmo pueda vislumbrar el punto de reenganche. Por tanto, la vuelta a alineamientos correctos debe interpretarse, en gran medida, como una afortunada coincidencia.

No obstante, importa destacar que el algoritmo no sólo permite alinear correctamente la mayor parte del texto sino que además permite reconocer aquellas zonas donde las divergencias entre un texto y otro son demasiado grandes como para ser debidas a las diferencias léxicas entre la lengua origen y la meta y, por tanto, debe suponerse que el alineamiento es incorrecto.

Cabe esperar que tamaños del buffer superiores mejoren los resultados. Esto es lo que parece confirmar la figura 5, donde puede observarse que la extensión de las mesetas ha disminuido. Sin embargo, en algún caso (véase la primera de ellas), su extensión sigue siendo aún mayor que la de la divergencia local entre los textos. La figura 6 muestra la influencia del tamaño del buffer para producir el alineamiento correcto. Aunque es posible que tamaños aún mayores produzcan mejores resultados, el coste temporal hace inviable llegar a tamaños mucho mayores.

Por tanto, hemos planteado el siguiente objetivo: conseguir un mejor alineamiento sin aumentar el tamaño de buffer, al menos en promedio. La estrategia planteada consiste en que cuando se observa un aumento persistente y significativo de la distancia entre los textos se exploran ventanas más grandes sin escribir nada en la salida. Cuando se encuentra un nuevo punto donde la distancia promedio es baja se vuelve al procedimiento normal.

En concreto, cada vez que la distancia promedio sube por encima de un umbral de forma consistente (superior a 0.55 en dos de las tres últimas iteraciones), se exploran además dos bandas del bitexto de tamaño $n \times N$ y $N \times n$ divididas en fragmentos de longitud $n/2$, siendo $N$ un múltiplo grande de $n$: en nuestro caso, dado que queremos detectar omisiones de capítulos enteros, hemos tomado $N = 50n$. De esta forma, se intenta alinear:

1. los primeros $n$ caracteres de $x$ con alguno de los bloques de $n/2$ caracteres consecutivos de $y$;
2. los primeros $n$ caracteres de $y$ con alguno de los bloques de $n/2$ caracteres consecutivos de $x$;

3. los primeros *n* caracteres de *x* con los *n* primeros de *y*.

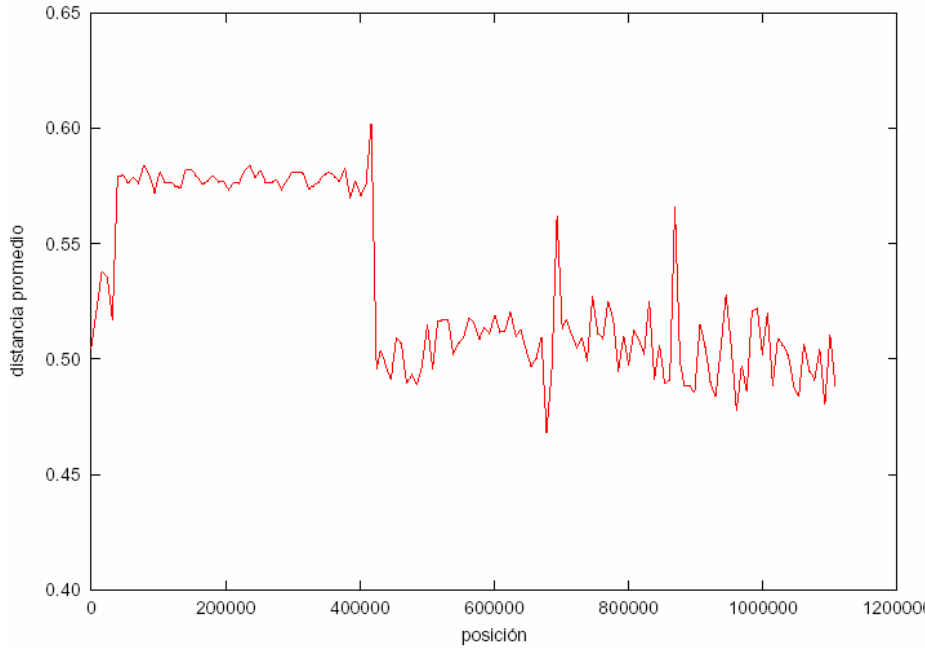


**Figura 5**: Distancia de edición normalizada entre los fragmentos alineados en función de su posición en el bitexto (*n* = 16000 y *m* = 8000).

Si alguna de las dos primeras opciones es sensiblemente mejor que la tercera (en nuestro caso, si la distancia promedio es menor que 0.550) se asume que se ha producido una inserción o borrado de bloques completos.

El algoritmo descrito en el párrafo anterior permite obtener el resultado representado en la gráfica 7. En ella, se observa que han desaparecido las mesetas, lo que indica que el algoritmo ha sido capaz de detectar las principales omisiones presentes en el texto (visibles aún como picos locales en la gráfica) y ha podido proseguir el alineamiento sin perder la correspondencia entre los textos. La distancia media obtenida ha sido 0,51, cercana al mejor valor esperable.

## 4 Conclusiones

En este trabajo hemos presentado un algoritmo basado en la distancia de edición para la presentación sinóptica de textos bilingües. Este algoritmo es rápido (en parte debido a su coste lineal) y su implementación muy sencilla, lo que permite su adaptación para tratar bitextos con divergencias muy grandes.

Otra mejora que estamos investigando es utilizar las similitudes entre los idiomas, bien entre caracteres (por ejemplo, una “f” se corresponde más probablemente con una “h” que con una “a”) bien entre palabras (utilizando diccionarios de traducción parciales o asociaciones entre grupos como “-us”/“-o”, “-ct-”/“ch” etc.).

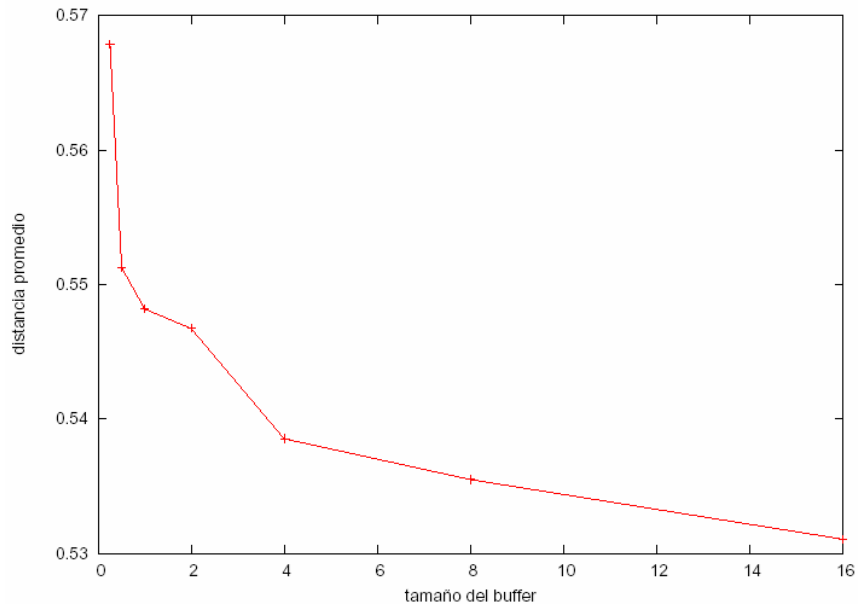


**Figura 6**: Distancia de edición promedio en función del tamaño del buffer utilizado.

Por último, estamos explorando el uso de este algoritmo para alinear bitextos de lenguas muy diferentes utilizando como referencia una traducción automática que,

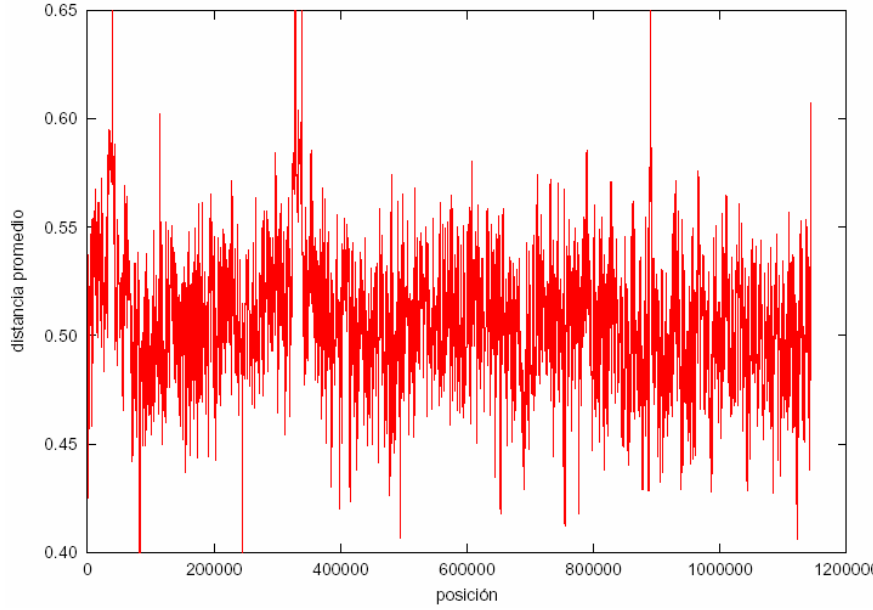


**Figura 7**: Distancia de edición normalizada entre los fragmentos alineados en función de su posición en el bitexto usando corrección automática de divergencias.

aunque no ofrece una calidad perfecta, permite obtener un texto intermedio con mayor similitud al traducido que el original.

## 5 Agradecimientos

Este trabajo ha sido financiado por la Comisión Interministerial de Ciencia y Tecnología (TIC2000-1599-C02-02).

## Bibliografía

Brown, P., Cocke, J., Pietra, S. D., Pietra, V. J. D., Jelinek, F., La®erty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Casacuberta, F. and Vidal, E. (1987). *Reconocimiento automático del habla.* Marcombo.

Church, K. W. (1993). Char align: A program for aligning parallel texts at the character level. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1992). *Introduction to algorithms*. MIT Press and McGraw-Hill Book Company, 6th edition.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Owen, C. B., Ford, J., Makedon, F., Steinberg, T., and Metaxaki-Kossionides, C. (2000). Parallel text alignment. *Int. J. on Digital Libraries*, 3(1):100–114.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173.