DISTANCIA DEPENDIENTE DE LA SUBSECUENCIA COMÚN MÁS LARGA ENTRE CADENAS DE CARACTERES

M. DÍAZ, J. PÉREZ, O. SANTANA.
Departamento de Informática y Sistemas.
Universidad de Las Palmas de Gran Canaria.
Aptdo. 550, Las Palmas de Gran Canaria. España.

Resumen:

En este trabajo se demuestra que el valor definido por Santana y otros en [SR90] —basado en la longitud de la subsecuencia común más larga entre cadenas de caracteres al objeto de reducir el número de cadenas que han de someterse al cálculo de la Distancia de Levenshtein, *DL*— es una distancia en el espacio de cadenas de caracteres sobre un alfabeto, que en adelante se denotará por *DS*. Se corrobora experimentalmente que *DS* mejora las realizaciones de los esquemas de búsqueda decreciente y creciente sobre el índice donde se estructuran las componentes de la distancia invariante trasposicional, *DIT*, [SD87], [SD89] y [DS90].

0.— INTRODUCCIÓN.

En este trabajo se demuestra que el umbral establecido en [SR90] basado en la longitud de la subsecuencia común más larga entre cadenas de caracteres verifica las propiedades de distancia en el espacio de cadenas de caracteres sobre un alfabeto. Se comprueba experimentalmente que, frente a la longitud de la cadena de búsqueda, mejoran las realizaciones los esquemas de búsqueda decreciente y creciente sobre la estructura *DITE* —organiza las componentes de la distancia invariante trasposicional *DIT*— optimizados con la introducción de la poda recomendada en [SR90].

En la sección 1 se definen las distancias utilizadas: la Distancia de Levenshtein, *DL*, [LE66], evaluada por Wagner y Fischer [WF74], cuyo cálculo se optimiza en [UK83], y es posteriormente utilizado por [UK85], [GG86], [LV85] y [LV86]; la Distancia Invariante Trasposicional, *DIT*, cuyo costo es inferior a *DL* [SD87] y [SD89]; y la Distancia de Santana, *DS*, introducida como umbral en [SR90]. Se demuestra que *DS* es una distancia en el espacio de cadenas de caracteres sobre un alfabeto en la sección 2. En la sección 3 se describen la estructura *DITE* y los esquemas de búsqueda decreciente y creciente, [SD87], [SD89] y [DS89], con las optimizaciones recomendadas en [SR90]. Se discuten los resultados experimentales obtenidos utilizando un diccionario de 89655 palabras y se presentan las conclusiones en la sección 4.

1.— DISTANCIAS.

Sea X una cadena sobre un alfabeto $\{\alpha_1,...,\alpha_m\}$, y sea μ la cadena nula. $/X/$ denota la longitud de X.

Una operación de edición es el par $(\Phi,\Omega)\neq(\mu,\mu)$ donde Φ y Ω son cadenas de longitud menor o igual que uno, es decir, o son μ o son un único carácter. La cadena Y resulta de aplicar (Φ,Ω) a la cadena X, que se escribe $X\rightarrow Y$, si $X=\sigma\Phi\pi$ e $Y=\sigma\Omega\pi$.

Wagner y Fischer, [WF74], consideran tres tipos de operaciones de edición. (Φ,Ω) es una operación de sustitución si $\Phi\neq\mu$, $\Omega\neq\mu$ y $\Phi\neq\Omega$; es una operación de extracción si $\Omega=\mu$ y es una operación de inserción si $\Phi=\mu$.

Sea S una secuencia $S_1,S_2,...,S_n$ de operaciones de edición. Una derivación-S de X hasta Y es una secuencia de cadenas $X_0,X_1,...,X_n$ tal que $X=X_0$, $Y=X_n$ y $X_{j-1}\rightarrow X_j$ vía S_j $\forall j=1,...,n$.

1.1.— DISTANCIA DE LEVENSHTEIN, *DL*

Sea Γ una función arbitraria de costo que asigne a cada operación de edición (Φ,Ω) un número real no negativo $\Gamma(\Phi,\Omega)$. Se puede extender Γ a la secuencia S definiendo:

$$\Gamma(S) = \sum_{j=1}^{n}\Gamma(S_j) \;\; si\; n \geq 1 \;\; y \;\; \Gamma(S) = 0 \;\; si\; n = 0$$

Se denomina distancia de Levenshtein, $DL(X,Y)$, entre las cadenas X e Y, al mínimo costo de todas las secuencias de edición que transformen X en Y.

En este trabajo se ha seguido el criterio de que el costo de cualquier operación de edición se considera unitario.

1.2.— DISTANCIA INVARIANTE TRASPOSICIONAL, *DIT*

$$DIT(X,Y) = \frac{1}{2}\left(\sum_{i=1}^{m} abs(X_{\alpha i} - Y_{\alpha i}) + abs(|X|-|Y|)\right)$$

Donde X_{α_i} e Y_{α_i} son las frecuencias de aparición del carácter α_i en X y en Y respectivamente.

Se ha probado, [SD87], que $DIT(X,Y)\leq DL(X,Y)$ y que computacionalmente es menos costosa.

1.3.— DISTANCIA DE SANTANA, *DS*

Una cadena $C=c_1c_2...c_p$ es una subsecuencia de otra cadena $X=x_1x_2...x_t$ si y sólo si existe una aplicación *f* de *{1,2,...,p}* en *{1,2,...,t}* tal que *f* es una función monótona estrictamente creciente y si $f(i)=k$ entonces $c_i=x_k$.

Una cadena *C* es una subsecuencia común de dos cadenas *X* e *Y* si y sólo si *C* es una subsecuencia de *X* y *C* es subsecuencia de *Y*. El cálculo de la subsecuencia común más larga entre dos cadenas ha sido estudiado en [HY75].

Dadas dos cadenas *X* e *Y*, cuya Subsecuencia Común Más Larga es *SCML(X,Y)*. Se define la distancia *DS(X,Y)* como la diferencia entre la longitud mayor de ambas cadenas y la longitud de la subsecuencia común más larga, es decir:

$$DS(X,Y) = máximo(|X|, |Y|) - |SCML(X,Y)|$$

Se ha probado, [SR90], que: $DIT(X,Y) \leq DS(X,Y) \leq DL(X,Y)$ y que su costo computacional es inferior al de *DL*.

2.— *DS* ES UNA DISTANCIA.

a) $DS(X,Y)=0$ si y sólo si $X=Y$

$DS(X,Y) = 0 \Leftrightarrow máximo\{ |X|, |Y|\} - |SCML(X,Y)| = 0 \Leftrightarrow$

$\Leftrightarrow máximo\{ |X|, |Y|\} = |SCML(X,Y)| \Leftrightarrow$

$\Leftrightarrow |X| = |Y| = |SCML(X,Y)| \Leftrightarrow X=Y$

b) $DS(X,Y) = DS(Y,X)$

Por la propia definición.

c) $DS(X,Y) + DS(Y,Z) \geq DS(X,Z)$

$DS(X,Y) + DS(Y,Z) = máximo\{ |X|, |Y|\} - |SCML(X,Y)| +$

$+ máximo\{ |Y|, |Z|\} - |SCML(Y,Z)| =$

$= máximo\{ |X|, |Y|\} + máximo\{ |Y|, |Z|\} - |SCML(X,Y,Z)| -$

$- (|SCML(X,Y)| + |SCML(Y,Z)| - |SCML(X,Y,Z)|)$

como:

$|SCML(X,Y)| + |SCML(Y,Z)| - |SCML(X,Y,Z)| \leq |Y|$

se tiene que:

$DS(X,Y) + DS(Y,Z) \geq máximo\{ |X|, |Y|\} + máximo\{ |Y|, |Z|\} -$

$- |SCML(X,Y,Z)| - |Y|$

dado que:

$máximo\{ |X|, |Y|\} + máximo\{ |Y|, |Z|\} - |Y| \geq máximo\{ |X|, |Z|\}$

se tiene que:

$DS(X,Y) + DS(Y,Z) \geq máximo\{ |X|, |Z|\} - |SCML(X,Y,Z)|$

y como: $|SCML(X,Z)| \geq |SCML(X,Y,Z)| \geq 0$

entonces:

$$DS(X,Y) + DS(Y,Z) \geq máximo\{|X|, |Z|\} - |SCML(X,Z)| = DS(X,Z).$$

3.— ESTRUCTURA *DITE* Y ESQUEMAS DE BÚSQUEDA DECRECIENTE Y CRECIENTE.

3.1.— ESTRUCTURA *DITE*

El diccionario se encuentra organizado como un árbol, [SD87], [SD89] y [DS90], en el cual se estructuran las componentes que intervienen en el cálculo de *DIT*.

Tiene un nodo raíz que discrimina por longitudes de cadenas; el encadenamiento asociado a cada longitud señala a una parte-árbol en la que, cada nodo contiene un carácter α y discrimina por las diferentes frecuencias *f* con que aparece tal carácter en el subdiccionario correspondiente, α se elige de tal forma que minimice la suma de los cuadrados de los cardinales de los subdiccionarios que genera; cuando una rama de la parte-árbol ya no discrimina se pasa a la parte-cadena formada por listas encadenadas constituidas por los restantes pares α/*f*, siendo *f* la frecuencia con que aparece ese carácter α en las cadenas sinónimas que, agrupadas en un nodo-*SIT*, penden al final de cada una de esas listas.

3.2.— ESQUEMAS DE BÚSQUEDA DECRECIENTE Y CRECIENTE

En ambos esquemas:

* Se dispone de un umbral igual a la distancia de Levenshtein mínima actual, *DLM*, de tal forma que sólo se calcula la *DL* entre la cadena de búsqueda y aquella cadena del diccionario cuya $DIT \leq DLM$, debido a la relación existente entre *DIT* y *DL*.
* En cada nodo se utiliza la denominada poda *PP*, [SR90], consistente en estimar el valor mínimo alcanzable por *DIT* en el camino que va desde la raíz hasta cualquier nodo-*SIT* que penda de ese nodo y utilizarlo para acortar el recorrido por la estructura si supera el umbral *DLM*.
* El recorrido de la parte-cadena es secuencial.
* Si se accede a un nodo-*SIT* se evalúan las *DL* entre la cadena de búsqueda y todos las de ese nodo.

— DECRECIENTE

A lo largo del proceso, partiendo del nodo raíz, tanto *DLM* como la respuesta se actualizan cada vez que se obtiene una $DL \leq DLM$. El valor inicial de *DLM* es infinito.

Durante el recorrido en el índice la exploración de alternativas en los nodos discriminantes se lleva a cabo en vaivén en torno al valor correspondiente del discriminante en la cadena de búsqueda.

— CRECIENTE

En este esquema se recorre la estructura partiendo del nodo raíz y con radio de búsqueda *DLM=0*; si no se encuentra respuesta se aumenta en una unidad el radio de búsqueda y se comienza la exploración, nuevamente desde el nodo raíz, hasta que al finalizar un recorrido exista respuesta. Debido a que en cada fase el radio de búsqueda es menor o igual que el de la respuesta, es por lo que el umbral *DLM* no es adaptativo; por tanto, la exploración en cada nodo discriminante se lleva a cabo en un solo sentido, mas rápida, ya que está determinada exclusivamente por su posición y el valor de *DLM*, ambos fijos en cada recorrido.

3.3.— ESQUEMAS DE BÚSQUEDA DECRECIENTE Y CRECIENTE CON INTERVENCIÓN DE *DS*

La única diferencia respecto a los esquemas anteriores es que cuando se accede a un nodo-*SIT* se evalúa *DS* entre la cadena de búsqueda y todas las de ese nodo y tan sólo para aquellos sinónimos *DIT* en los que no se supere el valor actual de *DLM* es necesario evaluar la *DL*, en virtud de la relación $DS \leq DL$, [SR90], con la consiguiente reducción en número de *DL* evaluadas.

4.— ESTUDIO EXPERIMENTAL Y CONCLUSIONES.

Al objeto de corroborar y completar los estudios experimentales llevados a cabo en [SR90], se utiliza un diccionario de una cardinalidad significativamente superior, compuesto por 89655 palabras. Para las pruebas se ha utilizado un ordenador HP-9000. Se construyen ficheros que van a ser utilizados como argumento de búsqueda de las más similares, para ello se selecciona al azar una palabra del diccionario y se transforma aplicándole diversas operaciones de edición, elegidas aleatoriamente, hasta obtener una cadena distorsionada con una *DL* determinada respecto a la correcta. *DL* toma los valores comprendidos entre uno y la mitad de la longitud de la palabra correcta.

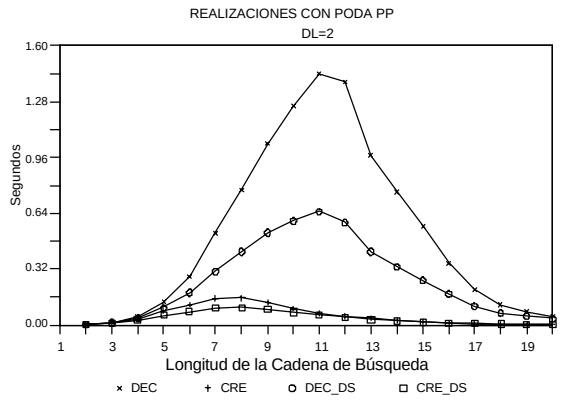

Figura 1

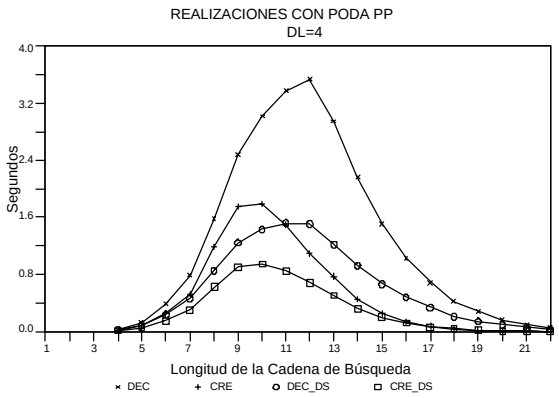

Figura 2

Las figuras 1 y 2, muestran un descenso acusado en los tiempos de búsqueda de los esquemas con poda *PP* al hacer intervenir *DS*, esta diferencia es más importante en el esquema decreciente.

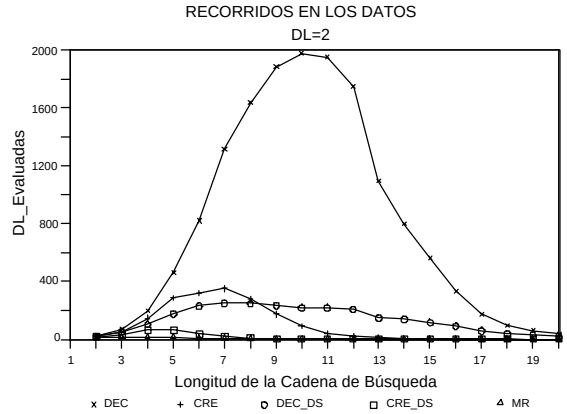

Figura 3

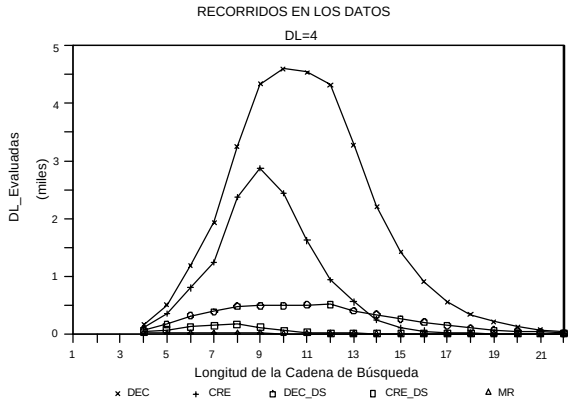

Figura 4

La mejor realización de ambos esquemas al incluir *DS* se debe exclusivamente al ahorro del número de *DL* evaluadas, figuras 3 y 4, ya que no afecta en absoluto al recorrido en el índice. Obsérvese en las figuras 3 y 4, como el número de *DL* evaluadas en ambos esquemas se acerca, bastante más en el creciente aunque más rápidamente en el decreciente, al tamaño o multiplicidad de la respuesta, *MR*. *MR* constituye el nivel mínimo teórico alcanzable para el número de *DL* a calcular.

Los resultados de estos experimentos vienen a corroborar que la utilización de *DS* es rentable y presentan un comportamiento muy similar sobre un diccionario de mucha mayor cardinalidad que el usado en [SR90].

La principal aportación de este trabajo es la demostración de que *DS* es una distancia, esto permite su utilización en estructuras de Burkhard-Keller, [SP88], [SP89] y [SP90]. Se recomienda para trabajos posteriores el desarrollo de una estructura específica para que pueda intervenir en forma más eficaz y mejorar así las realizaciones en las búsquedas.

Referencias:

[DS90] DÍAZ, M.; SANTANA, O.; RODRÍGUEZ, J.C.:"Búsqueda de las Cadenas Mas Similares: Esquema Decreciente con Radio de Búsqueda Ascendente, Esquema Creciente". XVI Conferencia Latinoamericana de Informática, Vol. I, 90/97, (1990).

[GG86] GALIL, Z.; GIANCARLO, R.:"Improved String Matching with k Mismatches", SIGACT News, 17, 4, 52/54, (1986).

[HI75] HIRSCHBERG, D. S.:"A Linear Space Algorithm for Computing Maximal Common Subsequences", Communications of the ACM, Vol. 18, 341/353, (1975).

[LE66] LEVENSHTEIN, V. I.: "Binary Codes Capable of Correcting, Insertions and Reversals". Soviet Phys. Dokl. 10, 707/710, (1966).

[LV85a] LANDAU, G. M.; VISHKIN, U.: "Efficient String Matching in the Presence of Errors". Proc. 26th IEEE FOSC, 126/136, (1985).

[LV85b] LANDAU, G. M.; VISHKIN, U.: "Efficient String Matching with k Differences". TR-36/85, Departament of Computer Science, Tel Aviv Univ., Submitted for Journal Publication, (1985).

[LV86a] LANDAU, G. M.; VISHKIN, U.: "Efficient String Matching with k Mismatches". Theoretical Computer Science, 43, 239/249, (1986).

[LV86b] LANDAU, G. M.; VISHKIN, U.; NUSSINOV, R.: "An Efficient String Matching Algorithm with k Differences for Nucleotide and Amino Acid Sequences". Nucleic Acid Research 14 (1), 31/46, (1986).

[SD87] SANTANA, O.; DÍAZ, M.; MAYOR, O.; REYES, J.: "Esquemas y Estructura para la Búsqueda de las Palabras Más Similares a una Dada". XIII Conferencia Latinoamericana de Informática, Vol. II, 1169/1189, (1987).

[SD89] SANTANA, O.; DÍAZ, M.; DUQUE, J.D.; RODRÍGUEZ, G.: "Estructuración de las Componentes de la Distancia Invariante Trasposicional, DIT, con Compartición de la Zona No-Discriminante en la Búsqueda de las Cadenas Más Similares". XV Conferencia Latinoamericana de Informática, Vol. II, 335/341, (1989).

[SP88] SANTANA, O.; PÉREZ, J.; LÓPEZ G.; RODRÍGUEZ, G.: "La estructura de Burkhard-Keller en la búsqueda de las cadenas más similares a una dada". XIV Conferencia Latinoamericana de Informática, Buenos Aires, Argentina, (1988).

[SP89a] SANTANA, O.; PÉREZ, J.; HERNÁNDEZ, Z; RODRÍGUEZ H., G.: "Sharing the Components of Transposition-Invariant Distance, DIT, on DIT-organized Burkhard-Keller Structure in Searches for Best Matching Strings". IEEE INTERNATIONAL WORKSHOP ON TOOLS FOR ARTIFICIAL INTELLIGENCE "Achitectures, Languages & Algorithms", Fairfax, Virginia, U.S.A., October (1989).

[SP89b] SANTANA, O.; PÉREZ, J.; ESPINO, M.; RODRÍGUEZ, J.C.: "Referencias Distanciales de Levenshtein en la Estructura de Burkhard-Keller Organizada según la Distancia Invariante Transposicional. Parte I". Actas de la XV Conferencia Latinoamericana de Informática, Santiago de Chile, Julio, Vol. II, 327/334, (1989).

[SP90] SANTANA, O.; PÉREZ, J.; RODRÍGUEZ, J.C.: "Increasing Radius Search Schemes for the Most Similar Strings on the Burkhard-Keller Tree". Cybernetics and Systems: An International Journal, 21: 167-180, (1990).

[SR90] SANTANA, O.; RODRÍGUEZ, J. C.; DÍAZ, M.: "Búsqueda de las Cadenas Más Similares: Incidencia de la Subsecuencia Común Más Larga en los Esquemas Decreciente y Creciente". XVI Conferencia Latinoamericana de Informática, Vol. I, 98/104, (1990).

[UK83] UKKONEN,E.:"On Approximate String Matching". Proc. Int. Conf. Found. Comp. Theor., Lecture Notes in Computer Science 158, Springer-Verlag, 487/495, (1983).

[UK85] UKKONEN, E.:"Finding Approximate Pattern in Strings". J. of Algorithms, 6, 132/137, (1985).

[WF74] WAGNER, R.A.; FISCHER, M.J.:"The String-to-String Correction Problem". JACM, 21 (1), 168/173, (1974).