Urdu Spell Checking: Reverse Edit Distance Approach

Saadat Iqbal, Waqas Anwar, Usama Ijaz Bajwa, Zobia Rehman
COMSATS Institute of Information Technology, Abbottabad, Pakistan
{saadat,waqas,usama,zobiarehman}ciit.net.pkemail@domain

Abstract

Spelling of words of a language are standardized by language authorities or consortiums and available in dictionaries or lexicons. For instance, “produk” does not belong to English dictionary. Similarly “درمیان” is a correctly spelled word, while “درومیانر” is a non-word in Urdu. Electronic representation of text is commonly used in today’s computing environment. The rich resourced languages like English have many applications with added tools. On the other hand, application development is in its infancy for less resourced language like Urdu. Spelling plays a vital role while humans write text electronically in computers. It is oblivious that terabytes of text is added to the internet resource daily is required to be spellchecked. A single book requires multilevel readers for spell checking and correction, if performed manually. Thus automated spell checkers and correctors are required. Spell checking is process of matching a given word with alphabetically ordered words in a dictionary or lexicon. For instance, in Urdu, a given word شاہین is compared with words in Urdu lexicon, if we get a match then the word is considered to be correctly spelled. On the other hand if we compare شاہین with words in lexicon, if we do not get a match then the word is considered to be misspelled. However it is worth mentioning here that the automated spell checking process is reasonably complex in case of Urdu as compare to English. The complex process of spell checking in Urdu is due to its morphologically richness, word space problem, and scarcely available electronic resources.

The spell checking process is a pre-requisite for language processing systems, e.g. Grammar checker, Part of speech (PoS) tagging, Information extraction, Machine translation, etc. The input to above mentioned language process systems must be correctly spelled, and the text must be passed through a dependent spell checker.

1 Introduction

Usage of computers became an essential component in human lives. In today’s computing environment, text processors, search engines, short messaging services, chatting applications, and many more are widely used. Google auto complete feature starts giving options; For instance when “mor” is typed options like: morphine, morphology, morphine drug, and morphological choices appear. On the other hand the user does not know the exact spelling. Google will make a search based on closed matched words and will display the message as well. All this happens in a fraction of second. This reveals that Google and other applications are using spell error detections and correction algorithms which are implemented in applications for user facilitation. Terabytes of text added to the internet resource daily is required to be spellchecked. A single book requires multilevel readers for spell checking and correction, if performed manually. Thus automated spell checkers and correctors are required. Spell checking is process of matching a given word with alphabetically ordered words in a dictionary or lexicon. For instance, in Urdu, a given word شاہین is compared with words in Urdu lexicon, if we get a match then the word is considered to be correctly spelled. On the other hand if we compare شاہین with words in lexicon, if we do not get a match then the word is considered to be misspelled. However it is worth mentioning here that the automated spell checking process is reasonably complex in case of Urdu as compare to English. The complex process of spell checking in Urdu is due to its morphologically richness, word space problem, and scarcely available electronic resources.

The spell checking process is a pre-requisite for language processing systems, e.g. Grammar checker, Part of speech (PoS) tagging, Information extraction, Machine translation, etc. The input to above mentioned language process systems must be correctly spelled, and the text must be passed through a dependent spell checker.

Section 2 narrates about the challenges exists about spell checking in Urdu, Linguist study is narrated in section 3, section 4 discusses proposed technique for Urdu spell checking, the review of review of reverse edit distance algorithms and discussion are presented in section 5, and section 6 is for conclusion.
2 Challenges of Urdu Spell Checking

Tokenization in Urdu, Diction problem, Loan Words, Morphological nature, Grammatical words and Initial letter Capitalization are some of the challenges that makes the Urdu Spell checking complex. Tokenization is process of to break text at word level.

2.1 Word Separator – space character

In English text, words are separated by spaces characters, as tokens (Sara Stymne 2011). To separate words in Urdu, spaces are not used after every word as in case of English language. The natives of Urdu language separate the words from each other by cognitive knowledge of the language while reading or writing text. For instance, in the following example same sentence is written in three variation of space usage:

Space is only used after a word that ends with joiner letter.

تھے۔

Space is not used at all after any of the word that ends with joiner letter.

ہمبازارمنکھلرہےتھے۔

Space is used after each word.

تھے۔

2.2 Morphological Nature

In contrast to languages like English, Urdu language bends toward agglutinative languages due to its complex morphological nature. The languages like Turkish and Finnish are agglutinative languages as multiple words generates from a single word by affixation, derivational, and inflectional suffixes. From the Turkish word “uggar” (means civilized) a word “uggarlastira-mayableceklerimizdenmissinzisine” is derived (Kemal Oflazer and Cemaleddin 1994). “Heat”, “heated”, “heats” are the word forms that are inflected from the root word “heat”. Urdu is morphological rich language and multiple words are inflected from a root word. The following are few inflected words that are inflected from the root word of Urdu بول (speak).

تے بول

تم بولو

تم بولنا

اب بولینے

اب بولنے

2.3 Diction Problem

Diction problem is defined as using choice of words from a set of word which has same meanings. The Urdu language is considered to be computationally complex due to its diction problem as well. Example words in Urdu that are different in spelling but having same meanings are: (تکیہ، تکیا)

2.4 Loan Words

The native speakers of the Urdu language take advantage of loan words from other languages like English. Engine: (انجن، O-Level)

2.5 Initial letter Capitalization

In English proper nouns, start of sentences can be recognized with the words with letter capitalized. For instance,

• The world is shrinking …. [initial letter of “The” is capitalized at the beginning of sentence]

• The delegation will meet Abu Bakar at Islamabad …. [initial letter of proper nouns “Abu Bakar” and “Islamabad” are capitalized.

2.6 Grammatical Words

Grammatical words are prepositions, adverbs, conjunctions etc. they themselves have not a very clear meaning. However, these words are used to complete sentences and their meanings are expressed in dictionaries with the help of examples. For instance, for, with, the, of, etc. In Urdu the examples of grammatical words are، سے،، نے، کا، respectively.

3 Literature Review

3.1 Historic perspective of Spell Checking

Wherever there is text processing, misspelled words come across, and these misspelled words are required to be detected and corrected. Thus research in spell detection and correction started in the period when text processing become common for computer users. In 1964, Fred J Damereau, articles (Fred J. Damereau 1964) explained fundamentals techniques for spelling detection and correction. According to Damereau, spell checking is a process of comparing an input index with a master list of acceptable terms, and rejects those word from the input which has no match in the master list. In tests conducted by Damereau, indicated that 80% of the spelling errors falls in single letter error, these errors are:

Substitution; a letter is wrongly substituted by another letter,
Insertion: a letter is wrongly inserted at some position,
Deletion: a letter is wrongly deleted from some position,
Transposition: two letters are wrongly transposed.

Few examples words that are taken from the test data of Damrau are given in a Table 1.

| Error type | Correct word | Misspelled word |
|------------|--------------|-----------------|
| Substitution | Absorbent    | Absorbant       |
| Insertion  | Commitment   | Commitmentt     |
| Deletion   | Governmentt  | Government      |
| Transposition | Weird       | Weird           |

Table 1: Extracts from Damrau Test data

In a test conducted by Damrau, a data of 964 spell errors was taken for conducting a test. The results are presented in Table 2.

| Correctly identified | Incorrectly identified | Not identified | Total |
|----------------------|-----------------------|----------------|-------|
| Replacement          | 549                   | 18             | 567   |
| Deletion             | 143                   | 10             | 153   |
| Transposition        | 23                    | 0              | 23    |
| Insertion            | 97                    | 2              | 99    |
| Multiple errors      | 0                     | 0              | 0     |
| Total                | 812                   | 30             | 964   |

Table 2: Major error types for spelling errors

### 3.2 Spelling Error classification

Spelling errors are classified into two types, namely typographic errors and cognitive errors (Kyongho Min, William H. Wilson, Yoo-Jin Moon), (Tahira Naseem 2004). Typographic errors are those errors in which the person typing the text knows spelling, however mistype the word. For instance, a user intends to type “listen” but wrongly types “listyen”. The additional adjacent key ‘y’ is pressed while the typist intended to type ‘t’ in the word “listen”. Thus, the “listyen” example pertains to insertion as explained by Damrau. In case of cognitive error, spelling of word is not known to the writer. Due to existence of homophone alphabet set in Urdu language, cognitive errors are found in the Urdu written text. For instance [ض، ڑ، ڑ]. For Urdu spelling errors trends, a study was conducted by (Tahira Naseem et al). The data for the study was taken from newspaper text, and students term papers. In English, [s, c] are phonetically equivalent, or termed as homophone alphabets; given in example [race, rase]. In case of Urdu text, there are several homophones characters sets. For instance [ض، ڑ، ڑ]. Similarly, visually similar character also exists in Urdu text [ض، ڑ]. Their study exhibited the results illustrated in Table 3.

| Error type   | Total errors | Visually Similar | Phonetically similar |
|--------------|--------------|------------------|---------------------|
| Substitution | 75           | 40               | 12                  |
| Deletion     | 42           | 4                | 5                   |
| Insertion    | 21           | 2                | 1                   |
| Transposition| 12           | 3                | 0                   |
| Total        | 150          | 49               | 18                  |

Table 3: Single Edit Distance Errors in Urdu

### 3.3 Spelling Error correction techniques

The spelling correction solution comprises of three phases:
- Detection of Spelling error
- Finding candidate word(s) for the misspelled words
- Order candidate according to match strength

The following are the techniques that are employed by various spell correction tools in many languages.
- Minimum Edit Distance technique
- Similarity Key technique
- Neural Networks technique

#### 3.3.1 Minimum Edit Distance Technique

In 1965, the Minimum Edit Distance technique was given by Vladimir Levenshtein, to compute minimum edit distance or edit operation required to transform one string $str1$ to another string $str2$. In this technique, a matrix is taken of dimension $m \times n$, where $m$, and $n$ represents the length of two strings $str1$, and $str2$. One of the string say $str1$, is placed at the top row of matrix, and the $str2$ is placed at leftmost column. On execution of Edit Distance algorithm, each cell of the matrix is filled with the difference of edit operations performed.

The Minimum Edit Distance algorithm measures distance between two strings. An insertion operation takes place, when a alphabet is inserted in a non-word sequence to make it correct word. Similarly, deletion operation takes place, when a alphabet is deleted from a non-word sequence to make it correct word. $substitution$ operation takes place when a alphabet is substituted in a non-word sequence to make it correct word. $str2$ is placed at leftmost column. On execution of Edit Distance algorithm, each cell of the matrix is filled with the difference of edit operations performed.

The Minimum Edit Distance algorithm measures distance between two strings. An insertion operation takes place, when a alphabet is inserted in a non-word sequence to make it correct word. Similarly, deletion operation takes place, when a alphabet is deleted from a non-word sequence to make it correct word. $substitution$ operation takes place when a alphabet is substituted in a non-word sequence to make it correct word. If there are $w$ number of words in a lexicon, the minimum edit distance algorithm performs $w$ comparisons of a misspelled word with all $w$ words in dictionary. To minimize the comparisons, re-
verse edit distance algorithm (Eric Brill and Robert C. Moore), (M. D. Kernighan, K. W. Church, and W. A. Gale 1990) is proposed. In modified algorithm only \(53n + 25\) words comparisons are performed, where \(n\) is length of misspelled English word.

### 3.3.2 Similarity Key Technique

This technique is based upon the key assignment to the words of a language. Words are composed of alphabet. In this technique a set of alphabet is taken based on sound similarity. It is to be noted that the key is not unique, and will be explained shortly. The following are the sets in Similarity Key technique.

| Digit of Key | Alphabet |
|--------------|----------|
| 0            | a, e, i, o, u, h, w, y |
| 1            | b, f, p, v |
| 2            | c, g, j, k, q, s, x, z |
| 3            | d, t |
| 4            | L |
| 5            | m, n |
| 6            | R |

Table 4: Similarity keys for English Alphabets

In this technique, the key is calculated for a misspelled word. Then the words from the lexicon that have the same key value are extracted for candidate of a correctly spelled word. The key are generated by keeping the first letter of the word followed by digits mapped from the Table 4. For instance the key t0140 is generated for the word *table*. The key are generated by keeping the first letter of the word followed by digits mapped from the Table 4. For instance the key t0140 is generated for the word *table*. The zero and are eliminated from the key. Similarly repetition of a character is collapsed. Thus in second step the key t0140 becomes t14 for the word *table*.

The Urdu character set is also composed of many homophones. A study has been conducted (Tahira Naseem 2004) on spelling mistakes of Urdu words in context to soundex.

The similar Urdu sounded letter are shown in Table 5, and Table 6 of soundex scheme 0-F, and 0-9 respectively.

| code | Alphabet |
|------|----------|
| 0    | س، ص، ص، ث |
| 1    | ت، حث |
| 2 ~ 8 | ................................. |
| 9    | 0 ~ 9 |

Table 6: Similar sounded letters – Urdu (Scheme 0 ~ 9)

### 3.3.3 Neural Networks Technique

Neural networks are used in environments where systems are trained on specific error patterns (O. Matan, C. J. C. Burges, Y. LeCun, and J. S. Denker 1992). Thus neural networks can be used for spell correction. The neural network is trained in for spell errors for a specific domain in which the spell correction system will be used. The back propagation method is commonly used in neural network training (V. J. Hodge and J. Austin 2003). The method comprised of three layers; the input, hidden, and output layer. The nodes from inner to output layer are connected through a link through hidden layer. In the training phase, weights are computed and assigned to the nodes from input layer to output layer. The weights represent the relation between the nodes.

### 4 Proposed Technique for Urdu Spell Checking - Reverse Edit Distance

It is obvious that \(q\) linear comparisons are required for a misspelled word with all words of lexicon containing \(q\) words. 70,000 words lexicon, requires 70,000 comparisons. Reverse edit distance technique is proposed (M. D. Kernighan, K. W. Church, and W. A. Gale 1990) in which permutation of edit distance one are generated and compared with lexicon words \(q\). The break-up is as under:

- Insertions of \(\text{ب، ا، ت،ث، ش، ص، } \) at \(n+1\) positions in a word. \(42(n+1)\)
- Deletion of one alphabet in turn from word. \(n\)
- Substitution of \(\text{ب، ا، ت،ث، ش، ص، } \) in turn at each position in word \(42n\)
- Transpositions of adjacent alphabets in word \(n-1\)

| code | Alphabet |
|------|----------|
| 0    | س، ص، ص، ث |
| 1    | ت، حث |

Table 5: Similar sounded letters – Urdu (Scheme 0 ~ F)

The reverse edit distance technique can be employed for Urdu Spell system. The usage of this technique is selected due to its efficient approach.

Urdu misspelled word \(w = \text{'نھک'}\)

Substitutes:
The first letter 'ن' of Urdu misspelled word نھک is substituted with 
ے ی،  ..…  ب، ا، in turn resulting:

اھک  
بھک  
پھک  
تھک  
۔۔۔۔۔۔۔۔۔۔۔۔۔  
ےھک

Likewise, The second letter 'ھ' of misspelled word نھک is substituted with alphabet 
ے ی، ..…  ا، ب، in turn resulting:

نمک ۔۔۔۔۔۔۔۔۔، 
نےک ۔۔۔۔۔۔۔۔۔، 
ناک  
نبک، 
نپک، 
نتک، 

Likewise, The third, and fourth letter get substituted.

Inserts :

ی، ب، ا، ب،  ...... at position 1,2,3, 
and 4 of the misspelled word نھک.

Transposes

 نک، نھک

Deletes

نھک ، نک

4.1 Dictionary lookup

Finite State Automata (FSA) are based on alphabets U of a language L. A string S, comprises of alphabet 
from U. If S belongs to U, then there will be a path 
from the initial state to the final state of the Finite 
state automata, else will declare misspelled word.

If a correct word that is not available in the lexicon would be required to be added while the spell checking 
application highlight as a misspelled word.

4.2 Edit Distance (Two operations)

Despite the fact that misspelled words are corrected by one edit operation. Thus it may happen 
that out of 86n + 41 permutations generated from 
an Urdu non-word may lead to zero match in the 
lexicon. Thus in our proposed work, edit distance is procedure is called again on all the permutation 
generated by the edit distance one operation. This drastically slows down the system as per 
mutations (2nd cycle ) for each of the permutation 
generated at edit distance 1 will be generated.

4.3 Candidate Generation

Candidates are selected upon the existence of any match of the permuted word in lexicon. In our example case two candidates are generated:

4.4 Reverse Edit Distance Efficiency

The Reverse Edit Distance algorithm is grossly better than the conventional Edit Distance algo-
spelled word based on the frequency of candidates word exists in the underlying corpus. The methodology is explained in the illustration:

| Misspelled word | Corpus | Candidate words | Top ranked word | Count |
|-----------------|--------|-----------------|-----------------|-------|
| ادر | Urdu corpus | ادا (34), اتر (23), اثر (2), اثر (8) | صدر | get selected |

Table 7. Illustration explaining methodology

In the above illustration, for the misspelled Urdu word ادر we have taken few candidate words ادا، اتر، اثر shown in the third column of Table 7 which are at edit distance one, that is one edit operation we can get these words from the misspelled word. Now to decide which of these candidate words is intended correct word that typist wanted to write cannot be flatly decided.

Figure 1. Flow chart depicting the process of spell checking and correction

5 Results and Discussions

5.1 Training Corpus

Two corpuses of 54,440 words, and 56,142 words [Center For Research In Urdu Language Processing] are taken that has been used as training lexicon. In subsequent work later, a comprehensive corpus will be taken for results. The training data is used for frequency count of the words. The frequency count is used for analysis of candidate words from a misspelled word. Let’s take a simple corpus of 20 Urdu words to understand the training set concept:

| لفظ (Words) | تعداد (Frequency) |
|--------------|-------------------|
| ابتدائی | 1 |
| نتائج | 2 |
| ایک | 1 |
| سال | 3 |
| سو | 1 |
| سے | 2 |
| پچھلے | 1 |
| نازمین | 1 |
| اس | 1 |

Table 8. Frequency – Mini training set data

Now taking misspelled word سق، let the spell correction system generate two candidates; the word سو [frequency count = 1], and the word سے [frequency count = 2]. Based on frequency, suggested word is calculated which comes to be سے for the misspelled word سق.

Taking another example of few misspelled words and executing our algorithm on a training set corpus of 54,400 words. We would see abstract behavior of algorithm and make a little discussion on the result generated. The following non-words are provided to the system for generation of candidate words, there after highlighting one word.

5.2 Urdu Non-words example

Test word- i

| استعمال | کلمے جاتے تی معین جنس کئی جنگل ایشک
|-----------|-------------------|
| Candidate word(s) | استعمال |
| Suggested word | استعمال |

Test word- ii

کلمی
In the above example, we have taken 09 non-words of Urdu and pass through the reverse edit distance algorithm. Candidate words in the range of one to three words are generated for each of the non-word. These are words at edit distance of one or two from the corresponding non-word. The candidate words are generated by insertions, deletion, substitution, or transposition operations.

5.3 Correct word suggestion

In the corpus the \( c(جاتی) = 22 \), and \( c(ساتھی) = 5 \).

Thus \( c(جاتی) > c(ساتھی) \) thus the word \( جاتی \) is suggested as the top ranked word for the misspelled word \( w = جاتپی \).

**Insertion**

The following are selected non-words examples specifically for insertion of alphabet that are passed through the proposed system to get candidate results:

Test word

Candidate word(s)

Suggested word

**Deletion**

The following are selected non-words examples specifically for deletion of alphabet that are passed through the proposed system to get candidate results:

Test word

Candidate word(s)

Suggested word

**Substitution**

The following are selected non-words examples specifically for substitution of alphabet that are passed through the proposed system to get candidate results:

Test word

Candidate word(s)

Suggested word

**Transposition**

The following are selected non-words examples specifically for transposition of two adjacent
alphabets that are passed through the proposed system to get candidate results:

Test word بنلد
Candidate word(s) بنلد بلند بنلد بلند
Suggested word بنلد

We have noticed that the transposition errors are poorly corrected using reverse edit distance as compared to edit distance algorithm. In above results, the words قیلہ پچھے تبیدل are not properly corrected by the algorithm.

6 Conclusion

Urdu language has rich literature, spoken in south asia, however having scarce resources in context of computer based applications. In this work, focus is on spelling error detection and correction feature in these electronic applications. Our this work is concentrated on gathering various spell checking and correction techniques that are suitable for correcting Urdu spelling errors. Reverse Edit Distance algorithm complexity is computed to be 86n + 41. The algorithm has been implemented in other languages like English, and has to be implemented for Urdu.

References

Sara Stymne, Spell Checking Techniques for Replacement of Unknown Words and Data Cleaning for Haitian Creole SMS Translation, Association for Computational Linguistics, Proceedings of the 6th Workshop on Statistical Machine Translation, pages 470–477, Edinburgh, Scotland, UK, July 30–31, 2011.

Kemal Oflazer and Cemaleddin, Spelling correction in agglutinative languages. In Proceedings of the fourth conference on Applied natural language processing (ANLC ’94). Association for Computational Linguistics, Stroudsburg, PA, USA, 1994.

Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. Commun. ACM 7, 3 (March 1964), 171-176. DOI=10.1145/363958.363994.

Tahira Naseem A Hybrid Approach for Urdu Spell Checking, MS Thesis 2004.

David Gries et al. Presenting an algorithm to find the minimum edit distance Department of Computer Science, Cornell University NSF Project 1988.

Eric Brill and Robert C. Moore, Microsoft Research, An Improved Error Model for Noisy Channel Spelling Correction YR + JNL.

M. D. Kernighan, K. W. Church, and W. A. Gale. A spelling correction program based on a noisy channel model. In Proceedings of the 13th International Conference on Computational Linguistics, volume 2, 1990.

O. Matan, C. J. C. Burges, Y. LeCun, and J. S. Denker. Multi-digit recognition using a space displacement neural network, Neural Information Processing Systems, volume 4, pages 488-495. Morgan Kaufmann Publishers, San Mateo, C.A., 1992.

V. J. Hodge and J. Austin. A comparison of standard spell checking algorithms and a novel binary neural approach. IEEE Transactions on Knowledge and Data Engineering, 2003.

Sarmad Hussain, Resources for Urdu Language Processing, Center for Research in Urdu Language Processing