Design and Implementation of Punjabi Spell Checker

Gurpreet Singh Lehal
Department of Computer Science, Punjabi University, Patiala, India.
gslehal@yahoo.com

Abstract :

Spellcheckers are the basic tools needed for word processing and document preparation. Designing a spell checker for Indian languages such as Punjabi poses many new challenges not found in English, which complicates the design of the spell checker. Punjabi language is far different from Western languages in phonetic properties and grammatical rules. Thus the existing algorithms and techniques that are being used to check the spelling and to generate efficient suggestions for mis-spelt words of English and other Western languages are not actually suitable for Punjabi; rather it needs different algorithms and techniques for expected efficiency. This paper presents the complete design and implementation of a Punjabi spell checker.

Keywords :

Punjabi, Spellchecker, Suggestion List, Typing errors

Introduction :

Spellcheckers are the basic tools needed for word processing and document preparation. A spell checker is a tool that enables us to check the spellings of the words in a text file, validates them i.e. checks whether they are right or wrongly spelled and in case the spell checker has doubts about the spelling of the word, suggests possible alternatives.

The main steps performed by the spell checker are:

1. Take the word from the file as its input.
2. Pre-process the word
3. Look for the word in dictionary
4. In case, the word exists, pass onto the next one.
5. If the word is not found, then seek for the closest matching patterns and put them up in the form of suggestions.

Even though this appears to be very simple at first glance but designing a spell checker for Indian languages such as Punjabi poses many new challenges not found in English, which complicates the design of the spell checker. Punjabi language is far different from Western languages in phonetic properties and grammatical rules. Thus the existing algorithms and techniques that are being used to check the spelling and to generate efficient suggestions for mis-spelt words of English and other Western languages are not actually suitable for Punjabi; rather it needs different algorithms and techniques for expected efficiency. Some of the typical problems faced during designing the Punjabi are:

1. There is no standardization of Punjabi keyboard layouts. There are more than forty keyboard layouts and fonts commonly being used, which means that the same Punjabi word can be internally stored in forty different ways. As for example, the word ਪੰਜਾਬੀ is internally stored as
   - pMjwbI in *Akhar* font
   - pMj`bI in *Amrit-Lipi2* font
   - pµj;bI in *Anandpur Sahib* font
   - gzikph in *Asees* font
   - ê³ÜÅìÆ in *Satluj* font

   The spell checker has to deal with each of these cases separately and check for the spellings. Even in the same font, a character can be typed and stored in more than one way. As for example, in *Asees* font, the character ਸ਼ can be typed by pressing either a single key *P* or a combination of two keys *;a*. Similarly the character addak (ੱ) can be typed by pressing key *Z* or key ~.
2. Another problem, typical to Indian language scripts was faced. Since Punjabi is not written in linear fashion, so same word could be internally stored in more than one way. The user can type consecutive occurring semi-vowels/upper vowels and half character/lower matra (਼ or ੍) in any order and they all look visually similar. For example, the word ਖੁਲ੍ਹੇ can be typed by the typist as ਖ + ਼ + ਲ + ੁ + ੇ or as ਖ + ਼ + ਲ + ੇ + ੁ and both will be displayed as ਖੁਲ੍ਹੇ. Thus if ਖੁਲ੍ਹੇ has been typed and stored in the database as ਖ + ਼ + ਲ + ੁ + ੇ if the user types ਖੁਲ੍ਹੇ as ਖ + ਼ + ਲ + ੇ + ੁ then the word will be signalled as wrongly spelled.
3. In some of the Punjabi fonts, the Punjabi characters such as *bindi, lava, onkar, dulainkar* etc. have zero width and so if by mistake a user makes multiple entries of such characters only a single entry is visible. If the spell checker flags such word as misspelled the user will not come to know where the error. Thus for example, consider the word ਪ੍ਰੈੱਸ it has been typed as ਪ ੍ ◌ੌ ਤ ੍ ◌ੱ and visually the word looks correct but internally it has stored wrongly and the user will not be aware where the error lies.
4. Unlike English, there is no well defined word boundary for Punjabi words written in different Punjabi fonts. As for example, in *Asees* font the following punctuation marks are encoded as Punjabi characters and thus are part of the word (‘ “ + / : ; ? [ ] \ { }). But there are many other fonts such as *Akhar, Satluj* etc. which do not encode the above punctuation marks as Punjabi characters. So the extraction of word boundary is font dependent in case of Punjabi.
5. There is no standardization of Punjabi spellings. A word may be spelled in more than one way and all the forms may be acceptable.

Brief Description of Punjabi Language :

Punjabi is the world's 12th most widely spoken language. The populace speaking Punjabi is not only confined to North Indian states of India such as Punjab, Haryana and Delhi but is spread over all parts of the world. Punjabi is a phonetic language and commonly written in Gurmukhi script. Some of the major properties of the Gurmukhi alphabet are:

- Gurmukhi script alphabet consists of three (ੳ ਅ ੲ) vowel-carrier letters and nine vowel signs. By using the vowel signs with the three vowel-carrier letters ten vowels are obtained. The vowel-carriers ੳ and ੲ are never used without a vowel sign.
- There are 38 consonants and all the ten vowel-signs are used with all the other consonants (Fig. 1).
- In addition there are two nasal signs ◌ਂ (*bindi*) and ◌ੰ (*tippi*) used for sounds produced through nasal cavity. The symbol *adhak* ◌ੱ , is used to produce the sound of a double consonant.
- There are three half characters in Gurmukhi alphabet. The complete character set of Gurmukhi is depicted in Fig. 1.

Vowels
ਅ ਆ ਇ ਈ ਉ ਊ ਏ ਐ ਓ ਔ

Vowel carriers
ੳ ਅ ੲ

Consonants
ਸ ਹ ਕ ਖ ਗ ਘ ਙ
ਚ ਛ ਜ ਝ ਞ ਟ ਠ ਡ ਢ ਣ
ਤ ਥ ਦ ਧ ਨ ਪ ਫ ਬ ਭ ਮ
ਯ ਰ ਲ ਵ ੜ ਸ਼ ਜ਼ ਖ਼ ਫ਼ ਗ਼ ਲ਼

Matras
◌ਾ ◌ਿ ◌ੀ ◌ੁ ◌ੂ ◌ੇ ◌ੈ ◌ੋ ◌ੌ

Vowel Modifiers or Half Vowels
◌ਁ ◌ਂ ◌ੱ

Half Characters
◌ੑ ◌ੵ ◌ੴ

Fig 1 : Gurmukhi character set

Lexicon Creation :

The first step in development of the spell checker is the creation of a lexicon of correctly spelled words, which will be used by the spell checker to check the spellings as well as generate the suggestions. Two issues are involved in lexicon development:

• **Size of the lexicon** : There are two approaches followed for storing the lexicon. The first approach stores the root words of a language and the rest of the words are derived from these root words. The other approach is to store all the possible words of the language in the lexicon. We have followed the second approach and stored all the possible forms of words of Punjabi lexicon. Around 1.5 lakh words were identified and stored in the database. The words are arranged according to the word size. During program execution the words are loaded into AVL trees and sixteen different AVL trees used for different word lengths. The number of nodes in each of these AVL tree is shown in Fig. 2.

• **Format of the words** : As there is no standardized Punjabi keyboard, a word in Punjabi may be written in more than forty different ways. It was necessary to formalize a format for storing the lexicon. One option was to store a word in ISCII (Indian Standard Code for Information Interchange) or Unicode. But that option was dropped, since the coding schemes are not closer to how a user actually types. As for example, if a user types ਪ੍ਰਿਸ then it will be stored in ISCII as (ਪ + chr(232) + ਰ + ◌ਿ + ◌੍ + ਸ), while the user will be typing it as (◌ਿ + ਪ + ◌੍ + ◌ਰ + ਸ). It is necessary to store the words in same order as they are typed as this knowledge is helpful in generating the suggestion list. It was also necessary to assign coding beyond ASCII 127 to the characters, since in most the databases the searching algorithms do not differentiate between upper and lower case characters. Thus two characters assigned codes corresponding to ASCII 'a' and 'A" will be treated to be same for sorting and searching purposes. Also care was taken that if a half character or a lower matra (◌ੑ or ◌ੵ) and a semi-vowel occur together , then semi-vowel will be stored after the half character or lower matra. The coding scheme displayed in Table 1 is used to codify the Punjabi letters.

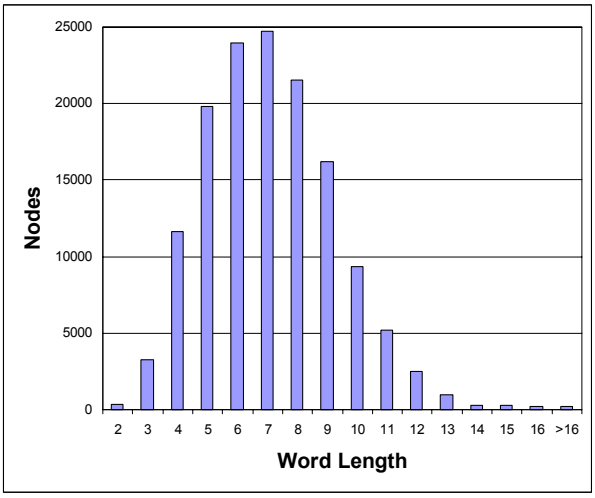


Fig. 2 : Node count in AVL trees of different word lengths

Table 1 : Gurmukhi coding scheme

Code	Punjabi Symbol	Code	Punjabi Symbol
128	◌ਁ	149	ਝ
129	◌ੱ	150	ਵ
130	◌ਂ	151	ਟ
131	ੳ	152	ਠ
132	ਓ	153	ਡ
133	ਅ	154	ਢ
134	ੲ	155	ਣ
135	ਸ	156	ਤ
136	ਸ਼	157	ਥ
137	ਹ	158	ਦ
138	ਕ	159	ਧ
139	ਖ	160	ਨ
140	ਖ਼	161	ਪ
141	ਗ	162	ਫ
142	ਗ਼	163	ਫ਼
143	ਘ	164	ਬ
144	ਙ	165	ਭ
145	ਚ	166	ਮ
146	ਛ	167	ਯ
147	ਜ	168	ਰ
148	ਜ਼	169	ਲ
170	ਲ਼	178	◌ੀ

171	ਢ
172	ੜ
173	ੰ
174	ੱ
175	ੲ
176	ੳ
177	ਿ

179	੍
180	ੂ
181	ੇ
182	ੈ
183	ੋ
184	ੌ

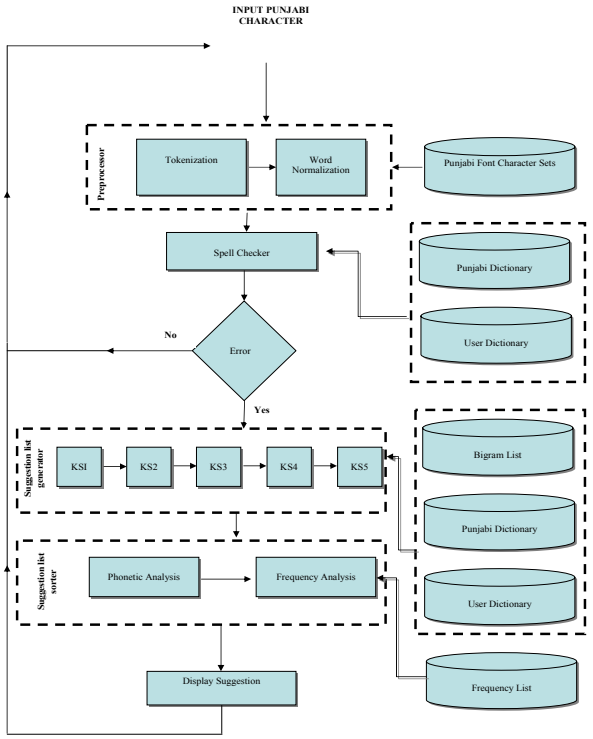
Spell Checker Architecture :

The major components of the architecture as shown in Fig. 3 are: Tokenisation and normalisation Pre-processing Module, Lexicon Lookup/Error Detection Module and Suggestion Module.

Tokenisation :

Tokenisation refers to process of breaking the text into tokens or words using punctuation marks and spaces as delimiters. In case of Punjabi, the punctuation delimiters vary from font to font. As for example, in font *Asees* the punctuation mark ; and *:* are used to store a Punjabi characters. Similarly, the locations corresponding to the characters ` and ~ are used by majority of the Punjabi fonts. So we have to associate with each font, the valid character set. The spellchecker reads the text character by character along with its font information and uses the information about the character set of that font to tokenise the text. The tokenisation process will check each character in the character set corresponding to that particular font. As for example, the following text ਜੋ ਦਿੱਲੀ; ਵਿੱਚ ਰਹੰਦੇ ਹਨ
is encoded internally as follows in *Asees* font:
i' fd~bhl ft~u ojzd/ jB
The tokeniser will break the line into following tokens:
i' fd~bh ft~u ojzd/ jB

We note that in the second token above, the character l has been excluded since it is not part of the character set for *Asees* font while the characters ' ~ and / have been included in the tokens, since they are part of the character set.



Normalisation :

The tokens are then made to pass through a normalisation process to convert them to the format in which the lexicon has been stored. Some of the major steps that are performed in this stage are : removal of redundant zero width characters, changing the order of upper and lower characters and mapping the token to format compatible with the lexicon. Thus for example, the word ਹੁੰਦੇ in typed in any Punjabi font will be normalised as {137, 179, 130, 158, 181}.

Lexicon Lookup/Error Detection :

In this module the normalised token is searched in the standard dictionary. The standard dictionary has been partitioned into sixteen sub-dictionaries based on the word length and at execution time each of these sub-dictionary is loaded in a height balanced binary search tree (AVL tree). To search for a word of length n, we look for its presence in the AVL tree storing words of length n. Thus ਹੁੰਦੇ will be searched in the AVL tree storing lexicon words of length 5. If the word is not found, then it is searched in the AVL tree storing the lexicon of user defined dictionary. If the word is still not found, then it is marked and sent to Suggestion module.

Suggestion :

In this module a list of possible correct words is presented to the user. The user selects the correct word, if it is present in the suggestion list, and can give the command to replace a single or all occurrences of the mis-spelled word with the word selected in the suggestion list. The details of this module are discussed in next section.

Suggestion List Generation :

Once the system has detected an erroneous word, it performs the following steps:

1. Generate a list of candidate corrections
2. Rank the spelling variations
3. Select the highest ranking as the most likely correction

The task of general purpose spelling correction has a long history (e.g. Damerau, 1964; Rieseman and Hanson, 1974; McIlroy, 1982), traditionally focusing on resolving typographical errors such as insertions, deletions, substitutions, and transpositions of letters that result in *unknown words* (i.e. words not found in a trusted lexicon of the language). Typical word processing spell checkers compute for each unknown word a small set of in-lexicon alternatives to be proposed as possible corrections, relying on information about in-lexicon-word frequencies and about the most common keyboard mistakes (such as typing *m* instead of *n*) and phonetic/cognitive mistakes, both at word level (e.g. the use of *acceptible* instead of *acceptable*) and at character level (e.g. the misuse of *f* instead of *ph*).

Several approaches based on minimum edit distance, similarity key, rules, N-grams, probability and neural nets are proposed to accomplish the task [1-5]. Of these, minimum edit distance based approaches are the most popular ones. The minimum edit distance is the minimum number of editing operations (insertions, deletions and substitutions) required to transform one text string into another. The distance is also

referred to as Damerau-Levenshtein distance after the pioneers who proposed it for text error correction[6-7]. In its original form, minimum edit distance algorithms require m comparisons between misspelled string and the dictionary of m words. After comparison, the words with minimum edit distance are chosen as correct alternatives. To improve the speed, a reverse minimum edit distance is used where a candidate set of words is produced by first generating every possible single-error permutation of the misspelled string and then checking the dictionary if any make up valid word.

In our present approach, we have used the reverse minimum edit distance to generate the primary suggestion list. Additional words which are phonetically similar to the misspelled words are also added to the suggestion list. The main time in reverse minimum edit distance approach is spent on checking the dictionary for valid word. To decrease the number of words for dictionary lookup, we filter them based on a bigram table. Thus for example, if in a word we have a vowel followed by another vowel, then there is no need to check for that word in the dictionary, since two vowels do not form a valid bigram just as in English *qq* is not a valid bigram. As another example, for the misspelled word ਕੀਰ the suggestion list generator will try to remove the deletion error by inserting a new character at each location. If say it is trying to insert an extra character at second place, then the following generated words would be rejected by the bigram analyser and the time for searching these words in the dictionaries can be saved.

ਕਾਰ ਕਗਿਰ ਕੁਰ ਕੂਰ ਕੋਰ ਕੈਰ ਕੌਰ ਕੌਰ ਕੀਰ ਕੰਰ ਕੱਰ ਕਓਰ ਕਓਰ ਕਅਰ

For suggestion list generation, the results of error analysis for Punjabi carried out earlier [8]were used. In that study about 20000 misspelled words generated by typists, both novice and experienced as well as students learning Punjabi typing were analysed and it was found that the following seven types of Punjabi typing errors occur:

i. **Insertion error (IE):** When at least one extra character is inserted in the desired word. ਪਰਿਵਾਰ -> ਪਰਿਵਦਾਰ

ii. **Deletion error (DE):** When at least one character is deleted in the desired word. . ਪਰਿਵਾਰ -> ਪਰਿਵਰ

iii. **Substitution error (SE):** When at least one character is substituted by the other character. ਪਰਿਵਾਰ -> ਪਰਿਵਨਰ

iv. **Transposition error (TE):** When two adjacent characters are transposed. ਪਰਿਵਾਰ -> ਪਰਿਵਰਾ

v. **Run-on error (ROE):** When there is space missing between two or more valid words**.** ਪਰਿਵਾਰ ਰਸਭਰੀ -> ਪਰਿਵਾਰਰਸਭਰੀ

vi. **Split Word error (SWE):** This is Opposite of Run-on error when some extra space is inserted between parts of a word. The error can be removed by removing the extra space. ਪਰਿਵਾਰ -> ਪਿ ਰਵਾਰ

vii. **Visual error (VE) :** When a word looks visually correct but internally it is invalid. As for example, the word ਉਹ visually looks to be rightly spelled but internally it is stored as ਉ ◌ੁ ◌੍ ਹ . The extra vowel symbol ◌ੁ is hidden under the character ਉ.This error is typical to Punjabi typing.

It was observed in the study [8], that the most common typing errors are substitution and deletion errors.

The Cognitive errors are mainly phonetic based, as Punjabi is a phonetic language. Main confusion is related to:

– Usage of *addak* (ਅਖਾਂ ਇਜ਼ਤ ਸਮਸਿਆ ਕਢ ਸ਼ੁਧ)
– Usage of *pairin bindi* characters (ਸ਼, ਗ਼, ਖ਼, ਫ਼, ਜ਼, ਲ਼) (ਆਜਾਦੀ ਸਖਤ)
– Others (ਕੀਤਾਬ ਗਯਾ)

For generating the suggestion list, the following five knowledge sources have been used:

- KS1 (Character Substitutor) : The Character Substitutor corrects mistakes that arise due to wrong typing of a character. It substitutes each character with other valid characters.
- KS2 (Character Inserter) : The Character Inserter corrects erroneous words that have one character missing.
- KS3 (Character Remover) : The Character Remover corrects an erroneous word by removing a single character to produce a correct word.
- KS4 (Subsequent Character Switcher) : The Subsequent Character Switcher swaps two consecutive characters in an erroneous word to generate a correct word.
- KS5 (Phonetic Similar Tester) : The Phonetic Similar Tester tests if the suggested word is phonetically similar to the erroneous word.

Sorting the Suggestion List :

After gathering the suggestions, the sorting procedure is executed to sort out the suggestion list efficiently so that user may get the suggestions in the most useful format. For the efficiency of the spell checking process, it is important that the right suggestion is presented as a default suggestion. In such a case, the user needs only to confirm the default suggestion and proceed with the next error. Otherwise, the user needs to scroll through a list of suggestions and pick one as the right one. Even worse, often the right suggestion is not on the list and thus the user needs to type the full word again. In order to sort the suggestion list most usefully, we have used three parameters:

1. Phonetic similarity of the suggested word with the related misspelt word
2. Frequency of occurrence of the suggested word
3. The smallest number of substitutions, insertions and deletions required in that order to change the misspelt word to the suggested word.

As an example, for wrongly spelled word ਦਸਿਆ
The Suggestion list generated is

- ਧਸਿਆ ਫਸਿਆ ਕਸਿਆ ਰਸਿਆ ਵਸਿਆ ਘਸਿਆ ਦਇਆ ਦਹਿਆ ਦਰਿਆ ਦਬਿਆ ਦਗਿਆ ਦਲ਼ਿਆ ਦਿਸਿਆ ਦੱਸਿਆ ਦੁਆ ਦੀਆ ਦੂਆ ਦਸਤਾ

After sorting, the suggestion list is reordered as

- ਦੱਸਿਆ ਦਿਸਿਆ ਦਰਿਆ ਦਬਿਆ ਦਗਿਆ ਦਲ਼ਿਆ ਦਹਿਆ ਦਇਆ ਦਸਤਾ ਦੀਆ ਦੂਆ ਦੁਆ ਧਸਿਆ ਫਸਿਆ ਕਸਿਆ ਰਸਿਆ ਵਸਿਆ ਘਸਿਆ

Similarly, for wrongly spelled word ਪੰਜਬ
The suggestion list before sorting is

- ਪੰਜੇ ਪੰਜੀ ਪੰਜਾ ਪੰਜੇਬ ਪੰਜਾਬ ਪੰਜ

The suggestion list after sorting is:

- ਪੰਜਾਬ ਪੰਜ ਪੰਜਾ ਪੰਜੇਬ ਪੰਜੇ ਪੰਜੀ

Evaluating the Suggestion List :

Evaluating the Suggestion List provided by a Spell Checker involves considering three factors:

- Whether the desired word appears on the suggestion list;
- the length of the list of words offered, and;
- the position of the desired word on the list.

The ideal spell checker would offer one word on the list, and that would be correct. If ten words are offered and the correct word is near the bottom, a poor speller must read through the choices and disregard the first few despite their precedence, instead looking for the word which they know, in some way, to be correct. If so many words are offered that they cannot all be seen at once, it is even more difficult to find the correct word.

Test words :

We used 225 most commonly mis-spelled words to analyse the performance of the spell checker. The words were drawn from several sources:

- Punjabi corpus prepared by Ministry of Information Technology
- Online Punjabi Newspapers
- Online Punjabi stories
- Punjabi Research Reports

Table 2 : Spellchecker test words, alphabetically listed

ਉਹਨਾ	ਆਣ	ਸਮੇ	ਕਢਣ	ਦਾਹੜੀ	ਭਣੋਈਏ
ਉਚੀਆਂ	ਆਂਦੀ	ਸਾਹਬ	ਕੱਲੀ	ਦਿਤਾ	ਭੁਖਾ
ਉਚੇ	ਆਵਾਜ	ਸਾਬਿਤ	ਕੱਲੇ	ਦੁਨੀਆਂ	ਮਜਬੂਰ
ਉਤਪੱਤੀ	ਐਵੇਂ	ਸਿਘ	ਕਾਗਜ਼	ਦੇਕੇ	ਮੱਦਦ
ਉਦਾਰਹਣ	ਇਸਤੋਂ	ਸਿਧਾਤਾਂ	ਕਾਨੂਨ	ਦੇਖਕੇ	ਮਧ
ਉਨੱਤੀ	ਇਹਨਾ	ਸਿਰਫ	ਕਾਲਿਜ	ਦੇਂਦੀ	ਮਨੁਖੀ
ਉਨ੍ਹਾ	ਇਹਨੇ	ਸੁਹਪੱਣ	ਕਿੰਨਾਂ	ਦੇਂਦੀਆਂ	ਮਰਜੀ
ਉਨਾਂ	ਇੱਕਠ	ਸੁਟ	ਕਿਵੇ	ਦੇਵੀ	ਮਾਨਣ
ਉਨੀ	ਇੱਕਲੇ	ਸੁਟਿਆ	ਖਿਚ	ਦੋਵੇ	ਮਾਫ
ਉਪਰਲੀ	ਇੰਜਨੀਅਰ	ਸੁਨਣ	ਖਿਲਾਫ	ਨਈਂ	ਮਾਂਬਾਪ
ਉਪਲੱਬਧ	ਇਜ਼ਤ	ਸੁਰੂ	ਖੁੱਲਾ	ਨਜਰ	ਮਾਰਕੇ
ਉਰਫ	ਇੱਨਾ	ਸੂਫੀ	ਗਲਬਾਤ	ਨਫਰਤ	ਮਿਤਰ
ਉਸ	ਇਨਾਂ	ਸੋਲਾਂ	ਗੁਸੇ	ਨਾਹੀਂ	ਮਿਤਰੋ
ਉਹ	ਏਨ੍ਹਾਂ	ਸ਼ਨਾਖਤ	ਗੁਰੂ	ਨਾਵਾਂ	ਮਿਲਕੇ
ਅਖਬਾਰ	ਸਖਤ	ਸ਼ਰੀਫ	ਘਟਨਾਂਵਾ	ਨਿਸਚਾ	ਮੀਤੀ
ਅਖਬਾਰਾਂ	ਸਖਤੀ	ਸ਼ੁਧ	ਚਕਰ	ਨਿਸਚਿਤ	ਮੁਸਕ੍ਰਾ
ਅਖਾਂ	ਸਤੱਹੀ	ਹਸ	ਚਿਠੀ	ਨਿਸਚੇ	ਮੁਸਕ੍ਰਾਉਂਦੀ
ਅੰਗਰੇਜੀ	ਸਬਜੀ	ਹਸਦਿਆਂ	ਚਿੰਨ	ਨਿੱਕਲ	ਮੁਸਕ੍ਰਾਹਟ
ਅਗੋਂ	ਸਬੰਧਤ	ਹਸਦੀ	ਚੀਜਾਂ	ਨਿਕੇ	ਮੁੰਹ
ਅਥਰੂ	ਸੰਬੰਧਤ	ਹਸਦੇ	ਚੁਕਿਆ	ਨੁਸਖਾ	ਮੁਕ
ਅਧਿਅਨ	ਸਬੰਧਾਂ	ਹਕ	ਚੁਕੀਆਂ	ਨੁਸਖੇ	ਮੁਢ
ਅਫਗਾਨਿਸਤਾਨ	ਸੱਭ	ਹਥੋਂ	ਛਡ	ਨੂ	ਮੁਨਾਫਾ
ਅਮ੍ਰਿਤਸਰ	ਸਮਸਿਆ	ਹਦ	ਛਡਣ	ਨੌ	ਮੁਫਤ
ਆਕੇ	ਸਮਾਜਕ	ਹਲ	ਛਾਂਵੇ	ਪਛਮੀ	ਮੁੜਕੇ
ਆਜਾਦੀ	ਸਮੁਚੇ	ਹੁਣ	ਛੁਟ	ਪਰਕਾਸ਼	ਮੈਨੂੰ
ਕੱਠੇ	ਦਸਦੀ	ਬੁਲ੍ਹਾਂ	ਜਹੀ	ਪਰਬੰਧ	ਮੋਤ
ਕਢ	ਦਸਦੇ	ਭਣੋਈਆ	ਜਹੀਆਂ	ਪਰਾਪਤ	ਯੁਧ
			ਜਹੇ	ਪਰੋਗਰਾਮ	ਯੁਨੀਵਰਸਿਟੀ
			ਜਜ	ਪੜ੍ਹਕੇ	ਰਖਕੇ
			ਜੱਥੇ	ਪੜ੍ਹਣ	ਰਖੋ
			ਜਦੋ	ਪ੍ਰਸਿਧ	ਰਫਤਾਰ
			ਜਾਕੇ	ਪ੍ਰੰਪਰਾ	ਰਾਜੀ
			ਜਾਨਣ	ਪ੍ਰੰਪਰਾਵਾਂ	ਲਗਣ
			ਜਿਨਾਂ	ਪ੍ਰੇੜਤਾ	ਲਗਿਆਂ
			ਜਿਵੇ	ਪੁਛਦਾ	ਲਗੀਆਂ
			ਜ਼ਿਲਾ	ਪੁਜ	ਲਭ
			ਜ਼ਿਲੇ	ਪੁਜੇ	ਲਭਦਾ
			ਟੁਟ	ਪੁਤਰ	ਲਾਈਕ
			ਤਰ੍ਹਾ	ਫੁਲਾਂ	ਲਾਕੇ
			ਤੁੰ	ਬਜੇ	ਲਿਜਾਣ
			ਤੁਸੀਂ	ਬਣਕੇ	ਲੈਕੇ
			ਤੋਂ	ਬਣਾਕੇ	ਵਖ
			ਤੈਨੂੰ	ਬਨਣ	ਵਖਰਾ
			ਥਾਵਾਂ	ਬਨਾਉਣ	ਵਖਰੀ
			ਥੋਡੇ	ਬਨਾਣ	ਵਖਰੇ
			ਦਸਣ	ਬਾਹਾਂ	ਵਫਾਦਾਰੀ
			ਦਸਣਾ	ਬੁਲ੍ਹ	ਵਰਨਣ
			ਵਰਨਣਯੋਗ	ਵਿਚੋ	ਵੇਖਕੇ

These words were fed to the spell checker and statistics such as the position of the word in the suggestion list, the maximum, minimum and average size of the suggestion list were generated (Table 3). It was observed that for 4.39% of the words, the correct word was not displayed in the suggestion list, while for 81.14% of wrongly spelled words, the correct word was displayed on the top of the suggestion list. The minimum and maximum size of the suggestion list is 1 and 52 respectively. The average size of the suggestion list is 15, which is on a higher side, though in 93.4% of cases, the correct word is present in top 10 words of the suggestion list.

Conclusion :

This is the first time that a spell checker for Punjabi language has been designed and implemented. The spell checker is part of the commercial Punjabi word processor Akhar. We have only taken care of non-real word errors. Detection and correction of real word errors is a subject of further research.

Table 3 : Average position of the correct word in suggestion list

Position	Percentage of Occurrence
Top	81.14%
Top 3	85.53%
Top 5	89.03%
Top 10	93.42%
In List	95.61%

References :

1. E. Brill and R. Moore, " An improved error model for noisy channel spelling correction," *Proceedings of the ACL 2000*, 2000, 286-293.
2. A. R. Golding, " A Bayesian hybrid method for context- sensitive spelling correction ," *Proceedings of the Workshop on Very Large Corpora*, 1995, 39-53.
3. A.R. Golding and D. Roth, " Applying winnow to context-sensitive spelling correction," *Proceedings of ICML*, 1996, 182-190.
4. K. Kukich, " Techniques for automatically correcting words in a text," *Computing Surveys* 24(4), 1992, 377-439.
5. E.J. Yannakoudakis and D. Fawthrop,"An Intelligent spelling corrector," *Information Processing and Management* 19(12), 1983, 101-108
6. F.J. Damerau, " A technique for computer detection and correction of spelling errors," *Communications of ACM* 7(3), 1964, 171-176.
7. V.I. Levenshtein , "Binary codes capable of correcting deletions, insertions and reversals,". *Sov. Phys. Dokl.*,10, 1966, 707-710.
8. G.S. Lehal and M. Bhagat, " Error Pattern in Punjabi Typed Text," *Proceedings of International Symposium on Machine Translation, NLP and TSS*, 2004, 128-141.