WORD SEGMENTATION PROBLEM IN URDU TEXT AND LANGUAGE MODEL BASED SOLUTION

Umrinderpal Singh

Punjabi University, Patiala

umrinderpal@gmail.com

Abstract

Word Segmentation is the primary and most significant task of Natural Language Processing (NLP) applications. To comprehend any Natural Language text we need to split it into individual tokens known as words and by processing individual words one by one, we get a sense of any given text. Asian languages are resource poor languages, but nowadays these languages get attention from NLP researcher, and researcher provides various algorithms to handle these morphological rich languages. Word segmentation is a preliminary phase but a challenging problem in NLP research area, especially when we deal with Indo-Aryan scripts like Urdu, which is written from right to left, and use Arabic script. Urdu word Segmentation problem deals on two stages, space omission and space insertion problem. We show how we can deal with these issues in Urdu. Until now word segmentation methods for Indo-Aryan or related group languages was constructed on resources like POS tagger, stemming and dictionaries. We have evaluated Language Model based method to resolve this problem and able to get 93.3% accuracy.

1. Introduction

Abstract- Word Segmentation is the primary and most significant task of Natural Language Processing (NLP) applications. To comprehend any Natural Language text we need to split it into individual tokens known as words and by processing individual words one by one, we get a sense of any given text. Asian languages are resource poor languages, but nowadays these languages get attention from NLP researcher, and researcher provides various algorithms to handle these morphological rich languages.

Word segmentation is a preliminary phase but a challenging problem in NLP research area, especially when we deal with Indo-Aryan scripts like Urdu, which is written from right to left, and use Arabic script. Urdu word Segmentation problem deals on two stages, space omission and space insertion problem. We show how we can deal with these issues in Urdu. Until now word segmentation methods for Indo-Aryan or related group languages was constructed on resources like POS tagger, stemming and dictionaries. We have evaluated Language Model based method to resolve this problem and able to get 93.3% accuracy.

Urdu Sentence [Non-segmented]	**Urdu Sentence [Segmented]**
قافلےکےصدراحمدشیرڈوگرانےکہا	قافلے کے صدر احمد شیر ڈوگرا نے کہ
*(Hindi Translation) काफलेकेसदरअहमदशियरडोगरानेकहा	*(Hindi Translation) काफिले के राष्ट्रपति अहमद शेर डोगर ने कहा
⁎(Romanization) K āphalēkēsadara'aha ḍōgarānēkahā	**⁎(Romanization) K āphilē kē rāṣṭrapati ahamada** śēra ḍōgara nē kahā
⁎(Translation) Qaflykysdrahmdsyrdugranykha	**(English Translation) Leader of Troop Ahmad Sher Dogra Said**

Table 1: Translated by Google Translated *https://translate.google.co.in/*

2. Related Work

Asian Languages like Hindi, Punjabi, Gujarati, and Bengali and European languages like English, French and Geek etc used white space and punctuation to isolate words in sentences. Language like Urdu, Arabic, Farsi, Persian and Chinese etc. are enormously affected by word segmentation problem because of its traditional written style where space is optional in a hand-written text.

Many types of research have been worked on word segmentation issues in different Asian languages. Urdu is still new in the field of Natural Language processing; therefore, much less work has been done for Urdu. To resolve segmentation issues, dictionary based approach is very prevalent among all the researchers, Longest matching string technique had been tried by (Poowarawan, 1986; Rarunrom,1991), Maximum matching technique (Sproatet al., 1996; Haizhou & Baosheng, 1998) comes under dictionary based approach and accuracy directly depends upon size of the dictionary. N-Gram based approached used by (Chang et al., 1992; Li Haizhouet al., 1997; Richard Sproat, 1996; Dai & Lee, 1994; Aroonmanakun, 2002), this approach also depends on the size of lexicon collection. (GS Lehal,2009,2010) discussed Rules based approach for space omission and insertion. The module was used for Urdu to Hindi transliteration system and depends on Urdu and Hindi dictionaries. (Durrani and Hussain2010) proposed system to resolve word segmentation issues in Urdu, knowledge based approach was used in the context of unigram, bigram and script knowledge. (M Akram, S Hussain;2010) proposed segmentation module for Urdu OCR, based on ligature and word n-grams.

As we examined, there much less work has been done for Urdu. Most of Urdu word segmentation work was done in the context of Urdu OCR. Urdu is a resource poor language as compared to European language. We have done efforts to collect a large Urdu Corpus and tried to resolve fundamental issues like word segmentation, which is a key process for any NLP application.

3. Segmentation issues in Urdu

Urdu is written and spoken in Pakistan and India by 100 million people (Durrani and Hussain2010). Urdu is Indo-Aryan language written in Arabic script. Where Arabic script is written horizontally, Urdu writing style is diagonal, known as Nastaliq. Nastaliq means latter appear to float or hang across the page. Nastaliq style words are so closely written that make space become optional in the hand-written text. Some Parso-Arabic languages like Arabic, Persian, Farsi, words can be written in continuum manner without using any space in-between words. Unlike other Asian languages where space is used to define word boundaries but does not infer boundary condition every time. Urdu Segmentation problem can be considered on two levels, space insertion, and space omission. Space omission problem deals with those words that can be joined after omitting space in-between words and retain its original meaning. For example:

Table2: Space omission examples

Before Concatenation	After Concatenation	English Meaning
لگا تار(Lgạ tạr)	لگاتار(Lgạtạr)	Continue
گر دی(Gr dy̱)	گردی(Grdy̱)	Done
کر تے(Kar atē)	کرتے(Karatē)	Do
آپ کا(Ập ḵạ)	آپکا(Ậpḵạ)	Yours

On the other hand, space insertion problem deals with inserting space to break the word string into two or more distinct words so, that reader can read isolate words in place of one long word string. For example:

Table3: Space insertion examples

Before Disjoint	After Disjoint	English Meaning
اورمعاشرتی(Ạwrmʻạsẖrṯy)	اور معاشرتی(Ạwr mʻạsẖrṯy)	**and social**
لیےناگزیرہے(La'ēnạgẕyrhē)	لیے ناگزیر ہے(Li'ē nạgẕyr hai)	**...is essential for**
کےپہلےدستےکی(Kēpahalēdasatēkī)	کے پہلے دستے کی(Kē pahalē dastē kī)	**first troop**
اوراِسی(Ạwrạisy̱)	اور اِسی(Ạwr ạisy̱)	**and this**

Space insertion and omission problem did not create any problem when we consider hand-written text where white space is an optional to isolate words. Readers are used to reading such text very fluently. White spaces separating word boundaries only had been considering when we typed Urdu text on the computer. On computer text editors, you cannot neglect any white space between isolate words otherwise shape of the word completely changes and it looks erroneous to the reader, which is not the case of hand-written text where chars retain its original shapes. Urdu Unicode encoding system has its feature to combine to different chars of the script and change its shape, but internally its original

Unicode remains unchanged only visible shape transformed. For example, word بھیموجو (Translation: “also present”) changed from بھی موجو(Translation: “also present”) visually incorrect.

In Urdu, we have two type of script letters, One is Connectors and other are Non-Connecters, as described following.

Table4: Connecter alphabets

Connecter								
خ	ح	چ	ج	ث	ٹ	ت	پ	ب
ف	غ	ع	ظ	ط	ض	ص	ش	س
ی	ھ	ہ	ن	م	ل	گ	ک	ق

Table5: Non-Connecter alphabets

Non-Connecter								
ء	ژ	ز	ڑ	ر	ز	ڈ	د	ا
						ے	و	آ

Connecter letters are those, which can connect to its next letter and change its original shape based on its position like if letter is at first place, intermediate or in end of the word. For Example, following chars are same but shapes are different based on its position, every connecter type letter can be in four shapes:

Table 6: Shape of a connecter letter based on different positions

Position	Example Word 1	Example Word 2
Start	جل	قریب

Intermediate	پنجاب	تعلقات
End	فوج	استعمال

Non-Connecter letter retains its original shape by not joining to its next char in word, whether it is on any position. For example:

Table 7: Shape of a Non-connecter letter based on different positions

Position	Example Word 1	Example Word 2
Start	دہشت	زندگی
Intermediate	اداروں	وزیر
End	انسداد	نواز

On computer typed text, where typist used Unicode for typing and used to insert space to isolate two words if ends with connector letters, otherwise word looks incorrect to the reader or may change its meaning. Space omission problem arises when we deal with proper nouns like the name of locations ends with حیدر آباد (Hyderabad: bad), رام پور (Rampur: pure), رام گڑھ (Ramghar: ghar) and few other frequently used words. Otherwise, we do not need to omit space. These location names can be written by joining both words or in an individual way but retain its original meaning. Urdu writers are used to writing location name in this isolated way and reader are very familiars with this style. Sindhi, Shamukhi languages are closing related language of Urdu also used the same style to write location names. Therefore, we cannot treat these words under space omission problem. We can say that by using the Unicode encoding, space omission problem, no more challenging part of Urdu segmentation problem. Following are few examples of space omission problem, which are perfectly fine in both ways, with space or without space. We have done frequency analysis of these bigram words, which demonstrates that this style is popular among Urdu writers, there is no need to omit space because of its acceptance in community, and these words retain their same meaning in both ways.

Table 8: Frequency analysis of space omission candidates

Concatenated form	Frequency	Isolated form	Frequency

وزیراعظم(Wẕyrạʿẓm)	1229	وزیر اعظم(Wẕyr ạʿẓm)	1140
جائینگے(Jā'ēṅgē)	311	جائیں گے(Jā'ēṅ gē)	1333
کیاگیا(Ḵyạgyạ)	566	کیا گیا(Ḵyạ gyạ)	39 17
ہرحال(Harahāla)	1420	ہر حال(Hara hāla)	734

The problem only arises in the case of space insertion, when we deal with words those ends with non-connecters and no white space inserted to make the words isolate from each other. Urdu readers can read those sentences easy because their brains are trained to automatically tokenize that long non-segmented word string into meaningful words, but a computer cannot process them and not able to generate any information based on them. For example:

Table 9: Non-segmented and Segmented Word Examples

Non-Segmented Words	Segmented Words	English Translation
ہےاوراسی(H ēāō rās	ہے اور اِسی(H ē Aura isī	and this
نےکہاہےکہ(N ēkahāhēke	نے کہا ہے کہ(N ēkah āh ēke)	said that
کرنےوالےاداروں(Karan ēōā	کرنے والے اداروں(Karan ē ōālē ādārōṅ)	agencies

4. Proposed Algorithm

We have developed word segmentation algorithm based on N-gram language model. The language model is used to generate natural language text. We have used Trigram Language model where n=3. N-Gram language model is based on Markov assumption, where the current state is dependent on previous states.

$$P(w_1, w_2 \dots w_n) \approx \prod p(w_i | w_{i-k} \dots \dots w_{i-1})$$

Or

$$P(w_i | w_{1,} w_2 \dots w_{i-1}) \approx \prod p(w_i | w_{i-k} \dots \dots w_{i-1})$$

Simplest model is Unigram model. Defined as:

$$P(w_1, w_2 \dots w_n) \approx \prod p(w_i)$$

Estimated as:

$$P(w_i) = \frac{Count(w_i)}{Count(*)}$$

In bigram model, current state or word is dependent on its immediate previous two states or words, defined as:

$$P(w_1, w_2 \dots w_n) \approx \prod p(w_i|w_{i-1})$$

Estimated as:

$$P(w_i|w_{i-1}) = \frac{Count(w_{i-1}, w_i)}{Count(w_{i-1})}$$

In trigram model, current state or word is dependent upon previous two states, defined as:

$$P(w_1, w_2 \dots w_n) \approx \prod p(w_i|w_{i-1}, w_{i-2})$$

Estimated as:

$$P(w_i|w_{i-1}w_{i-2}) = \frac{Count(w_{i-2}w_{i-1}, w_i)}{Count(w_{i-2}w_{i-1})}$$

We have used linear interpolation for better estimation, which take into its account trigram bigram and unigram estimates. Liner interpolation gives us weighted average of three different estimates. Liner interpolation is defined as:

$$P(w_i|w_{i-2}w_{i-1}) = \lambda * q(w_i|w_{i-2}w_{i-1}) + \lambda * q(w_i|w_{i-1}) + \lambda * q(w_i)$$

Where

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \text{ and}\lambda_i \ \geq 0\, for\ all\ i$$

Based on non-connecting letters, the algorithm tries to break char string in into different words ends with the non-connecting letter. Urdu has 12 non-connecters; max number of words can be generated based on length of the non-segmented word string. During the testing phase, we got the max number of possible generated words were 20. For example, following Non-segmented word string divided into 8 different possible words:

Table 10: Possible word division

Input String	اوروزیراعلی
First Token	اوروزیراعلی
Second Token	اور
Third Token	اوروزیر
Fourth Token	او
Fifth Token	اورو
Sixth Token	اوروز
Seventh Token	ا
Eighth Token	اوروزیرا

To break the string of chars based on non-connectors, defined as:

$$W\colon w_{i=0} \ \ldots . \, to \ \ldots \ w_{i=n}$$

If we have $w_{i=n}$ length word string, then it can be divided into different words stating from $w_{i=0}$ to occurrence of non-connecter letter. Following algorithm shows word boundary identification procedure:

Algorithm: Word Boundary

```
Array: NewPossibleWords[ ]
Loop: NC in NCArray[ ]
    Loop: char w in WordString
        If: x==NC
            Append: NewPossibleWords[ ]
            Continue
    EndLoop: w in WordString
EndLoop: NC in NCArray
```

To find the most likely word sequence, arg-max has been taken;

$$f(x) = arg\ max_x\ p(y|x)$$

Where X is different generated words and Y is dependent words to estimate the most likely sequence.

Algorithm: Most likely sequence

```
Read: input Urdu Text
    Words[ ]: Tokenize into words based on space
    Loop: Words[ ]
        If word in Words[ ]  is unknown in unigram dictionary.
            NewPossibleWords[ ]: Generate all possible words, based on non-connectors.
            Loop: NewPossibleWords[ ]
                Find Most likely sequence.
            End Loop:NewPossibleWords[ ]
    End Loop: Words[ ]
```

5. Result and Evaluation

The algorithm is completely based on Language Model, Therefore, to estimate language parameter we have used raw corpus of Urdu Unicode text related to several news domains. Raw corpus has been collected from BBC Urdu News website. Corpus contain following Unigram, Bigram and Trigrams after removing numbers and symbols. Following table shows the detail of training corpus.

Table 11: Frequency analysis of corpus

Unigram	156368
Bigram	1334219
Trigram	3071468
Total Sentences	353844

The Algorithm was tested on 1000 sentences taken from BBC Urdu website related to political news. For testing proposes test data contains few errors, which was not adequate examples to test algorithm on every aspect. Therefore, we have deliberately tried to manipulate sentences on various levels and join different words together, for example.

تقریب میں فوج کے سربراہجنرل راحیل شریف اوروزیراعلیٰپنجاب میاں شہبازشریف بھی موجودتھے ۔

All the testing data had created by Urdu linguist. After manipulating the sentences, there are 2078 candidate examples for segmentation, average two error words in each sentence. Our algorithm, able to detect all errors in given sentences because in such large training corpus it did not find any evidence of those 2078 word. We have evaluated the performance of algorithm based on standard metrics recall, precision and F1-Measure.

Recall: Relevant information extracted from text.

Recall defined as:

$$\text{Recall} = \frac{\text{No. of correct answers given by system}}{\text{Total No. of possible correct answers in text}}$$

Precision: **Actual correct answers returned by system.**

$$\text{Precision} = \frac{\text{No. of correct answer}}{\text{No. of answers given}}$$

F-Measure: **Balances of Recall and Precision by using a parameters** β.

The F-measure is defined as:

$$\text{F} - \text{measure} = \frac{(\beta 2 + 1)\ \text{RP}}{(\ \beta 2\text{P} + \text{R})}$$

β is weighted as β=1. **When** β=1 F**-measure is called F1-measure.**

The F1-measure is defined as:

$$\text{F1} - \text{measure} = \frac{2 * \text{RP}}{\text{P} + \text{R}}$$

Table 12: Possible word division

Test Data	**Recall**	**Precision**	**F1-Measure**
2078 Non-Segmented tokens	0.996	0.876	0.933

In proposed algorithm, we have able to achieve 99.6% accuracy in error detection rate, and 93.3% overall accuracy. Algorithm failed to generate correct tokens only in non-segmented person names, which were not part of training data and algorithm spilt person name into two words, and additional minor errors found where algorithm done over segmentation of a give token. The proposed algorithm is fast enough as compared to other approaches, where stemming and tagging process is required. This

algorithm can be part of any major Urdu NLP application like machine translation or part of speech tagging etc., to handle tokenization process of to check unknown words for segmentation issue.

6. Conclusion

The work presented in this paper was the effort to develop on an algorithm that is capable to tokenize any Urdu sentence into isolated words by detecting words boundaries in non-segmented word string. Word tokenization process is the fundamental and most important phrase of any NLP application. Any NLP application's accuracy depends on its basic tokenization task. We have projected algorithm based on Language Model, which is able to tokenize sentence into isolated words with the accuracy of 93.3%. As compared to other algorithms which are reliant on resources like Urdu dictionary, tagging, and morphological features, our proposed algorithm can easily use in any NLP application where Language is already part of the architecture like statistical machine translation, statistical part of speech tagger and statistical named entity recognition system etc.

REFERENCES

[1] Aroonmanakun, W. *Collocation and Thai Word Segmentation.* In Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop, pp.68-75. Pathumthani. (2002).

[2] Chang, Jyun-Shen, Chen, S.-D., Zhen, Y., Liu, X.-Z., & Ke, S.-J . *Large-corpus-based methodsfor Chinese personal name recognition.* Journal of Chinese Information Processing , pp. 7-15. (1992)

[3] Durrani, Nadir, and Sarmad Hussain. "Urdu word segmentation." *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. (2010)

[4] Haizhouet, L., & Baosheng, Y. *Chinese Word Segmentation*. In Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation, pp. 212-217. (1998)

[5] Jurafsky, D., & Martin., I. J *Speech and LanguageProcessing: An Introduction to Natural Language Processing (1st ed.)*. Computational Linguistics and Speech Recognition Prentice Hall. (2000)

[6] Lehal, Gurpreet Singh. "A word segmentation system for handling space omission problem in Urdu script." *23rd International Conference on Computational Linguistics*. (2010)

[7] Lehal, G . *A Two Stage Word Segmentation System for Handling Space Insertion Problem in Urdu Script*. World Academy of Science, Engineering and Technology 60. (2009)

[8] Misbah Akram, & Sarmad Hussain. "Word segmentation for urdu OCR system." *Proceedings of the 8th Workshop on Asian Language Resources, Beijing, China*. (2010)

[9] Poowarawan, Y. *Dictionary-based Thai Syllable Separation*. Proceedings of the Ninth Electronics Engineering Conference. (1986)

[10] Richard Sproat, C. S. *A Stochastic FiniteState Word-Segmentation Algorithm for Chinese*.Computational Linguistics. (1996)

[11] Sproat, R., Shih, C., Gale, W., & Chang, N. *A Stochastic Finite-State Word-Segmentation Algorithm for Chinese*. omputational Linguistics. (1996)