METODOS DE COMPRESION BIGRAMATICA POSICIONAL

Fco. Javier Gurruchaga Vázquez
Colaboradores: *Félix Ares de Blas, Julio González Abascal, J. Luis García de Madinabeitia*
Facultad de Informática de San Sebastián
San Sebastián - *España*

INTRODUCCION

En nuestros días, se podría decir que la informática forma parte de nuestras vidas. Casi el cien por cien de la información que manejamos y que día a día pasa por nuestras manos, ha sufrido algún proceso de tipo informático.

Imprentas, agencias de información, en general cualquier tipo de empresas, están dando un profundo cambio en sus métodos, estructuras y formas de trabajo, debido a la introducción de los ordenadores, ya que con ellos tenemos información rápida, de fácil manejo y con un menor costo.

Esto nos está llevando, por otro lado, a un problema de tipo informático como es el del almacenamiento de tanta información dentro del ordenador, que se refleja sobre todo en tres puntos a tener en cuenta:

- Costo de almacenamiento.
- Manejabilidad de los datos.
- Búsqueda de los datos.

Además de todo esto, debemos de tener en cuenta que los ficheros o bases de datos, debido a su diseño inicial y a las posteriores ampliaciones, llega un momento en el que dicho fichero tendrá información repetida y espacios vacíos (sin información).

Ante todos estos problemas que hemos de tener en cuenta a la hora de poner en marcha una infraestructura de información, debemos intentar optimizar el tamaño de los diferentes ficheros mediante una codificación que nos lo permita.

A esta reducción mediante una codificación apropiada, se le da el nombre de COMPRESION. La compresión de ficheros nos proporciona una serie de ventajas tales como:

- Disminución de los volúmenes de información.
- Menor cantidad de dispositivos necesarios para el almacenamiento de datos.
- Mayor rapidez en la búsqueda y el proceso de los datos.
- Mayor rapidez de transferencia entre periféricos y memoria, así como, entre terminal y ordenador, lo que nos da mayor economía.

Asímismo, la compresión de ficheros tiene una serie de inconvenientes que requerirán un estudio a la hora de su implantación:

- Obtención, la primera vez, de los diccionarios de compresión.
- Proceso de compresión-descompresión.

Por último, este estudio sólo trata de mejorar un método de compresión que es el bigramático, haciéndolo aún más rentable a la hora de almacenar información en un ordenador.

DEFINICIONES USADAS EN ESTE TRABAJO

TEXTO FUENTE: Texto inicial del que se parte.

CARACTER: Es cada elemento de un alfabeto fuente.

ALFABETO FUENTE (A_f): Conjunto de caracteres con los que podemos formar un texto fuente. $A_f = a_1, a_2, \ldots\ldots, a_n$

TEXTO CODIFICADO: Conjunto de símbolos resultantes de aplicar el método de compresión a un texto fuente.

SIMBOLO o PALABRA CODIGO: Es cada elemento del alfabeto codificador.

ALFABETO CODIFICADOR (A_c): Conjunto de símbolos con los que se puede formar un texto codificado. En este alfabeto se incluyen los caracteres del alfabeto fuente más los símbolos que sustituyen a los caracteres fuente.

$$A_c = b_1, b_2, \ldots\ldots, b_n, \ldots\ldots, b_m$$

PALABRA FUENTE: Serie de caracteres del alfabeto fuente limitados a ambos lados por blancos.

$$\text{Palabra fuente} = \not\beta a_1 \_ a_5 \_ a_7 \not\beta$$

PALABRA CODIFICADA: Serie de símbolos limitados a ambos lados por blancos, de la forma:

- Un primer símbolo del alfabeto codificador que sustituye a caracteres iniciales de una palabra del texto fuente.
- Puede existir un segundo símbolo del alfabeto codificador que sustituye a caracteres finales de una palabra del texto fuente.
- Pueden existir uno o más símbolos del alfabeto codificador que sustituyen a caracteres intermedios de una palabra del texto fuente.

$$\text{Palabra codificada} = \not\beta b_{2i} \not\beta$$
$$\text{Palabra codificada} = \not\beta b_{2i} \_ b_{3f} \not\beta$$
$$\text{Palabra codificada} = \not\beta b_{2i} \_ b_{3m} \_ b_{3f} \not\beta$$
$$\text{Palabra codificada} = \not\beta b_{2i} \_ b_{3m} \_ \ldots\ldots\ldots \_ b_{2m} \_ b_{3f} \not\beta$$

BLOQUE MULTILETRA FINAL (BMF): Conjunto de caracteres fuente que se sustituyen por un solo símbolo del alfa-

beto codificador.

FRECUENCIA ABSOLUTA DE BMF (FABMF): Número de veces que un BMF aparece en el texto fuente.

FRECUENCIA RELATIVA DE BMF (FRBMF):

$$FRBMF = \frac{FABMF}{\Sigma\ FABMF + RESTO}$$

RESTO: Caracteres de las palabras fuente que no han sido sustituídos.

COMPRESION ABSOLUTA (CA): Es la frecuencia absoluta de un BMF multiplicada por el número de caracteres menos uno del BMF.

$$CA = FABMF \times (\text{número caracteres BMF} - 1)$$

COMPRESION RELATIVA (CR):

$$CR = \frac{\text{núm. bits texto fuente} - \text{núm. bits texto codificado}}{\text{núm. bits texto fuente}}$$

NUMERO DE BITS POR LETRA A LA SALIDA:

$$\text{núm. bits/letra salida} = \frac{\text{núm. bits texto codificado}}{\text{núm. caracteres texto fuente}}$$

RENDIMIENTO DE UN BMF (RBMF): Es la frecuencia absoluta del BMF multiplicada por el número de caracteres del BMF menos uno.

$$RBMF = FABMF \times (\text{número caracteres BMF} - 1)$$

METODO DE COMPRESION BIGRAMATICA POSICIONAL

Podemos definir la bigramática posicional como una técnica para obtener cadenas de bigramas en un texto atendiendo a la posición que ocupan estos bigramas dentro de una palabra.

Diremos que la compresión bigramática posicional toma como base el método de compresión bigramática tradicional para crear otro sistema muy similar que además de tener en cuenta todas las características de la bigramática tradicional, tiene en cuenta la posición de los caracteres dentro de las palabras de un texto escrito en un idioma natural.

Podríamos preguntarnos, porqué añadir el concepto de posicionalidad. La respuesta se desprende de la tesis doctoral

de Félix Ares [ARES-83] en la que se demuestra que la entropía de un carácter varía según la posición que éste ocupe dentro de la palabra, por lo que podemos utilizar estas diferencias a la hora de eliminar redundancias.

Todas la experiencias que hemos realizado con diversos textos han sido siempre sin intentar introducir el carácter blanco, separador de palabras, dentro de ninguno de los bloques de bigramas -BMFs-. El motivo de tomar esta decisión ha sido el poder comparar nuestros resultados con trabajos previos [GARCIA-80].

Al tener en cuenta la posicionalidad se observa que las palabras se pueden considerar compuestas de tres bloques de símbolos: iniciales, finales e intermedios. Por ejemplo, FERNANDEZ se podría descomponer como:

- Bloque inicial ..... : FERN
- Bloque final ..... : EZ
- Primer bloque intermedio ..... : AN
- Segundo bloque intermedio ..... : D

Por lo tanto, una palabra código dependiendo de la posición que ocupe dentro de la palabra fuente tendrá distintos significados.

COMPARACION DE RESULTADOS ENTRE LA BIGRAMATICA Y LA BIGRAMATICA POSICIONAL.

Para comparar los resultados se ha tomado, como se ha dicho anteriormente, el texto utilizado por E. García Camarero y L. Bengoechea Martínez en su estudio "Un método para la compresión de textos" [GARCIA-80]. Ello ha sido debido a que éste es uno de los pocos trabajos publicados sobre el castellano y la codificación bigramática.

El texto fuente está formado por una secuencia de apellidos castellanos que tiene 14.266 caracteres y su alfabeto fuente ha sido: (A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z - Ø). Como alfabeto codificador se tomaron 125 símbolos. El número de blancos no reducibles dentro del texto es de 1.921.

Teniendo en cuenta lo expuesto anteriormente y utilizando un algoritmo de compresión bigramática, la longitud final del texto codificado fue de 8.356 símbolos. Con lo que se consiguió una reducción del 41,43% medida en caracteres.

En nuestro estudio se utiliza el mismo texto fuente aunque por errores mecanográficos de transcripción del texto, éste ha quedado con 14.240 caracteres. El alfabeto fuente es el mismo. Como alfabeto codificador se tienen las tres tablas:

- Una con BMFs iniciales
- Otra con BMFs finales, y
- Otra con BMFs intermedios.

Tomamos 128 símbolos para cada una de las tablas anteriores.

Para codificar los 28 caracteres del alfabeto fuente nos bastarían 5 bits, pero tomamos la decisión de ampliarlo a 6 bits, para dar la oportunidad de utilizar un alfabeto fuente más amplio. Asimismo, se va a utilizar un código de 7 bits para cada símbolo del alfabeto codificador, ya que, cada tabla va a contener 128 símbolos, de los que los 29 primeros serán los correspondientes a los caracteres del alfabeto fuente.

Con todo lo expuesto se presentan los resultados en la Figura I.

Fijémonos en la columna 11 de dicha figura. Vemos que nuestro método da una mejora del 6% sobre el método bigramático puro. Si nos fijamos en la columna 6 que nos da la compresión relativa en bits, vemos que la diferencia aumenta siendo del 7%. Esta diferencia todavía sería mayor si tuviéramos en cuenta que a todos los símbolos del texto codificado les hemos dado 7 bits, cuando realmente los símbolos denominados como RESTO podrían haberse codificado con sólo 6 bits.

También tenemos que tener en cuenta que se ha partido de un texto fuente con 1.921 blancos no reducibles, con lo que el número de caracteres reducibles en el texto fuente sería de 12.319 (14.240 - 1.921).

Si ahora efectuamos los cálculos sólo sobre los caracteres reducibles, se obtiene:

Aplicando la fórmula de compresión en caracteres:

$$CR = \frac{(14.240 - 1.921) - (7.486 - 1.921)}{(14.240 - 1.921)} = 0,5482$$

Según la bigramática posicional

$$CR = \frac{(14.266 - 1.921) - (8.356 - 1.921)}{(14.266 - 1.921)} = 0,4787$$

Según la bigramática pura

Aplicando la fórmula de compresión en bits

$$CR = \frac{(14.240 - 1.921) \times 6 - (7.486 - 1.921) \times 7}{(14.240 - 1.921) \times 6} = 0,4729$$

Según la bigramática posicional

$$CR = \frac{(14.266 - 1.921) \times 6 - (8.356 - 1.921) \times 7}{(14.266 - 1.921) \times 6} = 0,3918$$

Según la bigramática pura

La diferencia entre ambos métodos, en caso de aplicar la fórmula de compresión en caracteres, es que el método bigra-

mático posicional da una mejora del 6,95%, es decir, un 0,95% más que la calculada en la columna 11, mientras que al aplicar la fórmula de compresión en la que operamos a nivel de bits, la mejora es del 8,11%, es decir, un 1,11% mejor que los resultados calculados en la columna 6.

COMPRESION BIGRAMATICA POSICIONAL POR RENDIMIENTO

A la vista de los resultados que se obtenían en el método anterior, se intuía una mejora del algoritmo, si, en vez de seleccionar los BMFs por frecuencia, se seleccionaban por rendimiento.

El rendimiento tiene en cuenta la frecuencia y la longitud, variables ambas que forman parte del cálculo de la compresión.

La base del algoritmo es el método de compresión bigramática, teniendo en cuenta la posicionalidad de los caracteres dentro de las palabras. Si a todo esto, añadimos que la selección de los bloques que se comprimen se hace por máximo rendimiento, se puede conseguir un algoritmo que obtenga unos bloques con mejor rendimiento que el método de compresión bigramática posicional, ya que así la compresión del texto será mayor.

Las palabras, al igual que en el método de compresión bigramática posicional, se podrán componer de tres bloques de símbolos: iniciales, finales e intermedios. Cada uno de estos bloques es el resultado de aplicar el algoritmo de compresión bigramática posicional por rendimiento al texto fuente en tres fases:

- Empezando por la izquierda de la palabra.
- Empezando por la derecha de la palabra.
- Aplicándolo a los restantes caracteres de la palabra.

El algoritmo es el mismo para las tres fases, dado que antes de comenzar la segunda fase, se ponen los caracteres que no han sido comprimidos en la primera fase al revés, es decir, el último carácter de la palabra se pondrá el primero, el penúltimo carácter se pondrá el segundo, y así sucesivamente, hasta terminar con todos los caracteres de la palabra. Para pasar a la tercera fase haremos lo mismo con los caracteres que no han sido comprimidos en la segunda fase.

DESCRIPCION DEL ALGORITMO DE COMPRESION DE BIGRAMATICA POSICIONAL POR RENDIMIENTO.

1.- ELEMENTOS DEL ALGORITMO.

Para realizar el algoritmo vamos a necesitar tres tablas.

- TABLA F: Esta tabla es una matriz de dos dimensiones en la que cada fila esta formada por una palabra del texto fuente, junto con su frecuencia de aparición dentro de ese texto. La matriz tendra tantas filas como palabras

distintas tenga el texto y estará ordenada alfabéticamente.

- TABLA R: Esta tabla es también una matriz de dos dimensiones en la que cada fila esta formada por los distintos rendimientos que se puedan obtener de todos los BMFs posibles de cada palabra. También cada fila contendrá además una columna con el mayor valor de todos ellos y otra columna tendrá su posición dentro de esta fila.

- TABLA S: Esta tabla contendrá por cada BMF que el algoritmo seleccione el símbolo a sustituir y el rendimiento que se obtiene. Los N primeros símbolos serán los correspondientes a los caracteres del alfabeto fuente.

2.- FORMACION DE LA MATRIZ F.

La matriz F va a estar formada por todas las palabras diferentes que componen el texto.

Cada carácter va a ser un elemento $a_{i,j}$, donde i = 1,2,.....,n y j = 1,2,.....,m. El índice i nos indica la posición de la palabra de que se trata y el índice j la posición de cada carácter dentro de la palabra.

La matriz tendrá otras dos columnas:

- Columna de frecuencias: $f_i = a_{i,m+1}$, donde cada f_i será la frecuencia de aparición de la palabra i.
- Columna de máxima longitud: $l_i = a_{i,m+2}$, donde cada l_i indicará la longitud (en número de caracteres) de la palabra i.

Ejemplo Figura II.

3.- FORMACION DE LA MATRIZ R.

Esta matriz va a estar formada por todos los rendimientos $r_{i,j}$ que se obtienen de todos los BMFs posibles de cada palabra.

El cálculo de los $r_{i,j}$ se hará:

$r_{i,j} = P * (j - 1)$ siendo

$P = \sum_{x=s}^{t} f_x$, siempre que se cumpla que:

$a_{s,1}-a_{s,2}-\cdots-a_{s,j}=a_{s+1,1}-a_{s+1,2}-\cdots-a_{s+1,j}=\cdots=$

$= a_{t,1}-a_{t,2}-\cdots-a_{t,j}$

Además, los $r_{i,j}$ que cumplan las siguientes condiciones tendrán el valor cero:

- los $r_{1,j}$
- los $r_{i,j}$ pertenecientes a la misma fila que cumplan que:

  para j = k,k+1,.....,m-1

  donde m = l_i y k sería el menor número que cumpla una de las dos ecuaciones siguientes:

$a_{i-1,1}-a_{i-1,2}-\cdots-a_{i-1,k-1} = a_{i,1}-a_{i,2}-\cdots-a_{i,k-1}$

$a_{i,21}-a_{i,2}-\cdots-a_{i,k-1} = a_{i+1,1}-a_{i+1,2}-\cdots-a_{i+1,k-1}$

Esto es, si dentro de una palabra hay distintos BMFs de igual frecuencia sólo le calcularemos el rendimiento al BMF de más caracteres, el resto tendrá valor cero.

- los $r_{i,j}$, siendo j = k,k+1,.....,l y siendo l = q-1, donde q es el número más bajo tal que $a_{i,q} \neq a_{i+1,q}$, y siendo k el número más pequeño que cumple que $a_{i-1,k} \neq a_{i,k}$, siendo $k \neq q$. Ello es debido a la misma razón que el punto anterior, pero referido a un BMF perteneciente a varias palabras.

Esta matriz tendrá también dos columnas más:

- Columna de máximo rendimiento:

  $rm_i = r_{i,m+1}$, donde cada rm_i será el máximo $r_{i,j}$ de los calculados en la fila i.

- Indice de columna de máximo rendimiento:

  $xm_i = r_{i,m+2}$, donde cada xm_i contendrá la posición j del máximo $r_{i,j}$ de la fila i.

Ejemplo Figura IV

Una vez formadas las tablas F y R, se pasará a desarrollar el algoritmo para calcular los BMFs de mayor rendimiento, para lo cual se irán variando los valores de la tabla R en función de los BMFs que se vayan eligiendo.

4.- DESARROLLO DEL ALGORITMO.

A.- Escoger de entre todos los valores de rm el máximo. A igualdad entre dos filas escoger la fila con l mayor. A igualdad de l, escoger el primero de entre éstos últimos.

B.- Incorporaremos a la tabla S el BMF escogido con su correspondiente rendimiento. Cada BMF de esta tabla se corresponderá con un símbolo del alfabeto codificador. Ejemplo Figura III.

C.- Recalculamos los $r_{i,j}$ en los cuales el BMF escogido esté totalmente embebido.

Lo haremos de la siguiente forma:

1.- Sea $r_{u,v}$ el rendimiento máximo.

Haremos $r_{i,j} = 0$, $\forall$ i que cumpla:

$$a_{i,1}-a_{i,2}-\cdots-,a_{i,v} = a_{u,1}-a_{u,2}-\cdots-a_{u,v}$$

2.- Recalculamos los $r_{i,j}$ que sean:

$\forall$ j > v y $\forall$ i que cumpla:

$$a_{i,1}-a_{i,2}-\cdots-,a_{i,v} = a_{u,1}-a_{u,2}-\cdots-a_{u,v}$$

mediante la fórmula:

$r_{i,j} = P \times (j - v)$ donde

$P = \sum_{x=s}^{t} f_x$, siempre que se cumpla:

$$a_{s,1}-a_{s,2}-\cdots-a_{s,j}=a_{s+1,1}-a_{s+1,2}-\cdots-a_{s+1,j}=$$

$$=\ldots=a_{t,1}-a_{t,2}-\cdots-a_{t,j}$$

En el apartado 1 ponemos a cero todos los $r_{i,j}$ que ya no van a formar parte de la selección, dado que se ha

encontrado un $r_{i,j}$ mejor que ellos y que los aglutina. En el apartado 2 recalculamos los $r_{i,j}$ del resto de la palabra que no quedan cubiertos por el BMF escogido. El rendimiento se obtiene sustituyendo el BMF escogido por un símbolo, con lo cual la longitud varía, lo que hace que haya un cambio de rendimiento que se deberá volver a calcular.

Ejemplo: si FERNAN en la palabra FERNANDA lo sustituímos por "*" nos queda *DA con lo que el $r_{i,j}$ de *DA es:

8 x (3 - 1) = 16 (Frecuencia x (longitud - 1))

Con lo cual la nueva tabla R nos quedaría como en la Figura V.

D.- Recalculamos los $r_{i,j}$ en los cuales el BMF escogido está parcialmente embebido, es decir, los $r_{i,j}$ cuyo BMF asociado sea subconjunto del BMF escogido.

Estos subconjuntos deberán ser de la siguiente forma:

1.- El BMF subconjunto tendrá más de dos caracteres.

2.- Se deberá cumplir que:

$$a_{i,1}-a_{i,2}-\cdots-a_{i,j} = a_{e,1}-a_{e,2}-\cdots-a_{e,j}$$

siendo i el índice del subconjunto y e el índice del BMF escogido.

Ejemplo: al escoger el BMF FERNAN, los BMFs a los que éste puede afectar en sus $r_{i,j}$ serán: FE, FER, FERN, FERNA.

El recálculo de los rendimientos de estos BMFs se hará quitando la parte en que el BMF escogido afectaba a éstos. (Figura VI).

E.- Seguidamente volveremos al punto A de este algoritmo y así estaremos en un ciclo hasta que ya no queramos escoger más BMFs.

Las Figuras IV a XI muestran como se iría desarrollando el algoritmo. En la Figura III, sumando la columna de rendimientos, obtenemos la compresión absoluta del texto (CA) si utilizamos un símbolo para sustituir cada uno de

los BMFs de la tabla.

COMPARACION DE RESULTADOS.

En este apartado vamos a comparar los resultados obtenidos al utilizar el algoritmo de compresión bigramática posicional por rendimiento de dos formas:

- Con los resultados que obtuvimos con el algoritmo de compresión bigramática posicional y

- Con los resultados obtenidos por el método de compresión bigramática tradicional.

En ambos casos los resultados se compararán siempre con el fichero de apellidos que venimos usando habitualmente.

En las pruebas realizadas con el algoritmo de compresión bigramática posicional por rendimiento el texto fuente utilizado es el mismo con 6 caracteres más, que forman una fecha que se le introdujo al texto, por lo que, con los errores de transcripción mas estos 6 caracteres, el texto fuente ha quedado constituido por 14.246 caracteres.

Como alfabeto codificador se tienen tres tablas:

- Una con BMFs iniciales
- Otra con BMFs finales y
- Otra con BMFs intermedios.

Cada una de estas tablas tiene, para poder comparar con los anteriores estudios, 128 símbolos codificados cada uno con 7 bits, de los que los 29 primeros serán los correspondientes a los caracteres del alfabeto fuente.

Los caracteres del alfabeto fuente se codificarán con 6 bits cada uno.

Para comparar los resultados obtenidos nos fijaremos en la Figura XII en la que están expuestos todos ellos.

Si observamos las columnas 6 y 7 de dicha figura, que nos dan la compresión relativa, una calculada en base a bits y la otra en base a caracteres, vemos la mejora que nos da este algoritmo frente a los anteriores.

En la columna 6 se ve una mejora del 7,42% frente al método de compresión bigramática posicional, y del 14,42% frente al método de compresión bigramática pura.

En la columna 11 esta mejora se cifra en un 6,35% frente al método de compresión bigramática posicional, y en un 12,35 frente al método de compresión bigramática pura.

Asimismo el número de bits que harían falta para codificar un carácter del texto fuente baja a 3,23 bits en este método, frente a los 3,68 bits que nos harían falta aplicando el método de compresión bigramática posicional y

frente a los 4,10 bits necesarios en el método de compresión bigramática pura.

Todo ésto se ha hecho teniendo en cuenta que los blancos no son reducibles, ya que si éstos formaran parte de los BMFs y por lo tanto fueran reducibles, entonces el índice de compresión aumentaría de la siguiente forma:

A.- Aplicando la fórmula de compresión relativa en bits:

$$CR = \frac{14.246 \times 6 - (6.584 - 1.921) \times 7}{14.246 \times 6} = 0,6181$$

B.- Aplicando la fórmula de compresión relativa en caracteres:

$$CR = \frac{14.246 - (6.584 - 1.921)}{14.246} = 0,6726$$

En la Figura XIII se han extraido los resultados obtenidos cuando el número de símbolos es igual en las tres tablas. En estos tres casos hubiera sido muy fácil el insertar los blancos dentro de los BMFs tanto para su selección como para su codificación-decodificación. Queremos resaltar que hemos obtenido a pesar de ello, una compresión del 63% y que el número de bits necesarios para codificar un carácter ha sido, como ya hemos dicho anteriormente, de 2,96. Este resultado sería bastante superior si los blancos hubieran entrado a formar parte de los BMFs.

CONCLUSIONES

En este apartado de conclusiones, vamos a tratar de resumir las aportaciones que este estudio puede dar al campo de la compresión de textos o ficheros de apellidos escritos en castellano.

1.- Se ha desarrollado a partir del método de compresión bigramática un nuevo método al que se le ha agregado la idea de seleccionar los bigramas atendiendo a la posición que éstos ocupan dentro de las palabras.

2.- Se ha comprobado que la selección de bloques multiletra nos da un mejor rendimiento si la selección se hace primero escogiendo los BMFs iniciales y luego los finales, que si lo hacemos al revés, es decir, escogiendo primero los BMFs finales y luego los iniciales.

3.- Se ha demostrado que la compresión bigramática posicional da un mayor índice de compresión que la bigramática clásica cuando usamos un número de palabras código aproximadamente igual en los diccionarios de transcodificación.

4.- Se ha comprobado que la compresión bigramática posicional con posterior selección por rendimiento, da en la mayoría de los casos, mejor índice de compre-

sión que si aplicamos tan sólo el método de compresión bigramática posicional sin ningún tipo de selección.

5.- Se ha comprobado que este tipo de compresión no da tan buen rendimiento en ficheros de tipo literario donde los monosílabos constituyen casi un 30% del texto. Este rendimiento, aumenta considerablemente en textos donde este tipo de palabras prácticamente no aparecen como bien pueden ser, los ficheros de apellidos.

6.- Ante la necesidad de introducir una nueva variable - longitud - al método de compresión bigramática posicional, se ha desarrollado un nuevo algoritmo, al que se le ha dado el nombre de algoritmo o método de compresión bigramática posicional por rendimiento.

7.- El algoritmo de compresión bigramática posicional por rendimiento trata de buscar bloques multiletra lo más óptimos posibles, haciendo un análisis parcial de la palabra: es decir, sólo se tiene en cuenta o bien la parte inicial de la palabra para seleccionar bloques iniciales, o bien la parte final de la palabra para seleccionar bloques finales, o la parte intermedia de la palabra para seleccionar bloques intermedios.

8.- Se ha comprobado como este último método supera a los métodos de compresión bigramática tradicional y al de compresión bigramática posicional, en índice de compresión, bien se mida en caracteres o en bits. También los supera en el número de bits que son necesarios para codificar un carácter del texto fuente.

9.- Como conclusión y resumen final, se puede decir que se ha desarrollado un método de compresión, partiendo de la bigramática y de la posición de los caracteres dentro de la palabra, al que se le ha llamado bigramática posicional, obteniendo mejores resultados en ficheros de apellidos castellanos, que la bigramática clásica. Al final, se ha unido a esto la idea original de tener en cuenta el rendimiento a la hora de contruir los diccionarios. Y se le ha llamado bigramática posicional por rendimiento y los resultados son aún mejores que los de los dos métodos anteriores.

BIBLIOGRAFIA

[ARES-83] Ares de Blas, Félix, "Un método para disminuir la redundancia en la transmisión de textos escritos en castellano". Tesis doctoral. F.I.S.S. 1.983.

[GARCIA-80] García Camarero, Ernesto y Bengoechea Martínez, L. "un método para la compresión de textos". Boletín del

Centro de Cálculo de la Universidad Complutense. Número 36. Junio 1.980.

[GURRUCHAGA-85] Gurruchaga, J.,G. de Madinabeitia, J., Gonzalez Abascal, J. y Ares de Blas, F., "Un método de compresión bigramática posicional". Revista de Informática y Automática. Num. 63. PP. 32-35. Enero-Marzo 1.985.

[GURRUCHAGA-86] Gurruchaga, J.,G. de Madinabeitia, J., Gonzalez Abascal, J. y Ares de Blas, F., "Un método de compresión bigramática posicional por rendimiento". Revista de Informática y Automática. Aceptado para su publicación.

[GURRUCHAGA-86] Gurruchaga, J. "Nuevos métodos de compresión bigramática". Tesis de Licenciatura. FISS 1.986.

[GURRUCHAGA-85] Gurruchaga, J.,G. de Madinabeitia, J., Gonzalez Abascal, J. y Ares de Blas, F., "Entropy of Word Position". Revista IEEE ejemplar Mayo-1.985.

[HAMILTON-80] Hamilton, D.A., Herrold P.R., Ossefort M.J., "Digramatic text Compression". IBM Technical Disclosure Bulletin. Vol. 23. N. 2- Julio 1.980.

[HUFFMAN-52] Huffman, D.A., "A method for the construction of Minimun Redundancy Codes". Proc. IRE. 40. PP. 1.098-1.101. 1.952.

[RODRIGUEZ-78] Rodríguez, Prieto, Amador, "Algoritmo de compresión de datos. Una aplicación práctica". Inforprim 78. Madrid, Nov. 1.978.

[WELCH-84] Welch, Terri A., "A technique for High Performance Data Compression". IEEE-Computer. Vol 17, N. 6. June 1.984.

	1	2	3	4	5	6	7	8	9	10	11
	NUM. BMFs INIC.	NUM. BMFs FINAL	NUM. BMFs INTER.	NUM. BITS TEXTO FUENTE	NUM. BITS TEXTO CODIF.	COMPR. RELAT. (en bits)	NUM. BITS/ LETRA SALIDA	COMPR. ABSOL. BMFs INIC.	COMPR. ABSOL. BMFs FINAL.	COMPR. ABSOL. BMFs INTER.	COMPR. RELAT. (en carac)
Bigramatica Posicional	128	128	128	85.440	52.402	0,3866	3,6799	5.063	1.102	589	0,4743
Segun E.G.C y L.B.M	128	---	---	85.596	58.492	0,3166	4,1001	5.910	---	---	0,4143

FIGURA I

i \ j	1	2	3	4	5	6	7	8	9	10	f	l
1	F	E	R	M	I	N					5	6
2	F	E	R	N	A	N					10	6
3	F	E	R	N	A	N	D	A			8	8
4	F	E	R	N	A	N	D	E	Z		8	9
5	F	E	R	N	A	N	D	I	T	O	5	10
6	F	E	R	N	A	N	D	O			6	8
7	G	A	R	C	I	A					5	6
8	G	A	R	M	E	N	D	I	A		10	9
9	G	O	I	C	O	E	C	H	E	A	15	10
10	G	O	N	Z	A	L	E	Z			5	8
11	G	O	N	Z	A	L	O				5	7

FIGURA II

Para i=1,2,...,11 y

j=1,2,...,10

Simbolo	BMF	Rendim.
α	FERNAN	185
β	GOICOECHEA	135
γ	GARMENDIA	80
δ	GONZAL	50

FIGURA III

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		5	10			25					25	6
2											0	0
3							27	16			27	7
4							27		24		27	7
5							27			20	27	7
6							27	12			27	7
7		15	30			25					30	3
8		15	30						80		80	9
9											0	0
10		10				50		35			50	6
11		10				50	30				50	6

FIGURA VIII

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		5	10			25					25	6
2											0	0
3							27	16			27	7
4							27		24		27	7
5							27			20	27	7
6							27	12			27	7
7		15	30			25					30	3
8											80	9
9											0	0
10		10				50		35			50	6
11		10				50	30				50	6

FIGURA IX

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		5	10			25					25	6
2											0	0
3							27	16			27	7
4							27		24		27	7
5							27			20	27	7
6							27	12			27	7
7		5	10			25					25	6
8											0	0
9											0	0
10		10				50		35			50	6
11		10				50	30				50	6

FIGURA X

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		5	10			25					25	6
2											0	0
3							27	16			27	7
4							27		24		27	7
5							27			20	27	7
6							27	12			27	7
7		5	10			25					25	6
8											0	0
9											0	0
10								10			50	6
11							5				50	6

FIGURA XI

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		42	84			25					84	3
2		42	84			185					185	6
3		42	84			185	162	56			185	6
4		42	84			185	162		64		185	6
5		42	84			185	162			45	185	6
6		42	84			185	162	42			185	6
7		15	30			25					30	3
8		15	30						80		80	9
9		25								135	135	10
10		25				50		35			50	6
11		25				50	30				50	6

FIGURA IV

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		42	84			25					84	3
2											185	6
3							27	16			185	6
4							27		24		185	6
5							27			20	185	6
6							27	12			185	6
7		15	30			25					30	3
8		15	30						80		80	9
9		25								135	135	10
10		25				50		35			50	6
11		25				50	30				50	6

FIGURA V

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		5	10			25					25	6
2											0	0
3							27	16			27	7
4							27		24		27	7
5							27			20	27	7
6							27	12			27	7
7		15	30			25					30	3
8		15	30						80		80	9
9		25								135	135	10
10		25				50		35			50	6
11		25				50	30				50	6

FIGURA VI

i \ j	1	2	3	4	5	6	7	8	9	10	rm	xm
1		5	10			25					25	6
2											0	0
3							27	16			27	7
4							27		24		27	7
5							27			20	27	7
6							27	12			27	7
7		15	30			25					30	3
8		15	30						80		80	9
9										135	135	10
10		25				50		35			50	6
11		25				50	30				50	6

FIGURA VII

	1	2	3	4	5	6	7	8	9	10	11
	NUM. BMFs INIC.	NUM. BMFs FINAL	NUM. BMFs INTER.	NUM. BITS TEXTO FUENTE	NUM. BITS TEXTO CODIF.	COMPR. RELAT. (en bits)	NUM. BITS/ LETRA SALIDA	COMPR. ABSOL. BMFs INIC.	COMPR. ABSOL. BMFs FINAL.	COMPR. ABSOL. BMFs INTER.	COMPR. RELAT. (en carac)
Bigramatica Posicional Rendimiento	128	128	128	85.476	46.087	0,4608	3,2350	5.448	1.558	656	0,5378
Bigramatica Posicional	128	128	128	85.440	52.402	0,3866	3,6799	5.063	1.102	589	0,4743
Segun E.G.C y L.B.M	128	---	---	85.596	58.492	0,3166	4,1001	5.910	---	---	0,4143

FIGURA XII

1	2	3	4	5	6	7	8	9	10	11
NUM. BMFs INIC.	NUM. BMFs FINAL	NUM. BMFs INTER.	NUM. BITS TEXTO FUENTE	NUM. BITS TEXTO CODIF.	COMPR. RELAT. (en bits)	NUM. BITS/ LETRA SALIDA	COMPR. ABSOL. BMFs INIC.	COMPR. ABSOL. BMFs FINAL.	COMPR. ABSOL. BMFs INTER.	COMPR. RELAT. (en carac)
256	256	256	85.476	42.264	0,5055	2,9667	6.939	1.627	397	0,6291
128	128	128	85.476	46.087	0,4608	3,2350	5.448	1.558	656	0,5378
64	64	64	85.476	50.142	0,4133	3,5197	3.957	1.320	612	0,4133

FIGURA XIII