ضغط ملفات النص بدمج طريقتين من طرق القاموس (قاموس خاص للغة محددة مع LZ77)

احمد حسين عليوي*

تاريخ التسلم: 2007/7/4
تاريخ القبول:1/31 /2008

الخلاصة

في هذا البحث اقتراح لدمج طريقتين من طرق القاموس وهما طريقة القاموس المتكامل للغة محددة و LZ77 . يستخدم القاموس للتعويض عن كل كلمة موجودة (معرفة) في القاموس العام للغة معينة ببايتين (Two Bytes) والكلمات الغير موجودة في القاموس تبقى على حالها مع كتابة عدد من البتات قبلها للاشارة إلى طولها. قمنا بعدة تعديلات على هذه الطريقة للحصول على اقل حجم ممكن للملف. كل كلمة يعوض عنها ببايتين وسوف يعوضان عن تلك الكلمة في أي مكان ترد فيه هذه الكلمة ، فلذلك تم استخدام الطريقة الثانية LZ77 (الكلمات عن الناحية العملية بقت على ترتيبها). قبل ذلك قمنا بترتيب الملف بطريقة خاصة استخدمناها بحيث نستطيع الاستفادة من طريقة (LZ77)لتقليل البيانات الى آفل حد ممكن. هذه الطريقة جريت على ملفات نصوص حقيقية وأثبتت نجاحها.

Text Files Compression using Combination of two Dictionary methods (Specific dictionary for specific language and LZ77 Approach)

Abstract

In this paper we suggest combination between two dictionary methods. Specific dictionary (for specific language) and LZ77. The dictionary is used for replacing any word in it by its two Bytes index. A words not exists in the dictionary is written without changing preceded by four bits for length of this word. A modification was made on this approach for reducing the file to minimum size. Because of each word was replaced by two bytes (which are substituted in any appearance for this word in text), LZ77 can be used efficiently. Before this, the file is arranged specially in order to use LZ77 optimally for minimizing the data. This approach is tested on real text files and verifies it's successful.

1- المقدمة

عتبر ضغط البيانات من الأمور المهمة في مجال نقل البيانات وفي الأوساط الخزنية لخزن البيانات. هنالك تطبيقات ضخمة لمعالجة البيانات تحتاج لأوساط خزنية كبيرة لخزن البيانات. وهذه التطبيقات في تزايد مع التزايد المستمر لاستخدامات الحاسوب. وفي نفس الوقت هناك انتقال لكميات كبيرة من البيانات عبر قنوات الاتصال مع تطور شبكات الحاسبات. فضغط البيانات المخزونة او المنقولة يؤدي إلى تقليل حجم الوسط ألخزني المطلوب و / أو كلفة الاتصال. فعندما نقلل كمية البيانات المرسلة فسوف يؤدي إلى زيادة سعة قناة

* قسم علوم الحاسوب ، الجامعة التكنولوجية / بغداد

https://doi.org/10.30684/etj.29.16.18
2412-0758/University of Technology-Iraq, Baghdad, Iraq

الاتصال. بشكل مشابه للوسط الخزني فانه يزداد إلى الضعف [1][3].

التحدي الأكبر في عملية الضغط هو تقليل حجم البيانات إلى اقل حد ممكن , ولأننا نتعامل في هذا البحث مع ملفات النص (Text Files) ينبغي أن نتعامل مع طرق ضغط البيانات التي لا تسمح بفقدان البيانات عند استرجاع النص الأصلي (Lossless).

2- استخدام القاموس المتكامل للغة محددة لضغط النص

كما هو معلوم فان نصوص اللغة الطبيعية تتكون من كلمات ورموز تكون الجمل وأغلب هذه الكلمات هي مأخوذة من قائمة كلمات محددة تدعى القاموس لتلك اللغة , ويدعى هذا القاموس بالقائمة المرجع. ولأنه اغلب النصوص هي موجودة ضمن هذا القاموس , فلذلك يمكن استخدام إشارة مرجعية كرقم مثلا للإشارة إلى تلك الكلمة ضمن القاموس. وقد وجد عمليا يمكن استخدام بايتين كإشارة مرجعية لأي كلمة في اللغة الإنكليزية (هناك ما يقارب 40000 كلمة في اللغة الإنكليزية).[4]

بالاعتماد على القيم الإحصائية , معدل طول الكلمة في اللغة الإنكليزية هي (4.3) حرف بالطول هذا في حالة حذف الفراغ الموجود بين كلمتين . فاذا اخذ الفراغ بنظر الاعتبار فيكون طول الكلمة هو (5.3) أي نحتاج إلى (5.3) بايت كمعدل فإذا استخدمنا بايتين كفهرسة (إشارة مرجعية) وطبعا سوف تغطي كافة الكلمات الموجودة في القاموس كما ذكر سابقا فهذا سوف يجعل لكل كلمة بايتين وبالتالي فان معدل الضغط سوف يكون: [4]

$(5.3-2)/5.3 = 62\ \%$

تكمن الفائدة من حقيقة كون النصوص في اللغة تكون (الجزء الأكبر) مجموعة معينة من الكلمات (وهنا لا نقصد مجموعة قليلة).[4]

النص ربما يحتوي على كلمات غير موجودة في القاموس كان نكون أسماء أشخاص او اماكن غير معروفة... الخ , في هذه الحالة يجب ان نتخذ بعض العمليات لحل مثل هذه الحالة. في بحثنا هذا افترضنا صيغة جديدة لحل هذه الحالة والتي تلائم متطلبات البحث.

3- الضغط باستخدام LZ77 وطرق ZIP

تعتبر تقنية ZIP من التقنيات الأكثر شيوعا في الاستخدام لخوارزميات ضغط البيانات. وظيفا هي تكافئ PKZIP المتوفرة بشكل واسع لأنظمة الوندوز(Windows) والتي طورت من قبل PKWARE. [6,5] في سنة 1977 قام كل من Ziv Jacob و Abraham Lemple بوصف تقنية معتمدة بالأساس على خزان نافذة تزحيف (Sliding Window Buffer) والذي يحتوي على النصوص الأقرب معالجة للوقت الحالي. وأطلق على هذه الخوارزمية باسم LZ77 وبعدها LZ78 عام 1978. نسخة من هذه الخوارزمية تستخدم في ZIP. [6,5]

ان خوارزمية LZ77 تعتمد بالأساس على حقيقة كون الكلمات والعبارات ضمن النص تتكرر باستمرار. فعندما يظهر هذا التكرار فيمكن استبداله برمز اقصر وبالتالي تقليل حجم الملف. وبالتأكيد هنالك عملية بحث عن العبارات والكلمات المتكررة. وهذه الخوارزمية تعتبر من طرق القاموس [6]. يعوض عن التكرار برمز يتكون من:

- Pointer: مؤشر يشير إلى بعد بداية نفس العبارة عن العبارة المكررة الحالية.
- Length: (الطول) ويمثل طول العبارة المكررة (عدد الأحرف).

ممكن أن يكون عدد بتات المؤشر والطول ثابتة او متغيرة وغالبا ما يستخدم الخيارين التاليين؛

- 8 بت للمؤشر و 4 بت للطول.
- 12 بت للمؤشر و 6 بت للطول.

وللتمييز بين هاتين الحالين يضاف حقل إلى الحقلين السابقين يدعى (Header) طوله بت واحد يحدد استخدام الحالة الأولى أو الثانية. وبذلك سيكون هيئة الرمز الذي يعوض عن العبارة المكررة كالآتي؛

< Header> < pointer > < Length>

ولغرض معرفة هل هناك ضغط (إبدال للتكرار) ام لا يضاف بت إضافي لكل حرف أي يصبح طول كل حرف هو 9 بت بدلا من 8 بت. فإذا كان هذا البت 1 فان البتات الثمانية التي بعده هي حرف أما إذا كان 0 فالبتات التي بعده تمثل الصيغة:

< Header> < Pointer > < Length>

4- تطبيق الضغط باستخدام القاموس للغة محددة

كما مر ذكره بالفقرة (3) السابقة فان لكل كلمة موجودة في القاموس هناك فهرس(index) ممثل ببايتين يعوض عن تلك الكلمة في اي موقع تظهر فيه تلك الكلمة هناك بعض الفهرسة غير مستخدمة لأنه لدينا 2^{16} اي ما يعادل 65000 كلمة أما الكلمات الكلية فهي 40000 كلمة للغة الإنكليزية [4] يستخدم ما تبقى للإشارة إلى رموز وكلمات أخرى وهنا سوف نسرد الخوارزمية المقترحة لتطبيق طريقة الضغط باستخدام القاموس.

- **خوارزمية الضغط باستخدام القاموس مع التعديل**

أولا:- يعتبر الفاصل بين جميع الكلمات هو الفراغ (Space) أو رمز الإدخال (Return/ Linefeed).

ثانيا:- إذا كانت الكلمة موجودة في القاموس فيعوض عنها ببايتين يمثلان الفهرسة (Index) المقابلة لهذه الكلمة في القاموس.

ثالثا:- إذا كانت الكلمة غير موجودة في القاموس فتترك على حالها مسبوقة بأربعة بتات تمثل طول هذه الكلمة.

رابعا:- كل فراغ متبوع بكلمة موجودة في القاموس يعوض عنه ببت واحد قيمته 1.

خامسا:- يعوض ب 0 متبوع بأربعة بتات لتحديد ما يلي:

أ. الرمز 01111XXX للدلالة على عدد من الفراغات المتتالية حيث XXX تدل على عدد الفراغات الكلية مطروح منها واحد يعالج مع الكلمة اللاحقة. أما إذا كان عدد الفراغات أكثر من 7 فنستخدم الرمز 01111000XXX (يضاف قيمة XXXالى 7) اما إذا كانت أطول من 4 فنستخدم الرمز (01111000000XXX) وهكذا. كمثال الرمز 01111000000101 يدل على 19 فراغ. الرمز01111111 يدل على 7 فراغات اما الرمز 01111000001 فيدل على 8 فراغات.

ب. الرمز([XXX]00000XXX) للدلالة على رمز الإرجاع (Return / Linefeed) أما البتات الثلاثة XXX الأولى فهي:

- أول بت إذا كان 1 فيسبق رمز الإدخال الفراغ أما اذا كان 0 فلا يسبقه فراغ.
- البت التائي والثالث فهي:

1. 10 بعده كلمة موجودة في القاموس(بايتين) •
2. 11 بعده كلمة غير موجودة في القاموس والبتات الثلاثة التالية لها [XXX] تحدد طول تلك الكلمة. وهذه البتات الثلاثة تستخدم كما في الفقرة (أ)و(ز)•
3. 01 بعده حرف ويتبعه ذلك الحرف.
4. 00 بعده لاشيء أي يقرء ما بعده بداية رمز جديد.

جـ - الرمز 01101 يستخدم للدلالة على ان الكلمة التالية تتكون من حرف واحد وهذا مفيد في الرموز الخاصة عدا الفراغ او رمز الإرجاع.

د- الرمز 01100XXX يستخدم للدلالة على إن الكلمة التالية تتكون من حرف

واحد مكرر وان XXX تمثل العدد (عدد التكرار). يكون استخدام هذه البتات كما في الفقرة (أ) والفقرة(ز).

هـ - الرمز 01011 يستخدم للإشارة إلى إن هنالك حرف متصل بنهاية الكلمة السابقة (ينبغي أن تكون الكلمة موجودة بالقاموس) ويذكر بعده الحرف وهذه تستخدم مع الحروف (. , : ... الخ).

و - الرمز 0XXXX تدل على فراغ متبوع بكلمة غير موجودة في القاموس مع ملاحظة أن XXXX تدل على طول الكلمة ويجب عدم استخدام الرموز 0000, 1111 , 1101 , 1100
1011, و 1110 فهي تستخدم لأغراض أخرى كما مر بالخطوات السابقة والفقرة التالية.

ز - الرمز 01110XXX يستخدم في حالة كون الكلمة أطول من(10) عشرة حروف وان طول تكملة الكلمة هو XXX مع الاحتفاظ بالرمز 000 للدلالة طى تكملة أخرى فيما إذا كانت الكلمة أطول من 17 حرفا وهكذا كما في الفقرة(أ).

سادسا: - في بداية الملف فقط نستخدم عدة بتات [XXX]XX للإشارة إلى:

- اول بت هل يوجد فراغ (1) ام لا (0) في البداية.
- ثاني بت هل هنالك أكثر من فراغ (1) أم لا(0) وهو يحدد كيفية قراءة البتات التي بعده فإذا كان (1) فعدد البتات التي بعده هي ثلاثة[XXX] تمثل عدد الفراغات مطروح منها واحد يعالج مع الكلمة اللاحقة (اذا كان هنالك فراغ واحد فقط فانه يترك للمعالجة مع الكلمة التاليه) وهذه تعامل كما في الفقرة (أ) و (ز). أما إذا كان (0) فالذي بعده هو بداية رمز جديد تتم قراءته حسب الفقرات السابقة.

ولتوضيح بعض هذه النقاط ناخذ المثال التالي (العبارات في هذا المثال عشوائية) :

Parallel processing is established by distributing the data among the multiple functional units.

The operands in the registers are applied to one of the units

عند معالجة هذا النص حسب الخوارزمية السابقة سيكون الناتج حسب ما موضح بالشكل(1). حيث إن:

- 2B تعني بايتين يعوضان عن كلمة موجودة بالقاموس.
- 1B تعني بايت عوض عن حرف.

ففي البداية عوضنا بالرمز11100 حيث إن 11 تدل على وجود اكثر من فراغ و 100 تدل على أربعة فراغات. يتبع هذه الفراغات الأربعة فراغ ترك لكي يعالج مع الكلمة اللاحقة. ولان هذا الفراغ متبوع بكلمة موجودة بالقاموس, عوض عنه ببت واحد قيمته 1 متبو ع بالفهرسة المقابلة لتلك الكلمة في القاموس (بايتان). ثم يتبع هذان البايتان بت واحد قيمته 1 للدلالة على فراغ متبوع بكلمة موجودة بالقاموس وهكذا. بقية الرموز موضحة في الشكل رقم (1).

من الشكل (1) فان عدد البايتات الكلي بعد الضغط هو 57 بايت أما عدد البايتات الكلي قبل الضغط فهو164 بايت أي بنسبة ضغط 65.244 % (أخذنا (Return/ Linefeed كبايتين وهذا ما يحصل عند التعامل مع الملفات عمليا). وهذه النسبة ليست ثابتة بل تتغير اعتمادا على الكلمات هل موجودة في القاموس أم لا (أسماء أشخاص مثلا).

5- هيئة الملف بعد تطبيق طريقة القاموس

عند تطبيق خوارزمية القاموس فسيصبح لدينا ملف عشوائي التركيب أي عند البحث عن تطابق لكلمات او عبارات مكررة أو حساب تكرار الأحرف فلا يمكننا الحصول على هذا التطابق نتيجة للتزحيف الحاصل في الأحرف بسبب الرموز المتغيرة في الطول (تعويض 1 عن فراغ وتعويض 0000XXX

عن رمز الإرجاع ...الخ). هذه الحالة تعقد استخدام أي طريقة أخرى بعد هذه الطريقة.
قمنا هنا بحل هذه المشكلة عن طريق تقسيم الملف إلى قسمين. الجزء الأول يحتوي على الكلمات (البايتات المعوضة عن الكلمات والاحرف فقط) , أما الجزء الأخر فيحتوي على البتات الدالة على الفراغ أو طول الكلمة ... الخ. إن الجزء الثاني يكون بسيط وصغير مقارنة بحجم الملف الأصلي أو الجزء الأول. فما يهمنا هو الجزء الأول من الملف الذي يحتوي على الكلمات فقط بدون فراغات أو علامات الإرجاع (أحرف فقط). وكما يلاحظ إن تسلسل الكلمات بقي على نفس الترتيب, مما يتيح لنا استخدام طريقة أخرى للضغط مثل LZ77. سوف تكون هيئة الملف (File Format) كما موضحة بالشكل (2). حجم الجزء الأول يسجل في بداية الملف حيث ترك له ثلاث بايتات.
عملية فك الضغط من هذه النقطة تتم بقراءة بايت من الجزء الثاني ومنه نحدد كيفية التعامل مع البايتات الموجودة في الجزء الأول. عملية الوصول إلى الجزء الثاني تتم بتحديد حجم الجزء الأول في بداية الملف بثلاث بايتات. ولتسريع عملية القراءة ففي بداية عملية فتح الضغط نستطيع قراءة الجزء الثاني بالكامل و نضعه في مصعفوفة حروف. هذه العملية تخلصنا من القفز إلى نهاية الملف كلما نحتاج إلى قراءة بايت من الجزء الثاني لتحديد الأجراء على بايتات الجزء الأول وهذا ما قمنا به عمليا.
الجزء الثاني يمكن كبسه اذا كان الملف كبير باستخدام (Run length) وذلك للتسلسل المتكون من رموز الفراغات (الاكثر ظهورا) مع الرموز الأخرى. لم نستخدم أى عملية ضغط على محتويات الجزء الثائي للملف عمليا.

6- تطبيق الضغط باستخدام LZ77

كما مر ذكره فان LZ77 تعتمد بالأساس على التكرار في العبارات. ولغرض زيادة الكفاءة تستخدم LZ77 نافذة تزحيف (Sliding Window).
تم تنفيذ الخوارزمية كما يلي:

أولا:- حجم المؤشر (Pointer) هو 12بت (4k) ويمكن أن تأخذ 13 بت (8k) مع ملاحظة أننا أخذنا عمليا قيمة واحدة على امتداد الملف وهذا يعتمد على حجم الملف.

ثانيا:- الطول (Length) وضعنا له 6 بت او 5 بت واستخدامه مع المؤشر يحدد في بداية الملف.

ثالثا:- استخدمنا بت واحد للدلالة على الذي بعد هذا البت هل هو حرف (قيمة البت هي 1) أم الصيغة التالية (قيمة البت هي 0):

<Pointer><Length>

رابعا:- نافذتان (شكل 3) يستخدمان للمقارنة هما:

- Sliding Window (History) Buffer
نافذة تحتوي على أخر حروف من المصدر تم معالجتها. حجم هذه النافذة مساوي لما يستطيع المؤشر عنونته.
- Look Ahead Buffer:
نافذة تحتوي عى الحروف التالية التي يجب ان تعالج.

خامسا:- نطابق حرفين او أكثر من بداية Look Ahead Buffer مع السلاسل الرمزية الموجودة في Sliding Window Buffer وسوف يحدث احد الأمرين:

- اذا لم يحصل تطابق فان اول حرف في Look Ahead Buffer يكون هو الخرج ويضاف له بت تاسع. كذلك يزحف هذا الحرف إلى نافذة النزحيف Sliding Window (بدون إضافة بت تاسع إلى الحرف).
- اذا حصل تطابق فان الخوارزمية تستمر بالمسح لاستخراج اكبر تطابق (اكبر سلسلة رمزية متطابقة). سوف يكون الخرج هو

التعويض عن تلك السلسلة الرمزية بثلاثة حقول هي:

< Header> < Pointer > < Length >

ثم يزحف أحرف بعدد مساوي للطول (Length) إلى نافذة التزحيف.

7- النتائج والاستنتاجات:

طبقت الخوارزمية المقترحة في البحث على مجموعة ملفات متوفرة باللغة الانكليزية عامة الاستخدام في الحاسبة وتكون هذه الملفات مخزونة في المسار الذي تم تنصيب برنامج WinRAR فيه: وتم المقارنة مع نتائج الضغط للبرنامج المعروف (WinRAR) الإصدار 2005 وكانت النتائج كما موضحة بالجدول (1).

بعد ترتيب الملف بعد الضغط باستخدام القاموس فيمكن استخدام أي طريقه من طرق LZ لغرض الضغط بدلا من LZ77.

References

[1]Hoffman R. **"Data Compression in Digital Systems"**. Chapman & Hall.. 1997.

[2] Ian H., Radford M. Neal, John G. Cleary, **"Arithmetic coding for data compression"**, Communication of the ACM, New York, June 1987, Volume 30, Issue 6, pp: 520-540.

[3] Debra A. Lelewer and Daniel S. Hirschberg "**Data Compression** " Chapman & Hall, 2003.

[4] Udayan Khurana and Anirudh Koul **"Text Compression And Superfast Searching"** Thapar Institute Of Engineering and Technology, Patiala, Punjab, India-147004, 2002.

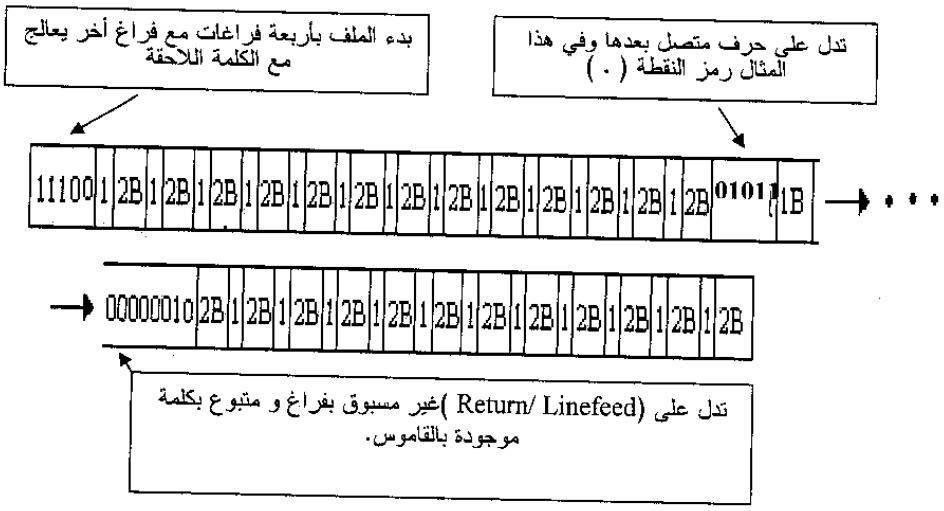
[5] William Stallings " Cryptography and Network Security : Principles and Practice " 1999 Prentice -Hall. Inc

[6]David Salomon "Data Compression the Complete Reference" Third Edition 2004.

جدول رقم (1)

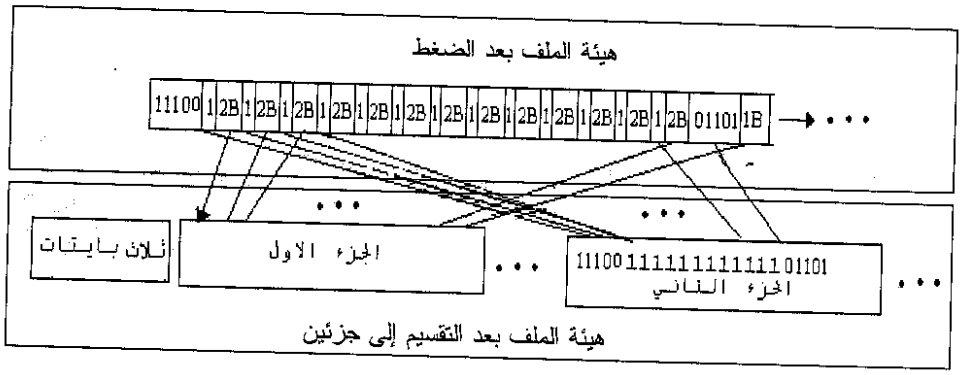
مقارنة النتائج للطريقة المقترحة مع نتائج البرنامج (WinRAR)

اسم الملف	الحجم الأصلي للملف (Bytes)	حجم الملف بعد الضغط (Bytes)			نسبة الضغط %	
		باستخدام WinRAR	باستخدام القاموس فقط	باستخدام القاموس مع LZ77	باستخدام WinRAR	باستخدام القاموس مع LZ77
Readme	1687	911	739	733	45.999	56.550
License	4482	1927	1890	1811	57.005	59.593
Whatnew	11569	4364	5196	4697	62.278	59.440

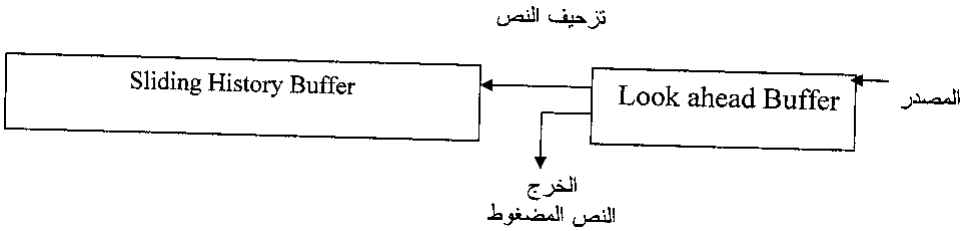


شكل رقم (1)
شكل النص بعد تطبيق طريقة القاموس عليه

ضغط ملفات النص بدمج طريقتين من طرق
القاموس(قاموس خاص للغة محددة مع LZ77)



شكل رقم (2)
تقسيم الملف إلى قسمين احدهما يحتوي على الكلمات والآخر على البتات الدالة على الفراغ... الخ



شكل رقم (3)
خوارزمية LZ77 مع استخدام نافذة التزحيف.