J.K.A.U.: Sci., vol. 5, pp. 109-119 (1413 A.H./1993 A.D.

Segmentation of Handwritten Arabic Words

T.S. EL-SHEIKH and S.G. EL-TAWEEL
Department of Computer Science, Faculty of Science, King Abdulaziz University, Jeddah, Saudi Arabia, and Inst. of Comm. Research, Cairo, Egypt

ABSTRACT. In this paper, we develop an efficient technique for the segmentation of on-line handwritten Arabic words. The segmentation process is followed by the recognition of the segmented characters and consequently recognizing the word. Some words contain one subword while others may contain two or more subwords. Thus each word, which may contain one character, is independently processed. The technique detects whether the subword contains one character or more. For the first case, no segmentation is required, while segmentation is necessary in the other case. Generally, a subword contains one main stroke and one or more secondary strokes. For the segmentation of the main stroke of a subword, special cases for the first character is considered first. Consequently, the last character of the subword is checked for special cases also. Having considered all special cases, other general cases of characters that are within the subword or at its beginning or end are considered in a different manner. The segmentation technique has been applied to a large set containing about one thousand subwords with a probability of success of about 0.99. This high probability is mainly due to the fact that the tested words have been entered by a single writer and were not completely free.

Introduction

Considerable attention has been paid to the cursive script recognition (CSR) problem, starting around 1960 with the work of Frishkopf and Harmon[1] at Bell Telephone Laboratories. There are two main approaches for the recognition of cursive words.

In the first approach, termed "whole-word identification", the features of the entire word are used to identify the word without segmenting it. i.e., without identifying its characters. In the system developed by Frishkopf[1], whole words are compiled according to certain features in a dictionary with which input words are compared. Recognition results were very poor with a limited vocabulary. Earnest[2] reported work on a word recognition system capable of recognizing about 10,000 words with only six features. More recent interest was shown by Farag[3] who dealt with recognition of cursive script in a limited vocabulary environment. This environment can be found in the recognition of handwritten programs and in shipping or manufacturing where key words direct further processing. Also, Berthod and Ahyan[4] proposed a system based on two complementary representations of words; the structure and the aspect.

The second approach, which is often termed "letter-by-letter identification", treats each word as a concatenation of a number of characters. Thus, this method starts with the segmentation of each word into its characters using generally complicated techniques. The second step involves recognizing each character and by using contextual information the word is identified. In the individual-letter method of analysis, proposed by Harmon[1], letter identification was based on the extraction of certain local features of individual letters incorporated in whole words. Using segmentation, and setting up a decision tree based on a number of classes of features, fair recognition results were achieved. The method of analysis-by-synthesis of handwriting has been used by Eden[5]. Eden and Mermelstein[6] have sequently carried out experiments on computer recognition of connected handwritten words by treating the handwritten pattern as a two-dimensional vector displacement function of time. A contextual cursive script recognition system has been proposed by Ehrich and Koehler[7]. This system makes use of letter context to determine word length, letter segmentation and character identity to achieve contextual recognition at the word level. A hybrid algorithm has been developed by Shinghal[8] by combining a Markov method with a dictionary method.

Very little work has been undertaken in the area of Arabic character recognition. The IRAC II system developed by Amin[9] recognizes Arabic words entered via a graphic tablet based on segmentation. Other authors developed different methods for the recognition of cursive Arabic writing using segmentation[10,11]. These methods deal with the recognition of off-line words.

In this paper, we develop an efficient technique for the segmentation of on-line handwritten Arabic words. The segmentation process is followed by the recognition of the segmented characters[12] and consequently recognizing the word.

Preliminary Consideration

The main objective of this paper is to develop an efficient technique for the segmentation of handwritten Arabic words. These words are entered via a graphic tablet. The samples of the real-time handwritten input are transformed into samples representing points equally spaced along the handwriting curve. The x and y coordi-

nates of each point are thus stored while the stylus is moving. Few steps are made to transform the original file into a file containing the x and y coordinates of the two points connnecting each stroke. If two strokes are directly connected, the first point of the second stroke is obviously identical to the second point of the first stroke. This would continue while writing a word or a subword until the writer lifts the stylus (i.e. pushes its button) indicating the end of the word or subword. Thus, these separate words or subwords are readily isolated by checking each two consecutive points that belong to consecutive strokes. If the two points are identical, then the two strokes belong to the same word or subword. On the other hand, if the two points are different, then the two strokes belong to two different words or subwords.

Some words contain one subword while others may contain two or more subwords. A subword contains generally, one main stroke and one or more secondary strokes. A secondary stroke may be a group of dots (one, two, or three), a zigzag, a vertical line or a diagonal line. A secondary stroke can be easily detected since it usually contains very few points, it is always not connected to its main stroke and it is usually within it. The segmentation technique, thus, starts by processing the main stroke of a word or subword with no regard to secondary strokes. After segmenting the word, each secondary stroke is added to the proper character by checking the x coordinates of both.

The segmentation technique detects whether the subword contains one character or more. This is achieved by checking the number of maxima in the y direction in the left half of the subword. Thus if no ymax exists, it means that the subword contains only one character, and no segmentation is required. On the other hand if one or more ymax exist in this region, the subword would contain more than one character and segmentation is necessary in this case.

In order for the segmentation technique to be independent of the size of the words, all distances are calculated relative to the dimensions of the words. Thus, all horizontal distances are calculated relative to the width which is defined as the horizontal distance between the two points, in the subword, having the largest and smallest x values. On the other hand, all vertical distances are calculated with respect to the height which is defined as the vertical distance between the two points, in the subword, having the largest and smallest y values. In the segmentation process, all thresholds are adaptive, i.e., they are independent of the size of different words.

For the segmentation of the main stroke of a subword, special cases for the first character is considered first. This is explained in detail in the next section. Consequently, the last character of the subword is checked for special cases also. The steps are given in detail in the fourth section. Having considered special cases, other general cases of characters that are within the subword or at its beginning or end are considered in a different manner. This step and the method of correcting segmentation points are considered in more detail in the fifth section. The segmentation technique has been applied to a set of about one thousand subwords with a probability of success of 0.99.

Segmentation of the First Character

For the segmentation of the main stroke of a subword, there are two cases regarding the first character. The first case, where the first character is included in the set S_1 containing the characters shown in Fig. 1a, is considered in the first step. Other characters, which are considered in a later step, are elements of the set S_2 and shown in Fig. 1b.

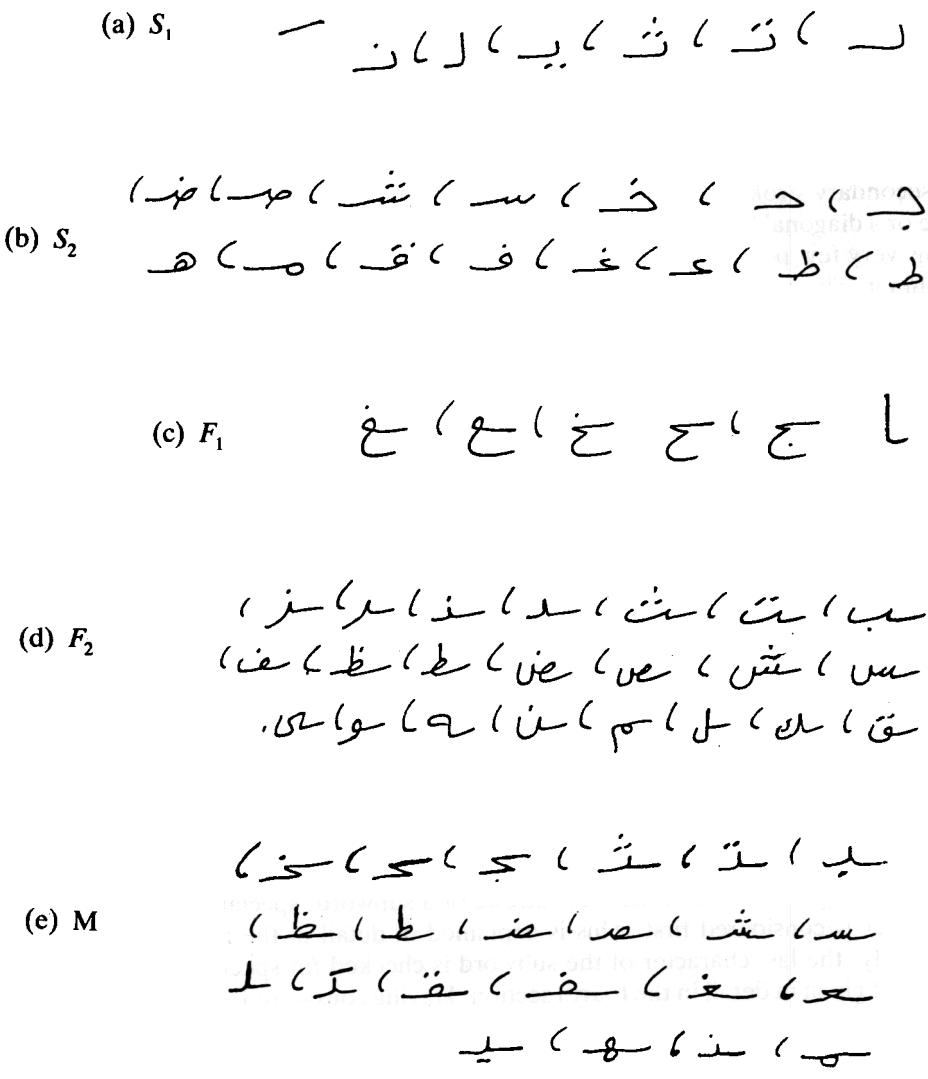

FIG. Different character sets.

Two parameters and two thresholds are defined first and consequently the segmentation step is explained. Thus, R_1, R_2, THR_1 and THR_2 are defined as follows :

$$R_1 = (x_L - x_F) / W \tag{1}$$

$$R_2 = (x_L - x_E) / W \tag{2}$$

$$\begin{array}{ll} 0.05 & W < 1 \\ 0.03 & 1 < W < 3 \end{array} \tag{3}$$

$$\begin{array}{ll} 0.02 & 3 < W < 5 \\ 0.01 & W > 5 \end{array}$$

$$THR_2 = \quad 0.12 \tag{4}$$

where x_L, x_F and x_E are respectively the x coordinates of the point having the largest x coordinate, the first point, and the first extreme point in the x or y direction other than x_L and W is the width of the subword in inches as defined before. Thus, for a character to be included in S_1, the following two conditions should be verified :

$$R_1 < THR_1 \tag{5a}$$

$$R_2 > THR_2 \tag{5b}$$

In this case we segment, before the extreme point having an x coordinate x_E, at a point having a horizontal distance of 0.1 W from the point having the largest value of x (x_L). Figure 2 demonstrates this special case where the segmentation process is shown in detail for the first character only. Having segmented the first character for this special case, we will later proceed from this segmentation point. In other words, the subword, now, starts at the next character and extends up to the last character. Other characters at the beginning of subwords, which do not satisfy the above conditions, are processed in a different manner in a later stage.

FIG. 2. Segmentation of first character for a special case.

Segmentation of the Last Character

The next step of the segmentation technique is to check the last character of the subword, starting from the last point of the main stroke up to the segmentation point, if any, of the first character. It is important to note that if the first character of the considered subword has been already segmented, then W is the width of the new subword after isolating the first character. Similar to the first character, two special cases are considered differently. The characters satisfying these two cases are shown in

Fig. 1c and are referred to as the set F_1. Other end characters, referred to as F_2, are shown in Fig. 1d.

For the first case, the last character is the character "L" where two subcases exist. In the first subcase, the subword does not contain a maximum in the y direction (ymax). Thus, starting from the last point we segment at a point having a horizontal distance of 0.1 W. The last character, thus, contains all points between the segmentation point and the last point.

For the other subcase, one or more maxima in the y direction would exist in the subword. The one maximum that is nearest, horizontally, to the last point is of special interest. We refer to these two points respectively as (x_M, y_M) and (x_T, y_T). Thus, the subcase is evident if the following conditions are satisfied :

$$y_T - y_M > 0.5\ H \tag{6a}$$

$$x_M - x_T > 0.2\ W \tag{6b}$$

In both subcases, we segment at a point having a horizontal distance of 0.1 W from the last point where, clearly, none of the points having maximum y is included. These two subcases are demonstrated in Fig. 3.

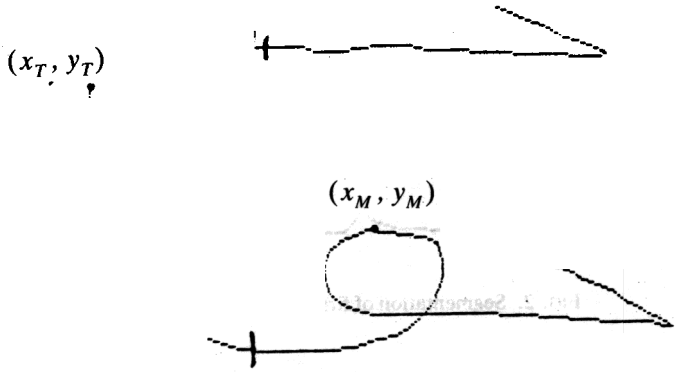

FIG. 3. Segmentation of last character for first special case.

The other case occurs if any of the characters included in F_1 (except "L") is the last character of a subword. The case is evident if three extreme points are detected in the x direction, where some or all have smaller x coordinates than the last point. These three points would belong to the last character. In this case we segment at a point

having the same x coordinate as the one, among the three extreme points, that has the largest x coordinate. This case is demonstrated in Fig. 4 where the only interesting character is the last one.

Having segmented the last character whether it belongs to the first or the second case, the segmentation process proceeds from the beginning of the subword (excluding the first character if it has been already segmented) until this segmentation point. Other characters at the end of subwords that belong to F_2 are processed differently in a latter stage.

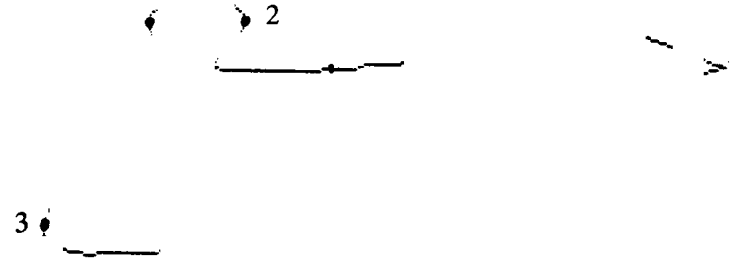

FIG. 4. Segmentation of last character for second special case.

Segmentation of Middle Characters

Having considered the above special cases, at the beginning and end of the subword, other general cases of characters that are within the subword or at its beginning or end are considered in a different manner. These characters are included respectively in M (shown in Fig. 1e), S_2 and F_2. First of all, all the extreme points in the x and y directions, for the remaining part of the subword, are detected and ordered descendingly according to their coordinates in the x direction.

The relative horizontal distance between each two consecutive extreme points is then calculated. If any of these relative distances is greater than 0.12 W, then a segmentation point may be marked at the middle point of the distance. This segmentation point, however, will be checked later for possible modification. On the other hand, if a relative distance does not exceed the threshold, we check the next distance. Thus, we continue comparing successive distances with the mentioned threshold until we reach a segmentation point. This whole process is continued until the subword is exhausted. This general case is shown in Fig. 5. Finally, we sequentially recheck each segmentation point for possible modification. If any character has a maximum or more in the y direction or it contains two or more minima and/or maxima in the x direction, then its sigmentation points are most likely correct. On the other hand, if none of the two conditions is satisfied for the character, then the segmentation point to the left of the character is moved to the left while the right segmentation point is kept at its place. We, now, recheck the character. If the character

does not satisfy the conditions, yet, then the left point is moved further to the left. This process is repeated until the segmented character satisfies, at least, one of the two conditions mentioned above. This process of modifying segmentation points continues until all segmented characters are verified and potentially modified.

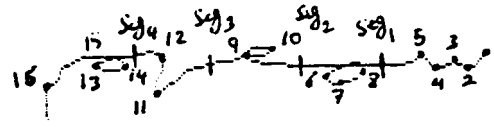

FIG. 5. Segmentation of middle characters for general cases.

Results and Conclusion

A flowchart for the whole segmentation technique is shown in Fig. 6. This technique has been implemented in "Turbo Pascal" version 4.0 with no optimization. The data set used in testing the segmentation technique contains about 200 different words. The words have been written by a single non-cooperative writer who has no familiarity with the recognition system, however, the writing was not completely free. The words have been entered using a GRAPHTEC KD 4030 graphic tablet where the x and y coordinates of each point were stored while the stylus is moving. Some of these words contain only one subword while other words may contain up to five subwords. Most of these words are entered in duplicate, triplicate or more. The total number of different subwords contained in this set is about one thousand. Using the developed segmentation technique, 99% of the subwords have been correctly segmented where a subword is incorrectly segmented if one character or more is incorrectly segmented.

The resulting characters are then classified using a technique developed by the authors[10]. Consequently, the recognition rate of the subwords approaches 98% where a subword is incorrectly recognized if one or more characters are misclassified.

The recognition system has been implemented on an IBM PC-XT with an 8088 microprocessor and on an IBM compatible (MICRONET) PC-AT with an 80286 microprocessor, 16 MHz clock (Turbo position). The time required to recognize a word (including segmentation and recognition of characters) depends on the word. This time, on the average, is 0.5 sec for the first hardware and is 50 msec for the second hardware.

The developed technique, in conjunction with the previously developed character recognition technique proved to be efficient and robust. Also, the size of the vocabulary may be easily increased since this does not affect segmentation efficiency.

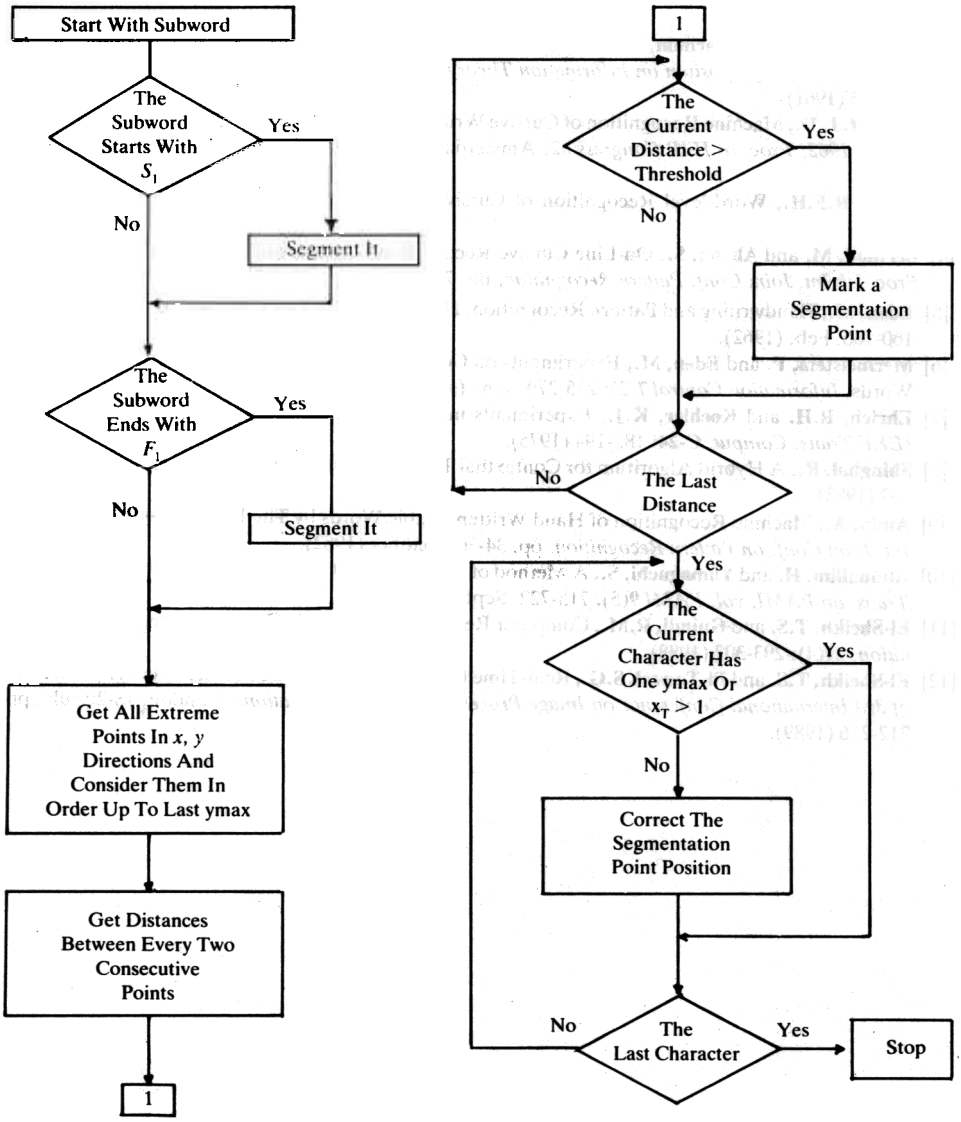

FIG. 6. A flowchart for the segmentation technique.

Acknowledgement

The authors would like to thank the department of Electronics & Communications, Cairo University, Egypt, for supporting the initial phases of this research. They are also grateful for the department of Computer Science, King Abdulaziz University, Saudi Arabia, for supplying the facilities for completing this work.

References

[1] **Frishkopf, L.S.** and **Harmon, L.D.**, Machine Reading of Cursive Script, *in:* **Cherry, C. (ed.)** *Information Theory, Symposium on Information Theory, London, 1960,* Washington, Butterworth, pp. 300-316 (1961).

[2] **Earnest, L.D.**, Machine Recognition of Cursive Writing, *in:* **Popplewell, C.M. (ed.)** *Information Processing 1962, Proc. of IFIP Congress 62,* Amsterdam, North-Holland Publishing Co., pp. 462-466 (1963).

[3] **Farag, R.F.H.**, Word-level Recognition of Cursive Script, *IEEE Trans. Compu.* **C-28:** 172-175 (1979).

[4] **Berthod, M.** and **Ahyan, S.**, On-Line Cursive Recognition: A Structural Approach with Learning, *Proc. 5th Int. Joint Conf. Pattern Recognition,* pp. 723-725, Dec. (1980).

[5] **Eden, M.**, Handwriting and Pattern Recognition, *IRE Transactions on Information Theory,* **IT-8**(2): 160-166, Feb. (1962).

[6] **Mermelstein, P.** and **Eden, M.**, Experiments on Computer Recognition of Connected Handwritten Words, *Information Control* 7(2): 255-270, June (1964).

[7] **Ehrich, R.H.** and **Koehler, K.J.**, Experiments in the Contextual Recognition of Cursive Script, *IEEE Trans. Comput.* **C-24:** 182-194 (1975).

[8] **Shinghal, R.**, A Hybrid Algorithm for Contextual Text Recognition, *Pattern Recognition* **16**(2): 261-267 (1983).

[9] **Amin, A.**, Machine Recognition of Hand Written Arabic Words by The IRAC II System, *Proc. 6th Int. Joint Conf. on Pattern Recognition,* pp. 34-36, October (1982).

[10] **Almuallim, H.** and **Yamaguchi, S.**, A Method of Recognition of Arabic Cursive Handwriting, *IEEE Trans. on PAMI, vol. PAMI* **9**(5): 715-722, September (1987).

[11] **El-Sheikh, T.S.** and **Guindi, R.M.**, Computer Recognition of Arabic Cursive Scripts, *Pattern Recognition,* **21**(4): 293-302 (1988).

[12] **El-Sheikh, T.S.** and **El-Taweel, S.G.**, Real-Time Hand Written Arabic Character Recognition, *Proc. of 3rd International Conference on Image Processing and Its Applications,* London, 18-20 July, pp. 212-216 (1989).

تقطـــيع الكلمـــات العربيـــة المكتوبـــة يدويًـــا

طلعت سالم الشيخ و صلاح جاد الله الطويل

قسم علوم الحاسب ، كلية العلوم ، جامعة الملك عبد العزيز ، جـــدة ، المملكة العربية السعودية ؛

و معهد بحوث الاتصالات ، القاهـــرة ، جمهورية مصر العربية

المستخلص . تهدف هذه المقالة إلى تصميم طريقة جديدة لتقطيع الكلمات العربية المكتوبة يدويًا فوريًا . ويتبع هذا التقطيع التعرف على الحروف ثم التعرف على الكلمة . وتبدأ الطريقة بفصل الكلمات ثم فصل أجزاء الكلمات . وتحتوى بعض الكلمات على أكثر من جزء كما أن بعض أجزاء الكلمة قد تحتوى على حرف واحد . ويحتوى كل جزيء من كلمة على خط رئيس وخطوط ثانوية مثل النقاط . الخطوة التالية في الطريقة هي معالجة الحالات الخاصة بالنسبة للحرف الأول ثم الحرف الأخير . وبعد الانتهاء من هذه الحالات الخاصة ، تبدأ معالجة الجزء المتبقي من الكلمة بطريقة أخرى كما يتم التأكد من مواضع التقطيع ويتم تعديل هذه المواضع في بعض الحالات . وقد تم اختبار هذه الطريقة باستخدام عدد كبير من الكلمات باحتمال نجاح حوالي ٩٩٪ . وقد تم الحصول على هذه النتيجة لأن الكلمات المستخدمة تم كتابتها بوساطة كاتب واحد دون حرية كاملة في الكتابة .