ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106372, December 2016* ISSN (Online) : 0974-5645

Automatic Text Correction for Devanagari OCR

Atul Kumar* and Gurpreet Singh Lehal

Department of Computer Science, Punjabi University, Patiala – 147002, Punjab, India;
atulkmr02@gmail.com, gslehal@gmail.com

Abstract

Objectives: This paper proposes a new technique for correcting errors done by Devanagari OCR (Optical Character Reader) system based on confusion matrix. **Methods/Statistical Analysis:** Confusion matrix is generated from large corpus of Hindi. The system takes each word of OCR output and generate number of strings from topmost five confused characters for each character of input word along with probability of these strings for ranking. Each string is validated with the character trigram dictionary and these valid strings are used for best suggestions. **Findings:** The topmost five words is taken as suggestions. The system has been tested for variety of OCR outputs documents of Devanagari script. The system provides suggestions for all the correct words at top position. For more than 10000 unique words in Devanagari OCR output, system gives the accuracy of 97%. **Application/Improvements:** This system is used in post-processing of Devanagari OCR. With some improvements, the system can also be used for Gurumukhi Script and Urdu script.

Keywords: Automatic Text Correction, Confusion Matrix, Devanagari, OCR, Trigram

1. Introduction

Due to advancement in technology, current OCR technology is improved but still there is large number of poor quality old documents like books, newspapers for which OCR does not give very good results. To improve the accuracy of OCR output, post-processing must be done. The main aim of post-processing is to remove errors which are done by OCR. The post-processing can be done in two ways.1) Manually 2) Automatically. As it takes so much time for manual post-processing so this paper focuses on new method for post-processing of Hindi OCR. Mainly the OCR misrecognizes the characters which are similar in shape (like in Devanagari script द with ढ). To correct this misrecognition is challenging one. The main aim of this research work is to develop a system that automatically corrects misrecognized words.

The rest paper has been organized as follows: Part 2 discusses previous work in the field of post-processing of OCR; Part 3 explains the proposed research work, Part 4 shows experimental results and discussions, Part 5 gives conclusion about research work and Part 6 contains References.

2. Previous Work

In past various post-processors were used to enhance the accuracy. Different post-processing techniques were proposed in past decades. In[1], developed the new program called *spell* that took the incorrect word and generate a list of candidate words along with probability ranking. The performance of system was increased up to 89%. In[2], done the survey on various kinds of errors that is caused by OCR and explained various kinds of techniques to remove these errors. In [3], developed a program based on a Noisy channel. The program generated a list of candidate corrections and sort them according to probability ranking. In[4], minimum edit distance technique along with substitution cost as confusion probabilities of characters was proposed. The accuracy of OCR was improved up to 33%. In[5], proposed the statistical OCR model. The accuracy of 90% to 97.34% has been found.

A shape based postprocessor for Gurumukhi OCR has been developed[6]. In[7], Shape encoding based post-processing for Punjabi OCR has been developed. The accuracy up to 4-7% was increased. In[8], developed algorithm which was based on morphological parsing

* *Author for correspondence*

and claimed the accuracy up to 84.27%. In[9], Lexicon free method for error correction using FSM (finite state machines) was proposed. The improvement of about 78% error reduction has claimed. In[10], developed Google's spelling suggestion scheme that was based on the probabilistic *n*-gram model. The success rate was increased to 3-4% by applying these algorithms. In[11], post-processing scheme which use statistical language models at the sub-character level was introduced. The accuracy was 92.67% for Malayalam text. In[12], new method in which building an OCR system for Telugu language script; mainly focussing on the character recognition module was discussed. In[13], ligature based segmentation OCR system for Urdu Nastaliq script was discussed. In [14], surveyed various techniques in correcting OCR errors and determines which techniques are better.

3. Proposed Solution

The proposed method firstly generates a confusion matrix from a large corpus text of Hindi Text.

3.1 Generation of Confusion Matrix

Confusion matrix is a two dimensional matrix that shows how many times one character is confused with other character in OCR output with given Input[15]. The confusion between characters is due to similar shape characters. In order to improve the accuracy of OCR, Confusion matrix is generated from large corpus of data of Devanagari script. The first row in matrix give actual characters in OCR input and first column shows the misrecognized characters. Each cell represents particular original character confused with column character as shown in Figure 1.

	ँ	ं	ः	अ	आ
ँ	39	185	2	0	1
ं	28	1516	1	0	2
ः	0	0	4	0	0
अ	0	0	0	955	4
आ	1	0	0	67	697

Figure 1. Confusion matrix for Devanagari OCR.

3.2 Mathematical Calculation for Probability

This step calculates the probability of one character confused with other character. Mathematically

Prob(b/a) = num(sub(a,b))/num(a)

Where

Prob(b/a) is probability of b with respect to a

num (sub(a,b)) is number of times a is substituted by b

num(a) is the total numbers of a's.

For example suppose अ is replaced by आ.

	ँ	ं	ः	अ
आ	0.0008	0.00397	0.00054	0.00193

Figure 2. Confusion Probabilities for Devanagari OCR.

So from Figure 2

pr(आ /अ) = num(sub(अ /आ)/num(अ)

= 67/34564 (calculated) =0.00193

3.3 Generation of Top Five Confused Characters List for Each Character Along With Probability Scoring

Since in confusion matrix, there are many entries which have very low probabilities as shown in Figure 2, so we have to ignore those probabilities along with characters. Table 1 shows the top five confusion probabilities of प.

Table 1. Confusion Probability of प

Characters	Confusion Probability
प	0.8756497657464784
ष	0.0978967583577568
य	0.0765874668547969
ऽ	.00347889876543676
म	0.0024534765498788

3.4 Generation of Character Trigram Dictionary

To validate the word, we search a word in dictionary and if search is successful the word is present in dictionary otherwise not present. But all inflectional forms are not available in dictionaries. For e.g. का, कि, की. For this purpose trigram dictionary is generated.

(i) For each unique word in the dictionary character

trigrams are generated. for e.g.: Trigram for बचपन is shown in Figure 3.

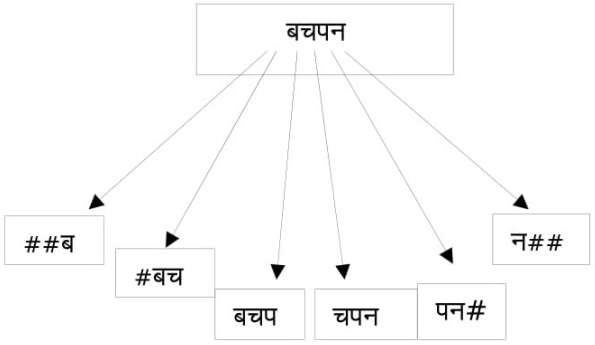

Figure 3. Character trigrams for बचपन.

These trigrams are generated for around 250000 unique words

(ii) Each trigram in stored Trie data structure for fast access and to remove duplicity.

(iii) All the trigrams are stored in file in sorted form

For around 250000 words, 26785 character trigrams are generated.

3.5 Suggestion generation for Devanagari OCR

Firstly, the character trigram dictionary (Figure 2) is stored in Trie. The algorithm is developed for generation of suggestions as follows:

- Let W be the OCR output word.
- Calculate the number of characters in word.
- For each character of word W, (a) Create list containing the top five confusing characters along with probabilities as shown in Table 1.
- Generate string by concatenating each character of one list with other lists of OCR output word W along with probabilities.
- Validation of word is done in this step. Trigrams are generated for each word and search all the trigrams in Trie. If all trigrams are present in Trie then word is validated.
- All the valid words are arranged according to probabilities and top three valid words suggested.

The flowchart for above system is shown in Figure 4.

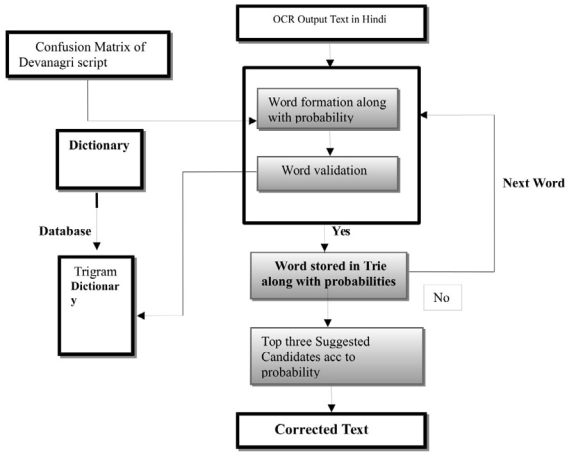

Figure 4. Flowchart of Developed System.

4. Results and Discussion

The performance of system is tested for large number of OCR output in Devanagari Script. Following are the initials taken

- Number of distinct words in dictionary = 250000
- No of trigrams generated = 26785

The system performs with 100 percent accuracy for correct words. First word is top suggestion. Second word is second top suggested word. For Example जगमग, Table 2 shows results. Figure. 5 shows the suggestion for already correct words.

Table 2. जगमग Top three suggestions

Rank.	Word	Probability
1	जगमग	0.99785787574643575
2	ज़गमग	0.056123987975675466
3	ज़गमग़	0.000473679964896436

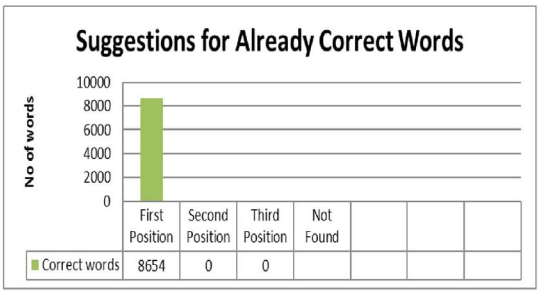

Figure 5. Suggestions for correct words.

Incorrect words are corrected by the above system for the 100 documents. The bar chart in Figure. 6 is same as mentioned in previous chart. Example दमाटर is incorrect word. The word suggestions for this shown in Table 3.

Table 3. दमाटर suggestions along with probabilities

Rank.	Word	Probability
1	टमाटर	0.97854772964746743
2	टमादर	0.04375464785489643
3	टगाटर	0.00076757556464646

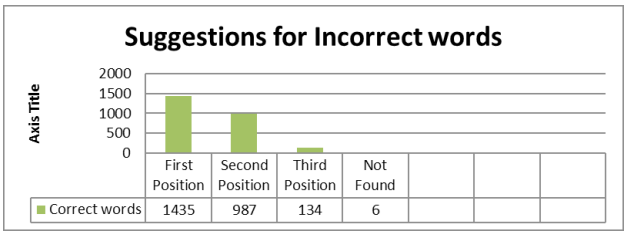

Figure 6. Suggestion for incorrect words.

5. Conclusions

This system used to confusion matrix, dictionary for Hindi text. It corrects non words as well as words which have some meanings. The accuracy rate of Devanagari OCR results has been improved up to 97%.Although it has been shown that the Post-processing can be done efficiently with promising success, the method outlined in this paper should in the future be evaluated in terms of its statistical reliability. Also the method can be used with other Indian as well as foreign languages.

6. References

1. Kenneth WC, William AG. Probability scoring for spelling correction. Statistics and Computing. Dec 1991; 1(2):93-103.
2. Karen K. Techniques for automatically correcting words in text. ACM Computing Surveys(CSUR). Dec 1992;24(4):377-439.
3. Mark DK, Kenneth WC, William AG. A Spelling correction program based on a noisy channel model. Proceedings of 13thconference on Computational linguistics.; USA 1990;p.205-10.
4. Rupy J, Santanu C. Probabilistic approach for correction of optically-character-recognized strings using suffix tree. Proceedings of the 3rd National Conference on Computer Vision. Pattern Recognition, Image Processing and Graphics; India 2011:p.74-7.
5. Masaaki N. Japanese OCR error correction using character shape similarity and statistical language model. Proceedings of the 17th international conference on Computational linguistics; USA 1998:p.922 – 8.
6. Gurpreet SL, Chandan S, Ritu L. A shape based post processor for gurumukhi OCR. Proceedings of the Sixth International Conference on Document Analysis and Recognition.; USA 2001:p.1105-9.
7. Dharam VS, Gurpreet SL, Sarita M. Shape encoded post processing of gurumukhi OCR. Proceedings of 10th International Conference on Document Analysis and Recognition.; USA 2009:p.788-92.
8. Umapada P, Pulak KK, Bidyut BC. OCR error correction of an inflectional Indian language using morphological parsing. Journal of Information Science and Engineering. Nov 2000; 16:.903-22.
9. Okan K, Philip R. OCR post-processing for low density languages. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing;. USA 2005:p.867-74.
10. Youssef B, Mohammad A. OCR post-processing error correction algorithm using Google's online spelling suggestion. Journal of Emerging Trends in Computing and Information Sciences. Jan 2012; 3(1):90-99.
11. Karthika M C VJ. A post-processing scheme for Malayalam using statistical sub-character language models. Proceedings of the 9th IAPR International Workshop on Document Analysis System, USA., 2010:p.493-500.
12. Jawahar C V, Pavin MNSSK, Ravi Kiran SS. A bilingual OCR for Hindi-Telugu documents and its applications. Proceedings of the 7th international conference on document analysis and recognition. USA., 2003:p.408-12.
13. Jyothi J, Manjusha K, Anand Kumar M, Soman P. Innovative feature sets for machine learning based Telugu character recognition. Indian Journal of Science and Technology. Sept 2015; 8(24):1-7.
14. Ankur R, Gurpreet SL. Offline Urdu OCR using ligature based segmentation for Nastaliq Script. Indian Journal of Science and Technology. Dec 2015;8(35):1-9.
15. Atul K.A Survey on various OCR errors. International Journal of Computer Applications.Jun 2016;143(4):8-10.