Ligature Segmentation for Urdu OCR

Gurpreet Singh Lehal
Department of Computer Science,
Punjabi University Patiala, Punjab,
India

***Abstract*–Urdu script uses superset of Arabic alphabet, but uses Nastaliq writing style. Nastaliq script is highly cursive, context sensitive and is written diagonally from top right to bottom left with stacking of characters, which makes it very hard to process for OCR. In addition, line and word segmentation are non-trivial tasks as we have frequently merging lines and vertically overlapping words and ligatures. But the real challenge is in character segmentation and so most of the researchers have taken the next higher unit, ligature, as recognition unit. A ligature is a connected component of one or more characters including diacritic marks and usually an Urdu word is composed of 1 to 8 ligatures. In this paper, we present a methodology for segmenting the Urdu text into ligatures. A hybrid approach, which uses top down technique for line segmentation and bottom up design for segmenting the line into ligatures, has been employed. The various challenges encountered during ligature segmentation such as horizontally overlapping and broken lines, merged ligatures and diacritic association have been discussed in detail.**

I. INTRODUCTION

Urdu language is one of the popular languages of Indian sub-continent. Urdu script uses superset of Arabic alphabet, but uses Nastaliq writing style, while Arabic uses Nashak style. Nastaliq script is highly cursive, context sensitive and is written diagonally from top right to bottom left with stacking of characters, which makes it very hard to process for OCR. In addition, line and word segmentation are non-trivial tasks as we have frequently merging lines and vertically overlapping words and ligatures and segmenting the characters is even more challenging, as the characters are completely merged and it is very difficult to detect the character segmentation points. Thus the next best option is to go for higher unit of recognition, which is ligature in case of Urdu [1-2]. A ligature is a connected component of characters along with associated dots and diacritic marks. A word in Urdu is a collection of ligatures and isolated characters.

In this paper, we have presented a methodology for extracting the ligatures from Urdu text image, which involves breaking page image into individual textlines and extracting and combining the related connected components to form the ligatures. Javed and Hussain[3] have presented a similar system for line and ligature segmentation and reported an accuracy of 94% for ligature separation. But their system has not handled some of issues such as overlapping textlines and the ligature segmentation accuracy needs improvement. In this paper we have presented a robust ligature segmentation technique which correctly segments the ligatures with 99.02% accuracy.

II. LINE SEGMENTATION

The global horizontal projection method is the most commonly used technique for line segmentation [3-5]. Although this global horizontal projection method is applicable for line segmentation of printed documents for other scripts, but for Urdu text images in many cases it results in over segmentation and under segmentation errors. Many times the ligatures in a line are arranged such that there is white space between the main bodies and the dots (and other diacritical marks) above and/or below the main bodies. As a result, the single text line is mis-segmented as two or three separate lines (Fig. 1).

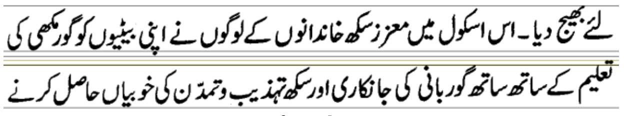

لئے بھیج دیا۔ اس اسکول میں معزز سکھ خاندانوں کے لوگوں نے اپنی بیٹیوں کو گورمکھی کی
تعلیم کے ساتھ ساتھ گوربانی کی جانکاری اور سکھ تہذیب وتمدن کی خوبیاں حاصل کرنے

Fig.1. Over segmentation of text lines

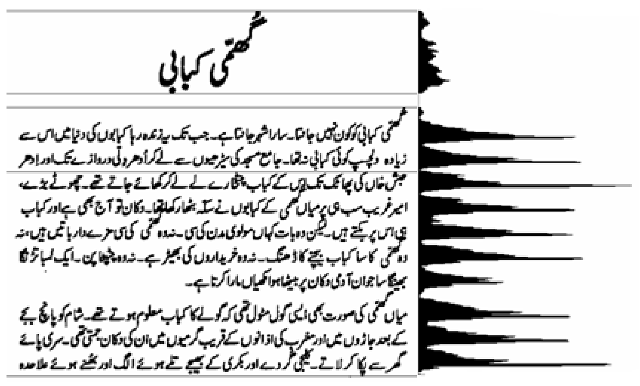


Fig.2. Merged text lines

Under segmentation error occurs due to multiple text lines getting merged because of touching ligatures in consecutive lines or horizontally overlapping lines. As seen in Fig.2, in second zone two text lines have been merged due to overlapping lines while in third zone eight text lines have been merged due to touching ligatures/overlapping lines in adjacent lines. Also there is no uniformity in line heights.

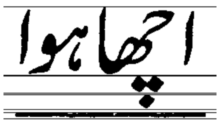


Fig.3. An underlined large sized text split into multiple zones

TABLE 1: ZONAL INFORMATION FOR IMAGE IN FIG. 3

Zone	**Height**	**Type**	**Density**
1	87	1	54
2	21	2	26
3	18	2	12
4	7	2	177

We propose the following modified horizontal projection profile algorithm for line segmentation:

a) Break the text image into horizontal zones using horizontal projection of pixels (Fig. 2).

b) *Estimate the row height and gap height between rows.* Row height was initially estimated as median of zone heights but it failed if majority of the zones contained multiple lines, as in Fig. 2. The image is divided into three zones with heights 254, 196 and 627 pixels respectively and the median value is 254. So we cannot depend solely on median of zone height (M1) to generate average row height. We also find the median of inter peak gaps (M2). Let M3=Median of row gap (at least three gaps needed, else take as 0). Then the row height is estimated as Min(M1, M2-M3), In above example, M1=254, M2=77 and M3=0 and so the average row height=77

c) *Label the zones,* based on their heights. Zones with height lesser than half of row height are labelled as Type 2. Type 2 zones could be : 1) zones containing dots/diacritic marks lying above or below the ligatures 2)zones containing underlines 3)zones containing small font sized text.

Zones with height greater than 1.5 times the row height are labelled as Type 3. Type 3 zones could be: 1) Zones containing multiple textlines 2) Zones containing single textline in large sized font.

Rest of the zones are labelled as Type 1.

d) *Merge the zones.* For type 2 zone (z2), select the nearest type 1 zone (z1) and merge the two, such that:

- Distance between z1 and z2 is lesser than row gap.
- Black pixel density of z2 is lesser than half of the black pixel density of z1. This is to ensure that z2 does not contain underline or text in smaller font. Black pixel density represents number of black pixels per row.

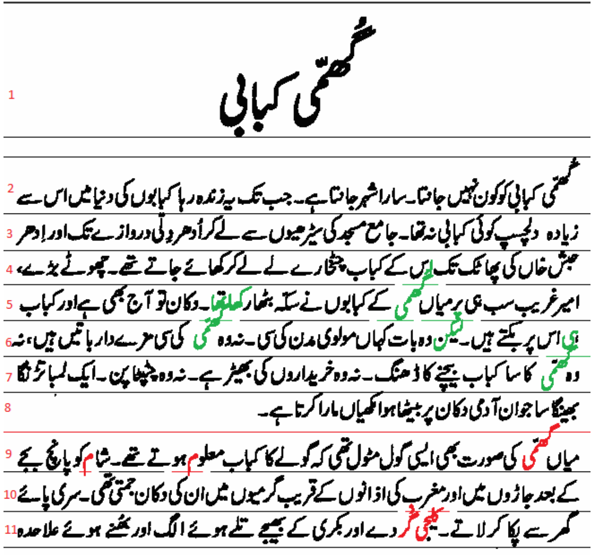
For example, consider a line containing a single underlined large sized text. The horizontal projection will break the headline into 4 zones (Fig. 3). Zone 1 contains the main text, while zone 2 and zone 3 contain the dots associated with the text above. Zone 4 contains the underline. The proper line segmentation routine should merge only zones1, 2 and 3. Before merging the zones, the zonal information is obtained(Table 1). After zone labelling, it is found that zone1 is type one zone, while zone 2, 3 and 4 are type two zones. That zones 2 and 3 are merged with zone 1 as their gap is lesser than row gap and density is also lesser than the threshold, but zone 4 is not merged as its density is too high.

e) *Split the merged textlines*. Next we analyse the type 3 zones. Type 3 zones could contain multiple textlines or large sized single text line. It is important to first identify if the zone contains multiple lines or single line. For this we use the row height information. We say that a type 3 zone is candidate for splitting if it meets following criteria:

- There exist valleys in the regions near the multiples of row heights.
- The black pixel density of proposed zone to be split should not be lesser than 50% of original zone.

In case, a type 3 zone meets the above requirements, we split the zone in the regions near the multiples of row heights.

The image in Fig. 2 has three zones with heights 254, 196 and 627 respectively. As the row height is estimated to be 77, so zone 1 has estimated 3 rows, zone 2 has 2 rows and zone 3 has 8 rows. But we find that in zone 1 there is only one valley, so at most two rows could be there. But when we try to split zone 1 at the valley, the split zone has low pixel density, so no splitting is done on the zone. While zone 2 and zone 3 are split around the valleys into two and eight rows respectively (Fig. 4).



1 گھمّی کبابی

2 گھمّی کبابی کو کون نہیں جانتا۔ سارا شہر جانتا ہے۔ جب تک یہ زندہ رہا کبابوں کی دنیا میں اس سے

3 زیادہ دلچسپ کوئی کبابی نہ تھا۔ جامع مسجد کی سیڑھیوں سے لے کر اُدھر دِلّی دروازے تک اور اِدھر

4 جنش خاں کی پھاٹک تک اس کے کباب چٹخارے لے لے کر کھائے جاتے تھے۔ چھوٹے بڑے،

5 امیر غریب سب ہی پر میاں گھمّی کے کبابوں نے سکّہ بٹھا رکھا تھا۔ دکان تو آج بھی ہے اور کباب

6 بھی اس پر بکتے ہیں۔ لیکن وہ بات کہاں مولوی مدن کی سی۔ نہ وہ گھمّی کی سی مزے دار باتیں ہیں، نہ

7 وہ گھمّی کا سا کباب بیچنے کا ڈھنگ۔ نہ وہ خریداروں کی بھیڑ ہے۔ نہ وہ چھچھا پن۔ ایک لمبا تڑنگا

8 بھینگا سا جوان آدمی دکان پر بیٹھا ہوا مکھیاں مارا کرتا ہے۔

9 میاں گھمّی کی صورت بھی ایسی گول مٹول تھی کہ گولے کا کباب معلوم ہوتے تھے۔ شام کو پانچ بجے

10 کے بعد جاڑوں میں اور مغرب کی اذانوں کے قریب گرمیوں میں ان کی دکان جمتی تھی۔ سری پائے

11 گھر سے پکا کر لاتے۔ کلیجی گردے اور بکری کے بھیجے تلے ہوئے الگ اور بھنے ہوئے علاحدہ

Fig.4. Image of Fig.2 after segmentation of merged lines

f) *Handle ligatures spread in multiple rows.* It is observed that in case of overlapping/touching lines, it could happen that ligatures in adjacent lines may be touching each other. (shown in green colour in Fig. 4) or a ligature may extend into its neighbouring text line, even though it may not be touching any ligature. (shown in red colour in Fig. 4). In first case, the ligature has to be segmented horizontally at appropriate location, while in second case we have to decide to which line the ligature belongs to. Also first of all decision has to be taken regarding the class of ligature. For this we determine the perimeter of the ligature bounding box. If the perimeter is greater than five times the average row height, then it is a multi-ligature shape and is candidate for segmentation. For such shapes, a horizontal cut is made along the row with minimum black pixels in the region {border row-1/6th of average line height, border row}. For single ligature shape, determine the line in which the ligature's major portion lies and the ligature is assigned to that row.

III. Connected Component Analysis

After line segmentation, the next step is ligature segmentation, where each line is split into ligatures. From OCR point of view, a ligature consists of one primary connected component and zero or more secondary connected components. The primary component represents the basic shape of the ligature, while the secondary connected component corresponds to the dots and diacritics marks and special symbols associated with the ligature. As for example, the ligature in Fig. 5a, has one primary connected component (Fig. 5b) and five secondary connected components (Fig. 5c).

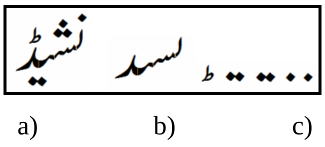


Fig.5. Urdu Ligature segmented into primary and secondary components

While a top down approach is followed to split the image into text lines, the text lines are segmented into ligatures using connected component based bottom up technique. The text line is examined from right to left and all the connected components are stored in a linked list. Each connected component could represent either a primary or a secondary connected component of a ligature. A primary component can have zero or more associated secondary components but a secondary component will have only one associated primary component. After collecting the connected components, the next task is to form back the complete ligature by identifying the primary components and its associated secondary components. For this purpose, we need to perform two operations: a) Classifying the connected component as primary or secondary component. b) Associating the secondary components with their primary components

After a careful analysis of the shapes and positions of the two components, we have developed a simple classifier, which uses six hand crafted features. We shall be using the terms vertically overlapping and enveloping in following section. We say component c1 envelopes component c2, if (min x_{c1} < =min x_{c2}) and (max x_{c1} >= max x_{c2}). Component c1 and component c2 overlap vertically, if either ((min x_{c1} <= min x_{c2}) and (min x_{c2} < max x_{c1} <= max x_{c2})) or((min x_{c1} >= min x_{c2}) and (min x_{c1} < max x_{c2} <= max x_{c1})).

The six binary features are:

A. *Component touches the baseline*

Every Urdu ligature and so ideally the primary component will be touching the baseline, while the secondary components will be lying above or below the baseline. An analysis of Urdu image revealed that though 99.11% of primary components touch the baseline but 9.81% of secondary components also touch the baseline. As, we can see in Fig. 6, the components which touch the baseline are shown in red, while others are shown in black. We can see all the primary components are marked in red, but there are two secondary components too, which are touching the baseline and are displayed in red colour.

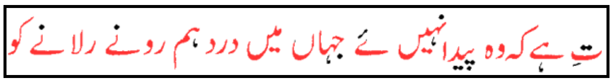


Fig.6. Primary and secondary components touching the baseline

B. *Black pixel density of component > 50%*

We determine the ratio of black and white pixel in the bounding box of the component. If the ratio is greater than 1, then the feature is true else it is false. In Fig. 7, we depict the components in red, for which the feature is true. From analysis it was found that this feature is true for 92.49% of secondary components and 17.57% of primary components.

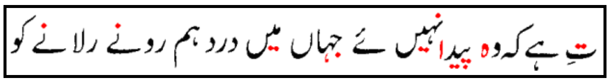


Fig.7. Components in red have greater than 50% black pixel density

C. *Height of the component < one-fifth of row height*

Generally, the size of secondary component is smaller than primary components. The next feature, uses this property and is true if the component height is lesser than one-fifth of row height. We found that this feature is true in 97.08% of secondary components, but 5.83% of primary components also have smaller heights.

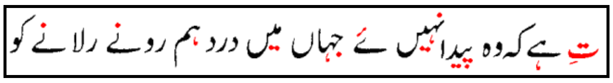


Fig.8. Components in red have height lesser than one-fifth of row height

D. *Component is vertically enveloped by some other component*

The secondary components are usually vertically overlapped by their concerned primary components. In fact 80.36% of the secondary components are completely overlapped vertically by some primary components, while 1.72% of smaller sized primary components are also covered by larger sized primary components.

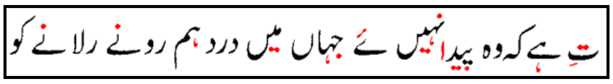


Fig.9. Vertically enveloped components displayed in red

E. *Component lies completely in upper half or lower quarter:*

Usually the secondary components lie completely either in upper half or lower quarter of the text line. We found that 72.07% of secondary components lie in these regions, while only 0.95% of primary components are found in these regions.

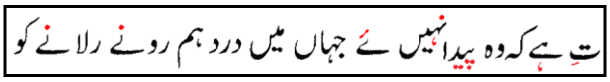

ت ہے کہ وہ پیدا نہیں ہے جہاں میں درد ہم رونے رلانے کو

Fig.10. Components lying in upper half/ lower quarter shown in red.

F. *Component contains a horizontal row made up of only black pixels:*

The primary components are usually large sized written along the diagonal, while the secondary components are usually small sized components written in horizontal direction. This feature searches for a horizontal line running across the bounding box of the component made up only of black pixels. Such a line was found to be present in 85.35% of secondary components while 8.73% of primary components also had at least one such row.

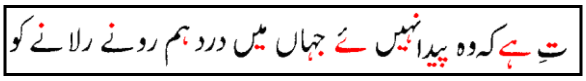

ت ہے کہ وہ پیدا نہیں ہے جہاں میں درد ہم رونے رلانے کو

Fig.11. Components containing a horizontal row of black pixels displayed in red

We performed a statistical analysis on 10,215 connected component images to determine the frequency of occurrence of the above features. The results are tabulated in Table 2.

TABLE II: FREQUENCY OF CLASSIFICATION FEATURES IN PRIMARY AND SECONDARY COMPONENTS

Feature	Secondary Component (psc)	Primary Component (ppc)	Secondary Component (asc)	Primary Component (apc)
	Present		Absent	
1	9.81%	99.11%	90.19%	0.89%
2	92.49%	17.57%	7.51%	82.43%
3	97.08%	5.83%	2.92%	94.17%
4	80.36%	1.72%	19.64%	98.28%
5	72.07%	0.95%	27.93%	99.05%
6	85.35%	8.73%	14.65%	91.27%

We have designed a simple classifier which uses the feature vector and these probabilities to decide the component class. From the table we can see that if for a component, feature 1 is true, then the probability of being primary component is .991 and probability of being secondary component is .0981. Similarly if in the same component feature 3 is missing, then the probability of it being primary component is .9417 and probability of being secondary component is .0292. We combine the probabilities for the above six features and design the following simple classification algorithm:

```
sc=pc=0;
for i=1 to 6
If(feature[i]==1)then sc=sc+psc[i]; pc= pc+ppc[i]
Else  sc=sc+asc[i] ; pc= pc+apc[i]
if(sc>pc) then component=secondary component else component = primary component
```

Thus if for example, the feature vector of a component is [0,1, 1, 0, 1, 0] then by above algorithm:

pc = 0.89+17.57+5.83+98.29+0.95+91.27 = 214.8
sc = 90.19+92.49+97.08+ 19.24+72.07+14.65 = 385.72

As sc > pc, thus the component is identified as secondary component

Fig. 12 depicts a text image separated into primary and secondary components. The primary components are shown in black colour and secondary components in red colour. It should be noted that punctuation marks such as hyphen, comma, full stop and quotes are also considered as secondary components, but they should not be associated with any primary components.

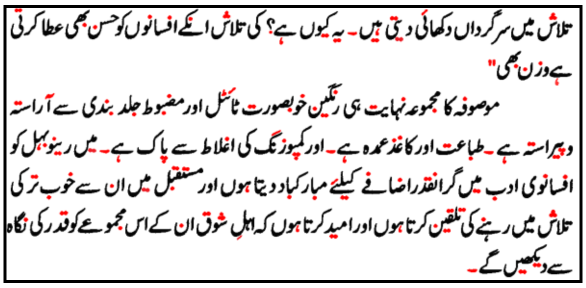

تلاش میں سرگرداں دکھائی دیتی ہیں۔ یہ کیوں ہے؟ کی تلاش اکے افسانوں کو حسن بھی عطا کرتی
ہے وزن بھی"
موصوفہ کا مجموعہ نہایت ہی رنگین خوبصورت ٹائٹل اور مضبوط جلد بندی سے آراستہ
وپیراستہ ہے۔طباعت اور کاغذ عمدہ ہے۔اور کپوزنگ کی اغلاط سے پاک ہے۔میں ریزبھل کو
افسانوی ادب میں گرانقدر اضافے کیلئے مبارکباد دیتا ہوں اور مستقبل میں ان سے خوب ترکی
تلاش میں رہنے کی تلقین کرتا ہوں اور امید کرتا ہوں کہ اہلِ شوق ان کے اس مجموعے کو قدر کی نگاہ
سے دیکھیں گے۔

Fig.12. Urdu Ligatures segmented into primary and secondary components

IV. COMBINING THE COMPONENTS

The secondary and primary components need to be combined to generate the complete ligature. For this purpose, the secondary components need to be assigned to the appropriate primary component. This is a non-trivial issue in Arabic script, as the secondary components vertically lie inside the associated primary component, but in Urdu we find a single secondary component could be completely enveloped/ vertically overlapping with multiple primary components (Fig.13).

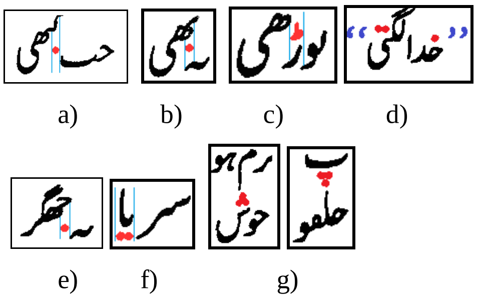


Fig.13. Primary and secondary components

Once the primary and secondary components are identified, for each secondary component, we identify all the primary components, which overlap or envelop that secondary component. Let *n* be the number of primary components vertically enveloping the secondary component and *p* be the number of primary components vertically enveloping the secondary component.

The following rules are used for component assignment:

If (*n*=1) then assign the secondary component to the enveloping primary component (Fig. 13a).

if *n*>1 (Fig. 13b), then find the distance between the secondary component and the enveloping primary components. Select the primary component with least distance. The distance is measured by finding the weighted shortest distance between the boundary black pixels. We assign smaller weight to vertical distance and more weight to distance on right side, by multiplying vertical distance with 0,75 and distance on right side with 1.25. The logic for taking weighted distance instead of actual distance is that the probability of the secondary component lying above or below the primary component is more than on lying on the right side of the primary component.

if (*n*=0) and (*p*>0), then assign the component to the vertically overlapping primary component with least weighted distance. Thus if the actual distances between a secondary component and its three enveloping primary components lying below, left and right are 5, 5 and 9 pixels respectively (Fig. 13c). But as the weighted distances are 3.75, 5 and 11.25 respectively, so the secondary component is assigned to the primary component lying below it.

If (n+*p*=0), then the secondary component will be considered as a punctuation mark and will not be assigned to any primary component. As shown in Fig. 13d, the primary components are in black colour, while the secondary components are in red colour. The secondary components for which both *n* and *p* have values zero are considered as punctuation marks are depicted in blue colour.

We always give first preference to enveloping components over overlapping components. So even though, an overlapping primary component is nearer than the enveloping primary component, we will still assign the secondary component to the enveloping primary component (Fig. 13e). Similarly, the overlapping primary component is given preference over primary components which have no common x-coordinates with the secondary component (13f).

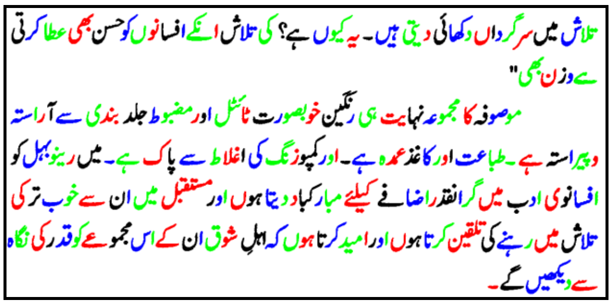
In case of overlapping text lines, it could happen that primary and secondary components maybe lying in different text lines (Fig. 13g). So we not only look for overlapping primary components in same line but also in adjacent lines too. In that case, we not only look for vertical gap with overlapping primary components, but also look at shape of secondary components. There are certain primary components such as three dots, which appear only above primary component, while the inverted three dots appear only below the primary components. Thus as shown in Fig. 13g, the three dotted secondary component is assigned to lower lying primary component, while the inverted three dotted secondary component is assigned to above lying primary component.

In Fig. 14, we see the text shown in Fig. 12 split into ligatures, after associating the secondary components with their primary components, using the above rules.

VI. Experimental Results

The line segmentation algorithm was tested on 448 text images and it successfully segmented 99.11% pages correctly. It was observed that 67.44% of pages contained at least one broken line, while 50% of pages contained at least one zone containing horizontally overlapping lines. At text line level, we found that 20.75% of text lines were overlapping and 8.84% of text lines were broken. There were only 12.79% pages which do not have any overlapping or broken lines.

We also analysed 45 Urdu images to study about the count of associated primary and secondary components. The images had 42,441 connected components, out of which 55.5% of the components were primary components and rest were secondary components. It was found that 93.22% of secondary components had no confusion in assigning to appropriate primary component, as there was only one overlapping/enveloping primary component. For 6.51% of secondary components, there were two overlapping primary components, while 0.27% of secondary components had three overlapping/enveloping primary components. The connected components were classified and associated correctly with 99.02% accuracy to form the ligatures.



تلاش میں سرگرداں دکھائی دیتی ہیں۔ یہ کیوں ہے؟ کی تلاش انکے افسانوں کو حسن بھی عطا کرتی
ے وزن بھی"
موصوفہ کا مجموعہ نہایت ہی رنگین خوبصورت ٹائٹل اور مضبوط جلد بندی سے آراستہ
و پیراستہ ہے۔ طباعت اور کاغذ عمدہ ہے۔ اور کمپوزنگ کی اغلاط سے پاک ہے۔ میں رینو بہل کو
افسانوی ادب میں گرانقدر اضافے کیلئے مبارکباد دیتا ہوں اور مستقبل میں ان سے خوب تر کی
تلاش میں رہنے کی تلقین کرتا ہوں اور امید کرتا ہوں کہ اہلِ شوق ان کے اس مجموعے کو قدر کی نگاہ
سے دیکھیں گے۔

Fig.14. Associating primary and secondary components

VII. Conclusion

In this paper we have presented a methodology for ligature segmentation, which was tested to classify the components correctly with 99.02% accuracy. A line segmentation algorithm for Urdu documents, which takes care of over segmentation and under segmentation issues and successfully segmented 99.11% pages correctly has also been presented.

References

[1] S. A.Sattar, S. Haque, Pathan, M. K. and Q. Gee, "Implementation Challenges for Nastaliq Character Recognition," In: *International Multi Topic Conference (IMTIC'08)*, 11-12 April 2008, Jamshoro, Sindh, Pakistan.

[2] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil and H. Moin, "Segmentation Free Nastaliq Urdu OCR," Proceedings of World Academy of Science, Engineering and Technology, 46, pp. 456-461. 2010.

[3] S.T. Javed, S. Hussain, "Improving Nastaliq Specific Pre-Recognition Process for Urdu OCR", In the Proceedings of 13th IEEE International Multitopic Conference 2009 (INMIC 2009), Islamabad, Pakistan. pp 1-6.2009.

[4] U.Pal and Anirban Sarkar, "Recognition of Printed Urdu Script", Proceedings of 7th Int. Conf., on Document Analysis and Recognition, pp.1183-1187, 2003.

[5] Noor Ahmed Shaikh, Ghulam Ali Mallah, Zubair A. Shaikh,"Character Segmentation of Sindhi, an Arabic Style Scripting Language, using Height Profile Vector", Australian Journal of Basic and Applied Sciences, 3(4): 4160-4169, 2009.