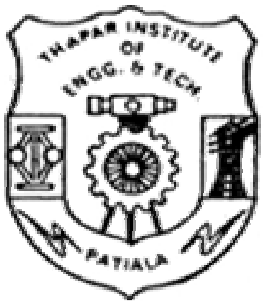
Spelling Error Pattern Analysis of Punjabi Typed Text

A Thesis Report
Submitted in the partial fulfillment of the requirements
for the award of the degree of ME in Software Engineering

Submitted by
Meenu Bhagat
M.E. (Software Engineering)
8013106

Under the supervision of:
Dr. Gurpreet Singh Lehal
Assistant Professor
School of Mathematics & Computer Applications
Thapar Institute of Engineering and Technology

Mrs. Rinkle Aggarwal
Lecturer
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology



Computer Science and Engineering Department
Thapar Institute of Engineering and Technology
Deemed University
Patiala

DECLARATION

I declare that the work presented in this thesis is, to the best of my knowledge, original and my own work, except as acknowledged in the text, and that it has not been submitted, either in whole or part, for a degree at this, or any other, university.

Meenu Bhagat

Regd. No. 8013106

CERTIFICATE

This is to certify that the thesis work entitled, "Spelling error pattern analysis of Punjabi typed text" submitted by Meenu Bhagat, in the partial fulfillment of the requirement for the award of degree of Master of Engineering in Software Engineering at Thapar Institute of Engineering & Technology (Deemed University), Patiala, is a record of candidate's own work carried out by her under my supervision and guidance.

Dr. Gurpreet Singh Lehal
Asst. Professor, SMCA
TIET,
Patiala

Ms. Seema Bawa
Head, CSED
TIET,
Patiala

Mrs. Rinkle Aggarwal
Lecturer, CSED
TIET,
Patiala

Dr. D.S. Bawa
Dean,
Academic Affairs
TIET, Patiala

The M.E. (Thesis) viva-voce examination of Meenu Bhagat, Regd. no. 8013106, M.E. (Software Engineering), Thapar Institute of Engineering & Technology (Deemed University), Patiala has been held on ……

Supervisor External Examiner

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Dr. G.S.Lehal, Assistant Professor, School of Mathematics & Computer Applications, for providing invaluable guidance, suggestions and sympathetic attitude, which inspired me to submit this thesis report on time. I also wish to express my gratitude to Mrs. Rinkle Aggarwal, Lecturer, Computer Science and Engg. Department, for her valuable advises and suggestions.

I am also thankful to Ms. Seema Bawa, Head, Computer Science and Engg. Department, for her kind help and cooperation.

I would also like to thank all the staff members of Computer Science and Engineering Department for providing me all the facilities required for the completion of this work.

I would like to extend my special thanks to Prof. B. B. Chaudhuri, Professor & Head Computer Vision & Pattern Recognition Unit, Indian Statistical Institute, Calcutta for providing me necessary papers.

I am deeply indebted to my parents and brother for the inspiration and ever encouraging moral support, which enabled me to pursue my studies.

Meenu Bhagat

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Error pattern analysis of a language is useful in language related technology development, such as Spell Checker and Corrector, Optical Character Recognition, Machine Translation, Natural Language Interfaces etc. Error pattern analysis includes analysis of various types of errors (insertion, deletion, transposition, substitution, run-on, split word error) positional analysis, word length effects, phonetic errors, first position error analysis, keyboard effects etc. Though considerable work has been done in the area for English and related languages, the Indian Language scenario presents a relatively more complex and uphill task. In this thesis, I have presented a statistical error analysis for Punjabi, the world's 14th most widely spoken language. For this purpose I have collected about 20000 misspelled words generated by typists. The application of the error analysis in designing the suggestion list for a Punjabi spell checker is also discussed.

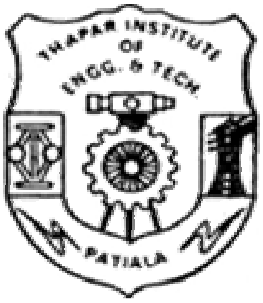
Spelling Error Analysis of Punjabi Typed Text

A Thesis Report
Submitted in the partial fulfillment of the requirements
for the award of the degree of ME in Software Engineering

Submitted by
Meenu Bhagat
M.E. (Software Engineering)
8013106

Under the supervision of:
Dr. Gurpreet Singh Lehal
Assistant Professor
School of Mathematics & Computer Applications
Thapar Institute of Engineering and Technology

Mrs. Rinkle Aggarwal
Lecturer
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology



Computer Science and Engineering Department
Thapar Institute of Engineering and Technology
(Deemed University)
Patiala

CHAPTER 1

INTRODUCTION

1.1 Introduction

The problem of detecting error in words and automatically correcting them is a great research challenge. Its solution has enormous application potentials in text and code editing, computer aided authoring, optical character recognition (OCR), machine translation (MT), natural language processing (NLP), database retrieval and information retrieval interface, speech recognition, text to speech and speech to text conversion, communication system for the disabled (e.g. blind and deaf), computer aided tutoring and language learning, desktop publication and pen-based computer interface.

The word-error can belong to one of the two distinct categories, namely, *nonword error* and *real-word erro*r. Let a string of characters separated by spaces or punctuation marks be called a candidate string. A candidate string is a valid word if it has a meaning. Else, it is a nonword. By real word error we mean a valid but not the intended word in the sentence, thus making the sentence syntactically or semantically erroneous word and either suggest correct alternatives or automatically replace it by the appropriate word.

There are several issues to be addressed in the error correction problem. The first issue concerns the error patterns generated by different text generating media such as typewriter and computer keyboard, typesetting and machine printing, OCR system, speech recognizer output, and of course, handwriting. Usually, the error pattern of one media does not match with that of the other. The error pattern issue of each media concerns the relative abundance of insertion, deletion, substitution and transposition error, run-on and split word error, single versus multiple character error, word length effect, positional bias, character shape effect, phonetic similarity effect, heuristic

tendencies etc. The knowledge about error pattern is necessary to model an efficient spell checker.

Another important issue is the computerized dictionary which concerns the size of the dictionary, the problem of inflection and creative morphology, the dictionary file structure, dictionary partitioning, word access techniques and so on. Dictionary look up is one of the two principal ways of spelling error detection and correction. The other approach, popularly used in OCR problems, is the N-gram approach. Construction of appropriate N-gram from raw text data is an important issue in this approach.

The detection of real word error needs higher-level knowledge compared to the detection of non-word error. In fact, detection of real word error is a problem that needs NLP tools to solve. Quite often, it is not possible to separate the problem of real error detection from that of correction.

Even for nonword errors, correction is a nontrivial task. Several approaches based on minimum edit distance; similarity key, rules, N-grams, probability and neural nets are proposed to accomplish the task. 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 [1,2]. 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.

This is the first time that a detailed error analysis for Punjabi is being carried out. For this purpose we have collected about 20000 misspelled words generated by typists, both novice and experienced as well as students learning Punjabi typing. We have done analysis of six main categories of errors. These errors are discussed in detail in following sections. Based on this error analysis, we have also designed a suggestion list for a Punjabi spellchecker.

1.2 A Brief Overview of Indian languages

There are 15 officially recognized Indian scripts. These scripts are broadly divided into two categories namely, Brahmi scripts and Perso-Arabic scripts. The Brahmi scripts consist of, Devanagari, Gurmukhi, Gujarati, Oriya, Bengali, Assamese, Telugu, Kannada, Malayalam, Tamil. And the Perso-Arabic scripts include Urdu, Sindhi and Kashmiri. Devanagari script is used by Hindi, Marathi and Sanskrit languages. The characteristics of the languages within the family are quite peculiar. They have the common phonetic structure, making the common character set. Within the same family again north Indian scripts like, Hindi, Marathi, Gurmukhi, Gujarati, Oriya, Bengali, Assamese have common features while Southern scripts like Tamil, Telugu, Kannada and Malayalam have common features. This clear division of characteristics has simplified the use on computers.

All these scripts mentioned above are written in a nonlinear fashion. Unlike English, the width of the characters is different even on a same script. The division between consonant and vowel is applied for all Indian scripts. The vowels getting attached to the consonant are not in one (or horizontal) directions; they can be placed either on the top or the bottom of consonant. This makes the use of the scripts on computers more complicated to represent them.

1.2.1 Structure of Indian Scripts [3]

Since the origin of all these scripts is same, they share a common phonetic structure. The alphabet may vary slightly and also the graphical shapes. Using this characteristic a transliteration facility between any Indian scripts is possible. Typically the alphabets get divided into following categories:

The Consonants-

All Indian scripts use 5 types of consonants groups, called varga. Some of the vowel like a is included in the consonant category. Each varga has 5 consonants, with primary and secondary pairs. The second consonant in each pair is derived from the first consonant with 'h' sound, and have separate graphical representation (see Table 1.1) .

Table 1.1 Table of Consonants and Nasals

Consonants				Nasal	Example
K	Kha	Ga	Gha	Na	gangA
Ca	Cha	Ja	Jha	N-a	manc
Ta	THa	Da	Dha	N	ghantA
Ta	Tha	Da	Dha	N	sant
Pa	Pha	Ba	Bha	M	stambha

Other consonants not present in this category are,
Ya Ra La Va 'Sa S.a Sa Ha
and invisible consonant like, Ra (halant) and (halant) Ra, get formed differently.

Vowels-

All the vowels are represented by separate symbols. These vowels are placed on the consonants either in the beginning or after the consonant. Each of these vowels are pronounced separately. Typical vowels are,
Vowel: A, i, Ee, u, U, ru, Ee
Usage: Ka, Ki, Kee, Ku, KU, Kru, and Kee

Vowel: e, E, a, o, O, au, ao

Usage: Ke, KE, Ka, Ko, KO, Kau, Kao

Halant-

While forming the conjuncts a use of broken consonants is activated by halant. On mixing of two or more consonants the shape of the conjunct varies. Many a times halant is required to indicate the vowel-less ending.

e. g. Ramnathan

Nukta-

Nukta is used to derive some of the characters used in Hindi, Punjabi and Urdu. d. or k. etc.

Punctuations and Numerals-

All the punctuations and numerals are common between English and Indian scripts. They are used on the computers using English symbols.

Vedic characters-

Apart from Hindi and Marathi, Devanagari has Sanskrit language, which uses Vedic symbols. The provision of these symbols is made by keeping the extended character set.

1.2.2 Text Processing

To use language for any applications its characteristics are required to be known. Once this is known, the application can make use of languages in a most uniform manner. Indian scripts have a very different structure and have communality amongst them. They follow almost same rules; the way of representing them is different.

Text processing on typewriter and computers

The text processing of Indian scripts on the mechanical typewriter and on computers is different. There are some limitations on the mechanical typewriter as compared to computers. Due to the complex nature of Indian script, formation of conjuncts is extremely difficult on the typewriters. The approach used in typewriters is most suited for

graphical representation. This is not a very user-friendly approach. At the same time it does not suit all Indian languages. Using computer the text processing of any kind is possible with the help of software and hardware. Specially in case of forming conjuncts the shapes of the characters vary, these various shapes can be provided on computers by software. On the other hand the output on the typewriter is not upto the mark. Many a times simultaneous use of English along with Indian script is required. This usage is made possible by standardizing the codes.

Indian scripts have tremendous applications in day-today life. These applications include, Word processing, Database management, DTP, Teleprinting, machine translation, OCR etc. Once the characteristics of the scripts are known, making use of them for any of these applications is possible. Standard code chart is designed for dedicated applications as well as with English. For any of this use, once the sequence of characters is known to form words or conjuncts, they can be sent to or received from the device and formation of exact word is done by software.

1.3 Introduction to Gurmukhi

The word 'Gurmukhi' literally means from the mouth of the Guru. Gurmukhi script is used primarily for the Punjabi language, which is world's 14th most widely spoken language. Punjabi is named after Punjab, which was divided between India and Pakistan during Partition in 1947. Punjab literally means land of five rivers; Punj meaning five and Aab, water. Gurmukhi script is syllabic in nature. Gurmukhi script-consists of 41 consonants called *vianjans*, 9 vowel symbols called *laga* or *matras*, 2 symbols for nasal sounds (N, °), one symbol for reduplication of sound of any consonant (`)and three half characters.

Consonant

a A e
Matra Vahak
s h Mul Varag
k K g G | Kavarg Toli
c C j J \ Chavarg Toli
t T f F x T æavarg Toli
q Q d D n Tavarg Toli
p P b B m Pavarg Toli
X r l v V Antim Toli
S ^ Z z & L Naveen Toli

Vowels

w ,i , I , u , U , y , Y, o , O

Semi-Vowels

N , ° , `

Half Characters

HH R Í

Fig. 1.1: Gurmukhi Vocabulary

The consonants of first row (a,A,e) are classified as open syllabics and called vowel consonants or semi consonants or "Matra Vahak" due to their inherent property that they are never used in work without any 'Laga' or 'Vowel'. The next two consonants are classified as root class consonants. The rest of the consonants except to the last two groups namely the - "Antim" and "Naveen" group, are categorized according to their phonetic structure.

There are five such categories namely the Kavarg toli, Chavarg toli, Tavarg toli and the Pavarg toli depending upon the different organs like throat, palate, mouth, tongue and lips, using which they are pronounced or from where they originate.

The last but one group consisting of 5 independent consonants (X,r,l,v,V) is called the "Antim" group and the last group (S,^,Z,z,&,L) is the "Naveen" group which has been introduced to accommodate the words of Persian, Arabic and Sanskrit.

It can be noted that most of the characters have a horizontal line at the upper part. The characters of word are connected mostly by this line called head line. A word in Gurmukhi script can be partitioned into three horizontal zones. The upper zone denotes the region above the head line, where the vowels reside, while the middle zone represents the area below the head line where the consonants and some sub-parts of vowels are present. The middle zone is the busiest zone. The lower zone represents the area below middle zone where some vowels and certain half characters lie in the foot of consonants.

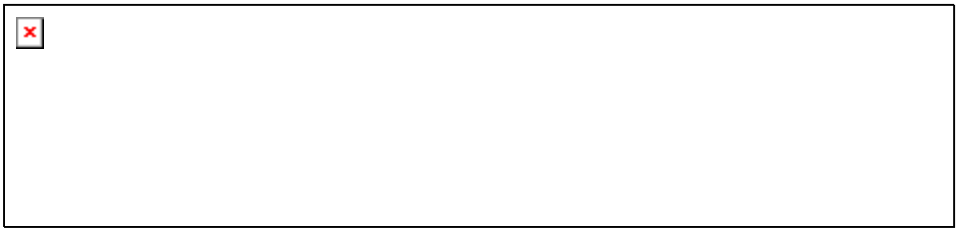


Fig. 1.2: Three zones of a word

1.3.1 Gurmukhi Orthography

Gurmukhi is written from left to right. The characters are normally aligned below the line of writing. The alphabet is also sometimes called 'the thirty - five ', from the fact that the basic repertoire of consonant and consonant - like symbols numbers thirty-five. Gurmukhi script like most of other Indian language scripts is written in a nonlinear fashion. The width of the characters is also not constant. The vowels getting attached to the consonant are not in one (or horizontal) directions; they can be placed either on the top or the bottom of consonant. There are fewer uncertainties and irregularities in either the reading or the spelling rules that are usual in languages of South Asia. For example there is only one way to write homorganic nasal clusters, in contrast with the three that are available in Hindi. The result is much less variation in spelling, and less uncertainty as to the correct spelling.

CHAPTER 2

SURVEY OF LITERATURE

2.1 Non-word error detection

A string of characters separated by spaces or punctuation marks may be called a candidate word. A candidate word is a valid word if it has a meaning; else it is a non-word. The two main techniques that have been explored for nonword errors detection are n-gram analysis and dictionary lookup. N-grams are n-letter subsequences of words or strings where n usually is one two or three .One letter n-grams are referred to as *uni-grams* or *monograms*; two letter n-grams are referred to as *bi-grams* and three letter n-grams as *trigrams.*

In general n-gram detection technique work by examining each N-gram in an input string and looking it up in a precompiled table of n-gram statistics to ascertain either its existence or its frequency of words or strings that are found to contain nonexistent or highly infrequent n-grams are identified as either misspellings. N-gram techniques usually require either dictionary look up technique or a large corpus of text in order to pre-compile an n-gram table.

Dictionary looks up techniques work by simply checking to see if an input string appears in a dictionary, i.e., a list of acceptable words.

2.1.1 N-gram Analysis Techniques

Text Recognition systems usually focus on one of three modes of text: handwritten text, hand-printed text or machine printed text. All three modes may be processed by optical character recognition devices.

N-gram tables can take on a variety of forms. The simplest is a binary bi-gram array which is a two dimensional array of size 26x26 whose elements represent all possible two letter combinations of the alphabet. The value of each element in the array is set to either 0 or 1 depending on whether that bi-gram occurs in at least the word in a predefined lexicon or dictionary .A *binary tri-gram* array would have three dimensions. Both of the above arrays are referred to as **non-positional** binary n-gram arrays because they do not indicate the position of the n-gram within a word.

More of the structures of the lexicon can be captured by a set of positional binary n-gram array, For example in a positional binary tri-gram array the i, j, kth element would have the value 1 if only if there exists at least one word in the lexicon with the letters l, m and n in positions i, j, and k. The trade-off for representing more of the structure of the lexicon is the increase in storage space required for the complete set of positional arrays. Any word can be checked for errors by simply looking up its corresponding entries in binary n-gram arrays to make sure they are all 1's.

2.1.2 Dictionary lookup Techniques

The most popular method of detecting errors in a text is simply to look up every word in a dictionary; any words that are not there are taken to be errors. Dictionary lookup is a straightforward task. However response time becomes a problem when dictionary size exceeds a few hundred words. In document processing and information retrieval, the number of dictionary entries can range from 25000 to more than 250,000 words.

The most common technique for gaining fast access to a dictionary is the use of a hash table .To look up an input string, one simply computes its hash address and retrieves the word stored at that address in the pre-constructed hash table. If the word stored at the hash address is different from the input string or is null, a misspelling is indicated.

Turba [4] provides a quick review of the pros and cons of using hash table for the dictionary look up. The main advantage is that the random access nature on a hash code eliminates the large no. of comparisons needed for sequential or even tree based searches of the dictionary. The main disadvantage is the need to devise a clever hash function that avoids collisions without requiring a huge hash table.

2.1.3 Dictionary Correction issues

A Lexicon for a spelling correction or text recognition application must be carefully tuned to its intended domain of discourse. Too small a lexicon can burden the user with too many false rejections of valid terms; too large a lexicon can result in an unacceptable high number of false acceptances i.e. genuine mistakes that went undetected because they happened to form valid low frequency or extra-domain words (for example lave, fen, verry, etc) but the relationship between misspellings and word frequencies is not straightforward.

Dictionaries themselves are often insufficient sources for lexicon construction. **Walker and Amsler[5]** observed that nearly two thirds (61%) of the words in the Merriam-Webster Seventh Collegiate Dictionary did not appear in an eight million-word corpus of New York Times News wire text, and conversely almost two thirds (64%) of the words in the text were not in the dictionary.

2.1.4 The Word Boundary Problem

For virtually all spelling error detection and correction techniques, word boundaries are defined by white space characters (for example blanks, tabs, carriage returns etc.).This assumption turns out to be problematic since a significant portion of text errors involve running together two or more words, sometimes with intrinsic errors (for example ofthe, understandhme) or splitting a single word (for example sp ent, th ebook). **Kukich[6]** found that a full 15% of all nonword-spelling errors in 40000-word corpus of typed

textual conversations involved this type of error (i.e. 13% were run-on words and 2% were split words).

Mitton[7] found that run-on words and splits frequently result in at least one valid word (for example forgot→ for got, in form→ inform) so one or both such errors may go undetected.

Jones et al.[8] found that OCR devices are more likely to split words than to join them. The difficulty in dealing with run-ons and split lies in the fact that weakening the word boundary constraint results in a combinatorial explosion of the number of possible word combinations or subdivisions that must be considered.

2.2 Isolated-word error correction

Simply detecting errors in text may be sufficient for some applications, but for most applications detection alone is not enough. For example, since the goal of text recognition devices is to accurately reproduce input text, output errors must be both detected and corrected. Similarly users have come to expect spelling checkers to suggest corrections for the non-words they detect. Indeed, some spelling correction applications, such as text to speech synthesis, require that errors be both detected and corrected without user intervention. To address the problem of correcting words in text, a variety of **Isolated – word error correction** techniques have been developed.

The characteristics of different applications impose different constraints on the design of isolated word error correctors and many successful correction techniques have been devised by exploiting application specific characteristics and constraints. Most application specific design considerations are related to three main issues: (1) **Lexicon issues** (2) **Compter human interface issues** and (3) **Spelling error pattern issues**.

Lexicon issues include such things as lexicon size and coverage, rates of entry of new term into the lexicon, and whether morphological processing, such as affix handling, is required.

Compter human interface issues include considerations as whether real time responses is needeed,whether the computer can solicit feedback from the user ,how much accuracy is required on the first guess, etc.. An empirical study by Durham demonstrated ,among other things, that considerable benefits accrued for a simple, speedy algorithm for correcting single error types that only corrected one quarter the spelling mistakes made by the users but did so in a unobtrusive manner.

Spelling Error pattern issues include such things as what the most common errors are , how many errors tend to occur within in a word whether error tend to change word length,whether misspellings are typographically ,cognitively, or phonetically based and in general whether errors can be characterised by rules or by probabilistic tendencies. Spelling error pattern issues have had perhaps the greatest impact design of correction techniques.

2.2.1 Spelling Error Patterns

Spelling error patterns vary greatly depending on the application task, for example **transcription typing errors**,*which are the most part due to motor coordinatin slips*,tend to reflect typewriter keyboard adjacencies.For example the substitution of **D** for **O**. More subtly even two similar text entry modes, such as transcription typing and conversational text typing, for example electronic mail may exhibit significantly different error frequency and distribution statistics due to greater cognitive overhead in the later task .

Distinctions are made between three different types of nonword misspellings :

(1) **Typographic errors** (2) **Cognitive Errors** (3) **Phonetic Errors**

Typographic errors : In the case of typographic errors it is assumed that the writer or typist knows the correct spelling but simply makes a motor coordination slip. For example the→ teh, spell→speel .

Cognitive Errors : The source of cognitive errors is presumed to be a misconception or lack of knowedge on the part of the writer or typist.For example receive→recieve ,conspiracy→conspiricy .

Phonetic Errors : Phonetic errors are a special class of cognitive errors in which the writer substitutes a phonetically correct but orthographically incorrect sequence of letters for the intended word , for example abyss→abiss, naturally→nacherly .

2.2.1.1 Basic Error Types

One of the first general findings on human generated spelling errors as observed by Damerau in 1964 and has been substantiated for many applications since then. **Damerau[9]** found that approximately 80% of all misspelled words contained a single instance of one of the following four types of errors: **insertion**, **deletion**, **substitution** and **transposition**. Misspellings that fall into this large class are often reffered to as **single error misspellings**; misspellings that contain more than one such error have been dubbed **multi-error misspellings.**

The 80% single error rule cannot be taken granted for all application however **Pollock and Zamora[10]** found that only 6% of 50000 nonword spelling errors in the machine readable databases they studied were multierror misspellings.Coversely ,**Mitton[7]** found that 31% of the misspellings in his 17001 word corpus of handwritten essays contained multiple errors.

OCR generated misspellings do not follow the pattern of human generated misspellings. Most error correction techniques for OCR output assume that the bulk of errors will be substitution errors. However , **Jones et al. [8]** report that :

" *The types of OCR errors that occur vary widely, not only from one recognizer to another but also based on font,input quality and other factors* " and he also report that " *a significant fraction of* errors *are not one to one errors (for example ri → n, m → iii)".*

Rhyne and Wolf [11] classify recognition errors into four categories: (1) substitutions (2) failures (no character exceeds a minimal recogition threshold); (3) insertions and deletions and (4) framing errors (one to one mapping failures).

2.2.1.2 Word Length Effects

According to **Zipfs law[12],** short words occur more frequently than long words and according to empirical study by **Landauer and Steeter[13]**, high frequency (i.e. short) words tend to have more single error neighbours than low frequency words thus making it difficult to select the intended correction from its set of neighbours. On the other hand if spelling errors occurs less frequently in short words then the problem may be less pressing.

Pollock and Zamora's study[14] of 50000 nonword errors indicated that errors in short words are indeed problematic for spelling correction even though their frequency of occurance may be low. They state that " *although 3-4 character misspellings constitute only 9.2% of total misspellings they generate 42% of the miscorrection.*"

Kukich[15] analyzed over 2000 error types in a corpus of TDIL conversations and found that over 63% of the errors occurred in words of length 2,3,4 characters.

2.2.1.3 First position errors

It is generally believed that few errors tend to occur in the first letter of a word. **Pollock and Zamora[14]** found that 3.3% of the 50000 misspellings involved first letter and **Yannakoudakis and Fawthrop[16]** observed a first position error rate of 1.4% in 568

typing errors.**Mitton[7]** found that 7% of all the misspellings he studied involved first position errors.In contrast Kukich observed a 15% first position error rate in a 40000 word corpus of typed textual conversations.

2.2.1.4 Keybord Effects

Gruddin[17] performed an extensive analysis for the typing errors made by six expert typists and eight novice typists while transcribing magazine articles totaling about 60000 characters of text. He found large individual differences in both typing speed and types of errors made. For example, error rates ranged from 0.4% to 0.9% for experts and averaged 3.2% for novices ; the majority of expert errors were insertions that resulted from hitting two adjacent keys simultaneously while the majority of novice errors were substitutions .

2.2.1.5 Error Rates

Data concerning the frequency of occurance of spelling errors is not abundant.The few data points that must be qualified by :

1) The size of the corpus from which they were drawn.
2) The text entry mode of the corpus, for example, handwritten ,transcription typing, conversational typing, edited machine readable text etc. .
3) The date of the study, since newer studies for some genres, such as edited machine- readable text, probably reflect lower error rates due to the availability of automatic spelling checkers.

Pollock and Zamora[10] examined a set of machine readable databases containing 25 million words of text.They found over 50000 nonword errors for a spelling error rate of 0.2%. An even lower rate holds for a corpus of AP (Associated press) news wire text studied by **Church** and **Gale[18].**

Spelling error rates of 1.5% and 2.5% for handwritten text has been reported by **Wing** and **Baddeley[19] and Mitton[7]** repsectively.

2.2.1.6 Phonetic Errors

Examples of applications in which phonetic errors abound include :

1) The automatic yellow pages like information retrieval service available in the French Minitel system.

2) A directory assistance progarm for looking up names in a corporate directory,a database interface for locating individuals by surname in large credit insurance ,motor vehicle bureau and law enforcement databases .

Two data points on the frequency of occurence have been documented.One is provided by **Van berkel and DeSmedt[20] .**They had 10 Dutch subjects transcribe a tape recording of 123 Dutch surnames randomly chosen from a telephone Directory.They found that 38% of the misspellings generated by the subjects were incorrect despite being phonetically plauisble.

2.2.1.7 Hieuristic Rules and Probabilistic Tendencies

Yannakoudakis and Fawthrop[16] sought a general characterization of misspelling behaviour. They compiled a database of 1377 spelling errors culled from a variety of sources, including mistakes found in the Brown corpus as well as those found in a 60,000 word corpus of text typed by the university of Bradford who believed themselves to be very bad spellers. They found that a large portion of the errors could be accounted by a set of 17 hieuristic rules, 12 of which related to the misuse of consonants or vowels in graphemes, and five of which related to sequence production. For example, heuristics related to consonants included :

1) The letter h is frequently omitted in words containing the graphemes ch, gh, ph and rh as in the misspellings against and *tecniques* and

2) Doubling and singling of consonants which frequently occur doubled is a common error.

3) The most frequent length of a misspelliing is one letter short of the correct spelling .

4) Typing errors are caused by hitting an adjacent key on the keyboard or by hitting two keys together.

5) Short misspellings do not contain more than one error and so on .

2.2.2 Techniques for Isolated word Error Correction

The problem of isolated word error correction entails three subproblems:

1) Detection of an error
2) Generation of candidate corrections
3) Ranking of candidate corrections

A convenient way to organize the approaches is to group them into six main classes:

1) Minimum edit distance techniques
2) Similarity key techniques
3) Rule based techniques
4) N-gram based techniques
5) Probabilistic techniques
6) Neural nets

For the sake of providing some insight into how techniques differ in their scope and accuracy, some addition relevant characeristics are reported whenever that information is available .These include:

1) Lexicon size
2) Test set size

3) Correction accuracy for single error misspellings

4) Correction accuracy for multi-error misspellings

5) Word length limitations

6) Type of errors handled

2.2.2.1 Minimum Edit distance techniques

The term minimum edit distance was defined by **Wagner[21]** as the minimum number of editing operations (i.e.insertions,deletions and substitutions) required to transform one string into another.The first minimum edit distance spelling error correction algorithm was implemented by **Damerau[1]**.

Although many minimum edit distance algorithms compute integer distance scores,some algorithms obtain finer-grained scores by assigning non integer values to specific transformations based on estimates of phonetic similarities or keyboard adjacencies.

2.2.2.2 Similarity Key techniques

The notion behind similarity key techniques is to map every string into a key such that similarly spelled strings will have identical or similar keys.Thus when key is computed for a misspelled string it will provide a pointer to all similarly spelled words in the lexicon.Similarity key techniques have speed advantage because it is not necessery to directly compare the misspelled string to every word in the dictionary.

A very early ,often cited similarity key technique , the SOUNDEX system, was patented by Odell and Russel for use in phonetic spelling correction applications .

2.2.2.2.1 The Soundex System

This problem has been around for a long time in the context of retrieving names from a list of names. Suppose you are working at an enquiry desk of a large organization, with a terminal connecting your office to the central computer. A customer comes in with a query about her account. She says her name is *Zbygniewski.* You don't want to ask her to spell it - perhaps her English is poor and other customers are waiting. To make matters worse, the name may be misspelt in the computer file. You want to be able to key in something that sounds like what she just said and have the system find a name that resembles it.

The Soundex system was devised to help with this problem **(Knuth[22],Davidson [23])**. It dates, in fact, from the days of card indexes - the name stands for 'Indexing on sound' - but has been transferred to computer systems. A Soundex code is created for every name in the file. The idea of the code is to preserve, in a rough-and-ready way, the salient features of the pronunciation. Vowel letters are discarded and consonant letters are grouped if they are likely to be substituted for each other - an *s* may be written for a *c,* for instance, but an *x* for an *m* is unlikely. The details are presented in Table 2.1, with some examples.

1) Keep the first letter (in upper case).
2) Replace these letters with hyphens: *a, e, i, o, u, y, h, w.*
3) Replace the other letters by numbers as follows:
   b,f,p,v : 1
   c,g,j,k,q,s,x,z : 2
   d,t : 3
   l : 4
   m,n : 5
   r : 6
4) Delete adjacent repeats of a number.
5) Delete the hyphens.
6) Keep the first three numbers or pad out with zeros.

Table 2.1: The Soundex code, with some examples

Birkbeck	Zbygniewski	toy	car	lorry	Bicycle
B-621-22	Z1-25---22-	T--	C-6	L-66-	B-2-24-
B-621-2	Z1-25---2-	T--	C-6	L-6-	B-2-24-
B621	Z125	T000	C600	L600	B224

So, every name in the file has one of these codes associated with it. The name *Zbygniewski* has code Z125, meaning that it starts with a *Z,* then has a consonant in group 1 (the *b*), then one in group 2 (the *g*) and then one in group 5 (the *n*), the remainder being ignored. Let's say you key in *Zbignyefsky.* The computer works out the Soundex code for this and retrieves the account details of a customer with the same code - *Zbygniewski* - or perhaps the accounts of several customers with somewhat similar names.

It is fairly obvious how this system can be applied to spelling correction. Every word in the dictionary is given a Soundex code. A Soundex code is computed from the misspelling, and those words that have the same code are retrieved from the dictionary. Take as an example the misspelling *disapont.* A corrector would compute the code D215 from *disapont* and then retrieve all the words with code D215: *disband, disbands, disbanded, disbanding, disbandment, disbandments, dispense, dispenses, dispensed, dispensing, dispenser, dispensers, dispensary, dispensaries, dispensable, dispensation, dispensations, deceiving, deceivingly, despondent, despondency, despondently, disobeying, disappoint, disappoints, disappointed, disappointing, disappointedly, disappointingly, disappointment, disappointments, disavowing.*

Pollock and Zamora[10] used in findings of their study of 50000 spelling errors from seven chemical abstract service databases to devise a similarity key technique, called SPEEDCOP.

2.2.2.2.2 The SPEEDCOP System

The purpose of the SPEEDCOP project was to devise a way of automatically correcting spelling errors - predominantly typing errors - in a very large database of scientific abstracts. A key was computed for each word in the dictionary. This consisted of the first letter, followed by the consonant letters of the word, in the order of their occurrence in the word, followed by the vowel letters, also in the order of their occurrence, with each letter recorded only once, for example the word *xenon* would produce the key *XNEO* and *inoculation* would produce *INCLTOUA.* The words in the dictionary were held in key order, as illustrated in Figure 2.1.

PLTDOE	plotted
PLTE	pellet
PLTEI	pelite
PLTIO	pilot
PLTNGAI	plating
PLTNSUO	plutons
PLTNUO	pluton
PLTOU	poult

Fig. 2.1: A Section of the SPEEDCOP dictionary

When the system was given a misspelling, such as *platin,* it computed the key of the misspelling and found its place in the dictionary. In this example, the key of *platin* would be *PLTNAI,* which would come between *PLTIO* and *PLTNGAI.* Moving alternately forwards and backwards from that point, it compared the misspelling with each of the words to see if the misspelling could be a single-error variation on that word, until either it had found a possible correction or had moved more than fifty words away from its starting point. The SPEEDCOP researchers found that, if the required word was in the dictionary, it was generally within a few words of the starting point. In the example, the

corrector would quickly find the word *plating* as a possible correction (*platin* being an omission-error variant of *plating*).

The Soundex code and the SPEEDCOP key are ways of reducing to a manageable size the portion of the dictionary that has to be considered. Confining the search to words of the same length (plus or minus one) restricts the search even further. The price to be paid is that, if the required word is outside the set of those considered, the corrector is not going to find it.

2.2.2.3 Rule Based Techniques

Rule Based Techniques are algorithms or heuristic programs that attempt to represent knowledge of common spelling error patterns in the form of rules for transforming misspellings into valid words. The candidate generation process consists of applying all applicable rules to a misspelled string and retaining every valid dictionary word that results. Ranking is frequently done by assigning a numerical score to each candidate based on a predefined estimate of the probability of having made the particular error that the invoked rule corrected.

2.2.2.4 N-gram Based Techniques

Letter n-grams, including tri-grams, bi-grams and uni-grams have been used in a variety of ways in text recognition and spelling correction techniques. They have been used by OCR correctors to capture the lexical syntax of a dictionary and to suggest legal corrections.

Riseman and Hanson[24] provide a clear explanation of the traditional use of n-grams in OCR correction. After partitioning a dictionary into sub-dictionaries by word length, they construct positional binary n-gram matrices for each sub-dictionary. They note that because these matrices provide answers to the question “is there some word in the

dictionary that has letters α and β in positions i and j respectively," the matrices capture the syntax of the dictionary.

An OCR output string can be checked for errors by simply checking that all its n-grams have value 1. If a string has a single error, it will have a 0 value in at least one binary n-gram; if more than one 0 value is found, the position of the error is indicated by a matrix index that is common to the 0 value n-grams. If the intersection results in only one n-gram with value 1, the error can be corrected, if more than one potential n-gram correction is found the word is rejected as ambiguous.

2.2.2.5 Probabilistic Techniques

N-gram based techniques led naturally into the probabilistic techniques in both the text recognition and spelling correction paradigms. Two types of probabilities have been exploited:

- **Transition Probability**
- **Confusion Probability**

Transition Probabilities represent probabilities that a given letter will be followed by another given letter. These are language dependent. They are sometimes referred to as *markov probabilities* based on the assumption that language is a Markov Source. They can be estimated by collecting n-gram frequency statistics on a large corpus of text from the domain or discourse.

Confusion Probabilities are estimates of how often a given letter is mistaken or substituted for another given letter. Confusion probabilities are source dependent. Because different OCR devices use different techniques and features to recognize characters, each device will have a unique confusion probability distribution. For that reason confusion probabilities are sometimes referred to as *channel characteristics*.

2.2.2.6 Neural Net techniques

Neural nets are likely candidates for spelling correctors because of their inherent ability to do associative recall based on incomplete or noisy input. Further more, because they can be trained on actual spelling errors, they have the potential to adapt to the specific error patterns of their user community, thus maximizing their correction accuracy for that population.

Back propagation algorithm is the most widely used algorithm for training a neural net. A typical back propagation net consists of three layers of nodes: input layer, an intermediate layer, usually referred to as hidden layer and an output layer. Each node in the input layer is connected by a weighted link to every node in the hidden layer. Similarly each node in the hidden layer is denoted by a weighted link to every node in the output layer. Input and output information is represented by on-off patterns of activity on the input and output nodes of the net. A 1 indicates that a node is turned on and a 0 indicates that a node is turned off.

Processing in a back propagation net consists of placing a pattern of activity on the input nodes, sending the activity forward through the weighted links to the hidden nodes, where a hidden pattern is computed and then on to the output nodes, where an output pattern is computed. Weights represent connection strengths between the nodes. For example, In a spelling correction application, a misspelling represented as a binary n-gram vector might serve as an input pattern to the net. Its corresponding output pattern might be a vector of *m* elements, where *m* is the number of words in the lexicon and only the node corresponding to the correct word is turned on.

CHAPTER 3

ERROR ANALYSIS OF PUNJABI SPELLINGS

3.1 Introduction

Error pattern analysis of each language helps in making an efficient spellchecker. It includes analysis of various types of errors (insertion, deletion, transposition, substitution, run-on, split word error) positional analysis, word length effects, phonetic errors, first position error analysis, keyboard effects etc.

This is the first time that a detailed error analysis for Punjabi is being carried out. For this purpose, I have collected about 20000 misspelled words generated by typists, both novice and experienced as well as students learning Punjabi typing. I have done analysis of six main categories of errors. These errors are discussed in detail in following sections.

3.2 Data Collection and Analysis

I have collected the material from type colleges, professional typists and government institutions and private printing presses and every document is carefully checked and the misspelled words are manually collected and analyzed. Out of text containing more than eight lakh words around 20000 misspellings are found.

In Punjabi there are many ways of writing the same word and all the ways could be correct. For example, 5ircY→5rIcY→5ircX→5rIcX. All the four words are delivering the same meaning and are the different iterations of the same word. So it might be wrong to collect the raw typed text as the data for analysis. Because analysis of that raw text does not surely direct us to the *typing mistake* but can mislead us to the *spelling mistake* of that word. Our main interest is to analyze the typing mistakes instead of

spelling mistakes since the study will be used to design a suggestion list for a Punjabi Spellchecker. I have made a careful analysis of each and every word and collected information like single/multi-error misspellings, mistake positions and word length analysis, types of errors for single/multi-error misspellings, special character errors, errors related to vowels, phonetic occurrences etc.

3.3 Reverse Minimum edit distance algorithm

In reverse techniques, a candidate set is produced by first generating every possible single-error permutation of the misspelled string and checking the dictionary to see if any make up valid words. This means that for a given misspelled string of length n and an alphabet of size 57, the number of strings that must be checked against the dictionary is 57(n+1) insertions plus n deletions plus 56n substitutions plus n-1 transpositions, or 115n+56 strings, assuming there is only one error in the misspelled string.

3.4 Nature of Errors

I have divided my analysis work into following categories:

1) Type of error: Substitution, Insertion, Deletion, Transposition, Run-on, Split word error.
2) Positional analysis: Based on the position at which mistake occurs.
3) Word length Effect: Analysis based on the number of characters in the word.
4) Number of mistakes in a misspelling
5) Phonetically Similar Character Analysis
6) First Position error Analysis
7) Other Findings

All the collected misspellings were sorted out for single/multi-error misspellings. Out of the total no. of misspellings 91.13% were the single error misspellings and 8.87% were multi error misspellings.

3.4.1 Error Pattern based on Type of Error

I have done an analysis of six types of errors i.e.

1) **Insertion error (IE):** When at least one extra character is inserted in the desired word.
2) **Deletion error (DE):** When at least one character is deleted in the desired word.
3) **Substitution error (SE):** When at least one character is substituted by the other character. The maximum of misspellings in Punjabi contain substitution errors.
4) **Transposition error (TE):** When two adjacent characters are transposed.
5) **Run-on Error (ROE):** When there is space missing between two or more valid words**.**
6) **Split Word error (SWE):** This is opposite of Run-on error when there is some extra space is inserted between parts of a word. The error can be removed by removing the extra space.

It is analysed that error rate is at its peak due to substitution errors in single as well as multi-error misspellings. In single error misspellings 42.17% and in multi-error misspellings 47.91% of substitution error rate is found. The reasons for the maximum substitution rates are discussed in the later sections. Table 3.1 is showing the detailed statistics of the various types of errors in single/multi-error misspellings.

Table 3.1 Percentages of various types of errors

Type of Error	SE	DE	IE	TE	ROE	SWE
Percentage in Single error misspellings	42.17	33.78	14.68	1.85	5.20	2.32
Percentage in Multi-error misspellings	47.91	32.84	17.0	1.43	0.60	0.22

While in Bangla[26] for the text containing 1,24,431 misspellings, percentages of substitution, deletion, insertion and transposition errors are 66.32, 21.88, 6.53 and 5.27 respectively. Thus the substitution and transposition error rates are high in Bangla as compared to Punjabi, while deletion and insertion error rates are low.

Comparison Of Various Types of Errors in Single/Multi-Error Misspellings

The order of error for various types of errors in single error misspellings is SE> DE> IE> ROE > SWE> TE while in multi-error misspellings is SE> DE> IE> TE> ROE> SWE. Run-on error and split word error are found to be lesser in multi-error misspellings.

3.4.1.1 Substitution error Analysis

This error occurs when at least one character is substituted by the other character. For example ausdw→vusdw, Swl→swl, suxdw→buxdw etc.

In the above three words, a→v, S→s, s→b are the various substitution character pairs respectively. Table 3.2 is showing the contribution of various substitution character combinations. It can be observed from Table 3.2 that the top 6 pairs contribute to more than 24% of the substitution errors.

The common reasons for substitution errors are:

1) Naveen Group elements: It is seen that 9.91% of the substitution errors are due to the naveen group elements. For example, zyl→jyl, Sihd→sihd.
2) Due to assignment to same keys (shifted and unshifted modes) on the keyboard, for example, n→l, a→v, x→d .
3) Words that are usually used in various forms, for example jyhw→ijhw, kRm→krm.
4) Vowels having similar sounds, for example y→ Y, o→ O, u→U, i→I.
5) Due to substitution of half characters, for example r→ ®, v→Í, h→H.

Table 3.2 Most common wrongly typed character pairs

Sr.no.	Wrongly Typed character pair	Percentage out of total no. of substitution errors	Cumulative Percentage	Percentage out of total no. of errors
1	S→s	6.13	6.13	2.65
2	&→P	4.51	10.64	1.95
3	^→K	3.85	14.49	1.66
4	x→d	3.54	18.03	1.53
5	z→j	3.09	21.12	1.34
6	n→x	2.95	24.07	1.27
7	□→=	2.81	26.88	1.21
8	r→ R	2.54	29.42	1.10
9	○○ → ○	2.38	31.80	1.03
10	n →l	2.06	33.86	0.89
11	Z→g	1.82	35.68	0.79
12	Yy→ Y	1.64	37.32	0.71
13	a→ v	1.34	38.66	0.58
14	L→l	1.28	39.94	0.55
15	e →h	1.06	41.00	0.45
16	q →d	0.91	41.91	0.39
17	‘ →°	0.72	42.63	0.31
18	v→ Í	0.43	43.06	0.18

Note: "→" is showing the bi-directional confusion

3.4.1.2 Deletion error Analysis

Deletion error: When at least one character is deleted in the desired word. For example g‘l→gl, c‘l→cl. These errors also give rise to real word errors, for example Pu‘l→P‘l, pwxI→pwx.

In the above example P‘l, pwx are two valid words but they are not the desired word. It is observed that deletion related errors contribute significantly after substitution errors. It is seen that the characters ‘ , N characters are most commonly missing characters. The percentage of missing ‘ is 20.51% and the percentage of missing N is 17.47% and these two characters alone contribute to 38% of deletion errors. Table 3.3 is showing the percentages of most commonly missing characters.

Table 3.3 Commonly missing characters

Sr.no.	Character	Percentage out of total no. of deletion errors	Cumulative Percentage	Percentage of the total no. of errors
1	‘	20.51	20.51	6.89
2	N	17.47	37.98	5.87
3	W	5.54	43.52	1.86
4	Uu	4.07	47.59	1.37
5	—	3.72	51.31	1.25
6	H	3.20	54.51	1.43
7	°	3.13	57.64	1.05
8	I	2.96	60.60	0.99
9	I	2.30	62.90	0.77
10	®	0.32	63.22	0.10
11	Í	0.03	63.25	0.04

3.4.1.3 Insertion Error Analysis

Insertion error: When at least one extra character is inserted in the desired word. For example, zhr→zihr , here i is the extra inserted character. These errors also give rise to real word errors, for example Xogqw→Xoigqw, swrw→swr.

In the above example Xoigqw,swr are two valid words but they are not the desired words. In the multiple form words confusion regarding insertion errors are due to:

1) The use of '' ,for example isiKAw→is'iKAw words on both side are delivering the same meaning.
2) The use of i ,for example swihq→swih'q words on both side are delivering the same meaning.

It is seen that the characters N , ' characters are mostly extra inserted characters. The percentage of insertion N is 17.53 % and the percentage of ' is 12.52% and these two characters contribute around 30% of Insertion errors (see Table 3.4).

Table 3.4 Most common insertion errors

Sr.no.	Character	Percentage out of total no. of insertion errors	Cumulative Percentage	Percentage of the total no. of errors
1	N	17.53	17.53	2.65
2	'	12.52	30.05	1.89
3	W	7.33	37.38	1.10
4	u	4.14	41.52	0.63
5	°	3.52	45.04	0.53
6	I	2.44	47.48	0.37
7	H	2.09	49.57	0.31
8	I	1.88	51.45	0.28
9	–	1.49	52.94	0.22

3.4.1.4 Transposition error analysis

Transposition error: This type of error occurs when two adjacent characters of the word are typed in swapped manner. For example, svyr→svry,rwq→rqw. In the above two words y→r,w→q are transposed character pairs.

It is found that these transpositions (like substitution) also give rise to real word errors. For example, krm→kmr,sUrq→sUqr where kmr,sUqr are two valid words. The percentage of transposition errors is 1.85% and 1.43% in single and multi-error misspellings respectively. No prominent transposition character pairs were found.

3.4.1.5 Run-on errors

Run-on Error[26]: This type of error occurs when two or more valid words are mistakenly written side by side without a space in between. For example, ijs dw→ ijsdw,dwdI mW→dwdImW .

In the above two word substitutions ijs,dw,dwdI,mW are four different words. Sometimes these errors give rise to real word errors. For example, aus dy→ausdy, ijs dy→ijsdy. Words ausdy,ijsdy are two valid words. The percentage of run on error is found to be 5.20% in single and 0.61% in multi-error misspellings.

3.4.1.6 Split word errors

Split Word Error[26]: This is opposite of Run-on error when there is some extra space is inserted between parts of a word. The error can be removed by removing the extra space.For example, skUl→s kUl, dIvwr→dI vwr etc.
Sometimes these errors give rise to more than one real word errors, for example ausdy→aus dy, ijsdy→ijs dy. Words aus,dy,ijs,dy are four valid words. The percentage of split word error is found to be 2.32% in single and 0.20% in multi-error misspellings.

3.4.2 Positional Analysis

The mistake position also plays an important and significant factor in the error pattern study. This can lead us to error zone of high probability. It is analyzed that pattern for the mistake position is almost similar in both single/multi-error misspellings. The maximum of the mistakes occur at the third position. The positional error zone decreases after 3rd position.

Table 3.5 Position wise distribution of misspellings

Sr. no.	Position	Percentage in single errors	Percentage in multiple errors
1	1st	13.11	13.0
2	2nd	18.98	16.25
3	3rd	26.80	23.16
4	4th	17.95	16.87
5	5th	11.30	13.20
6	6th	5.43	6.52
7	7th	3.78	4.50
8	>7th	2.65	6.50

It is generally believed that few errors tend to occur in the first letter of a word . The percentage of first position errors in Punjabi language is considerable. It is observed that in single error misspellings 13.10% and 13.0% in multi error misspellings are found to be first position errors. This rate is more than as expected. Concluded reasons are:

1) Naveen group Elements: Out of the total first position misspellings, 32.91% were the misspellings who have mistakes due to (S,^,Z,z,&,L) ,i.e. where the typist has typed S→s,^→K,Z→g,z→j,&→P,L→l. It means at least 32.91% of the first position misspellings are due to substitution errors. Though there are many more other substitution pairs that are also found. It is clearly signifying the probability of the

substitution errors at the first position. Table 3.6 is showing the distribution of errors evolving due to each element of the group.

Table 3.6 Commonly occurring character pairs at first Position

Sr.no.	Character Pair	% age out of total no. of first position misspellings	Cumulative Percentage
1	S→s	13.26	13.26
2	^→K	7.80	21.06
3	&→P	5.98	27.04
4	a→ v	4.43	31.47
5	z→j	2.99	34.46
6	Z→g	2.88	37.34
7	n→l	1.89	38.23
8	L→l	0	38.23

Note: "→" is showing bi-directional substitution

2) Shifted and Unshifted modes of typing: for example, n→l, a→v.

3) Multiple forms for a word: for example, jyhw→ijhw, vIcwr→ivcwr. The percentage of Substitution of the above word pairs out of the total no. of first position error misspellings is 3.15%.

3.4.3 Word length Effect

In English **Kukich[25]** analyzed over 2000 error types in a corpus of TDIL conversations and found that over 63% of the errors occurred in words of length 2,3,4 characters. According to our results the maximum of the misspellings have word length of five. It is observed that about 56% of errors are in words of length 3,4,5 (Table 3.7), and the words having word length of five contain maximum of errors.

Table 3.7 Word Length wise distribution of misspellings

Sr.no.	Word length	Percentage of errors	Cumulative Percentage
1	1	0.1	0.1
2	2	5.15	5.25
3	3	16.75	22
4	4	20.64	42.64
5	5	21.18	63.82
6	6	16.13	79.95
7	7	8.93	88.88
8	$>7^{th}$	11.12	100

Table 3.8 is showing the percentages of various types of errors in various word length zones. It is seen that about 63% of the errors (SE, DE, IE, TE, SWE, ROE) occur in word length 2,3,4,5.Out of total 21.30% of four character misspellings, 11.54% errors are due to substitution errors and similarly out of total 20.33% of five character misspellings, 8.87% errors are due to substitution errors. In the misspellings of word length 2,3,4,5,6 about 36% of errors are substitution errors.

Table 3.8 Distribution of misspellings according to word length and Type of error

Word Length / Type Of Error	1	2	3	4	5	6	7	>7	Total	Cumulative Percentage
SE	.01	2.60	7.07	11.54	8.87	6.87	2.82	3.42	43.20	43.20
DE	.02	0.40	4.60	6.56	7.77	5.43	4.04	4.79	33.61	76.81
IE	.07	2.09	3.17	2.67	2.51	2.24	0.88	1.46	15.09	91.90
TE		0.19	0.18	0.41	0.44	0.30	0.13	0.13	1.78	93.68
ROE			0.06	0.24	0.58	1.24	0.76	1.49	4.37	98.05
SWE		0.04	1.15	0.28	0.16	0.11	0.10	0.11	1.95	100.00
Total	0.10	5.32	16.23	21.70	20.33	16.19	8.73	11.40	100	
Cumulative Percentage	0.10	5.42	21.65	43.35	63.68	79.87	88.6	100		

3.4.4 Multiple Error Distribution

An analysis was also carried out for multi-error misspellings and it is observed that majority of the multi-error misspellings contain two mistakes (see Table 3.9).

Table 3.9 Showing the percentages of no. of mistakes in multi-error words

Sr.no.	No. of mistakes	Percentage out of total no. of multi-error misspellings	Cumulative Percentage
1	2	81.79	81.79
2	3	11.67	93.46
3	4	5.81	99.27
4	5	0.67	99.94
5	6	0	99.94
6	7	0	99.94
7	>7	0.06	100.00

3.4.5 Phonetically Similar Character Error Analysis

Phonetic errors are a special class of cognitive errors in which the writer substitutes a phonetically correct but orthographically incorrect sequence of letters for the intended word. Punjabi language also contains these type of confusion characters where the typist generally type the phonetically similar but wrong character.We have classified the phonetic errors into four categories :

1) Type 1 g→G, j→J, d→D, f→F, n→x, b→B
2) Type 2 S→s, ^→K, Z→g, z→j, &→P, L→l
3) Type 3 yy → Y , u → U , o→ O, i → I , ‘→ °
4) Type 4 r→ ®, v→ Í, h→ H

It is analysed that 17% of the errors are due to phonetically similar substitution pairs in above four categories .Out of the total no. of phonetic errors 59.28% are due to the Type 2 group elements.

Percentage of phonetically similar sounding vowel pairs is also considerable. It is concluded that about 23.83% of the misspellings contains mistakes due to Type 3 vowel pairs and 8.09% of the misspellings are due to Type 4 phonetically similar pairs.

3.4.6 Other Findings

3.4.6.1 *Addak* (‘) and *Bindi* (N)

These characters play an important role in insertion and deletion errors. It is estimated that 37.98% of deletion errors and 30.05% of insertion errors are due to these characters (see Table 3.10). For example, gl→g‘l,cl→c‘l,iv‘c→ivc, ibMdI→ibdI etc.

Table 3.10 Percentages of *Addak* and *Bindi* in Insertion and deletion errors

Character	Percentage out of total no. of Deletion Errors	Percentage out of total no. of Insertion errors
‘	20.51	12.52
N	17.47	17.53

3.4.6.2 Hyphen (-)

In Punjabi there are words containing where half the word came before and half the word is written after this “-” (hyphen) sign. For example, BlIN-BWq,Fih-FyrI,ieDr-auDr,by-rihmI etc.

It is found that about 0.25% of misspellings are due to the insertion and 1.39% of misspellings are due to deletion of the hyphen. A list is proposed for these categories of words that contains hyphen as part of them so that spellchecker give suggestions regarding the absence of "-".

3.4.6.3 Distorted words

Sometimes errors are also found in the condition when half or a part of the word is at the end of the line and remaining part is at the beginning of the next line. For example,

auh skUl jw irhw sI ik ausny dyiKAw ik ausdy bsqy ivc A'j vI pMjwbI qy AMMgRyzI dI **ik**

qwb nhI sI [

The above line is showing the exact problem.

3.4.6.4 Most commonly misspelled words

We have collected a list of the most commonly substituted word pairs and concluded the common reasons of misspellings occurring in them. For example,

Awpxw→Awpdw, g'l→gl zihr→jihr, jyhw→ijhw etc .

The common reasons are:

1) Deletion or insertion of *addak*, *bindi* and *tippi*.
2) Due to mostly substituted consonant-consonant and vowel-vowel pairs.
3) Due to substitution of S→s, ^→K, z→j, &→P, Z→g, L→l.
4) Multiple forms of the same word.

3.4.6.5 Distribution of Multi-error misspelling by ignoring errors involving similar sounding characters.

By ignoring the errors involving *addak*, *bindi* S→s,^→K,z→j,&→P,Z→g,L→l and similar sounding characters we found that 24.63% of the multi-error misspellings contain <=1 error and 7.15% were the misspellings who have >=2 errors and the remaining 68.22% do not have errors due to above reasons. This information will be useful in designing suggestion list for multi-error words. If one takes care of errors involving these characters then as already observed in 68% of cases the errors can be removed easily in multi–error misspelled words. It is showing the effect of the above errors in multi-error misspellings.

3.4.6.6 Error Rates of Naveen group elements

As I have discussed earlier that naveen group elements plays an important role in error generation so on a detailed analysis of the error rates I found that error rates are in the order of Z→g >&→P >^→K >L→l >z→j >S→s .

Table 3.11 Error rates of naveen group elements

Sr.no.	Substitution pair	Percentage
1	Z→g	44.30
2	&→P	19.31
3	^→K	15.59
4	L→l	12.75
5	z→j	4.95
6	S→s	4.10

Note: "→" is showing bi-directional confusion

3.5 Suggestion List for Punjabi Spellchecker

Based on the above findings we have designed a generator for suggestion list for wrongly spelled Punjabi words. The features we have looked in the suggestion list are:

1) It should be small, say between 5 to 10 words, so that the complete list is visible without the need to scroll it.
2) The correct word should appear at the top of the list or in first 2-3 positions, so that the user need not scan the complete list.
3) The correct word should be present in the suggestion list.

Presently the suggestion list is for single error only. In future it will be enhanced for multiple error word. From the error analysis, it was gathered that the majority of single errors are due to substitution errors, followed by deletion, insertion and transposition errors. Thus the words in the suggestion list are sorted in this following order:

1) Substitution error related words
2) Deletion error related words
3) Insertion error related words
4) Transposition error related words

Thus, for example for the word `sJm` the suggestion list is generated in following order `som sem sm smJ`. The first two words correspond to substitution error, while the next two words correspond to the insertion and swapping errors respectively.

Here again as it was noted that the two characters *addak* and *bindi* contribute to substantial number of insertion and deletion errors. So in case of the suggestion list, words corresponding to these errors were generated, they were placed higher than other words in the same category. Thus, for example for the word `jada`, if the suggestion list is generated in following order corresponding to substitution and deletion errors:

jala jadU jaNa jaNda ja~da and jagda .

Then as the fifth word is a deletion related error corresponding to bindi character, so it will be placed above other words and will be put in the first place.

Similarly for the word s‘B, if the suggestion list is generated in following order corresponding to substitution, insertion and deletion errors: s‘c s‘s s‘p sB and s‘Be .Then as the fourth word is an insertion related error corresponding to *addak* character, so it will be placed above other words and will be put in the first place. The suggestion list will be displayed as sB s‘c, s‘s s‘p and s‘Be .

As already observed, nearly 10% of substitution errors are related to the characters belonging to naveen group (S,^,Z,z,&,L) and their counterparts (s,K,g,j,P,l). So if in the suggestion list, we have a word generated by substitution, which is differing from the misspelled word by a character from naveen group or its counterpart, then that word is placed higher in the suggestion list. Thus if, for example if a misspelled word is Sma and the suggestion list generates the following list of words sorted according to the error type.

dma jma sma Smal Srma ma Sam

The first three words correspond to substitution error; the next two words correspond to deletion error, while the last two words correspond to insertion and transposition error respectively.

Then in the list corresponding to substitution related error, we have the word sma differing from the misspelled word by S, which belongs to naveen group and so it will be placed higher than other words in the list. So, the final ordered suggestion list is:

sma dma jma Smal Srma ma Sam .

Based on the error analysis of Punjabi, we have been able to design a powerful suggestion list for Punjabi spellchecker, in which the misspelled word is usually present

in top 2-3 positions of the list. The only limitation of the suggestion list is that presently it works for single error words only.

CHAPTER 4

CONCLUSION/FUTURE SCOPE

4.1 Conclusion

In this thesis, I have discussed about pattern of typing errors in Punjabi text. For this purpose an analysis was made on about 20000 wrongly typed words. These words were collected from Type Colleges, Professional typists and Government institutions and private printing presses and every document was carefully checked and the misspelled words were manually collected and analyzed.

This is the first time that a detailed study has been made on the pattern of Punjabi typing errors. I have done analysis based on type of errors, positional effects, first position error analysis, phonetic effects, word length effects etc. Besides the usual typing mistakes, the other reasons for majority of the misspellings in Punjabi language are due to:

1) Multiple forms of the same word or the non-standardization of Punjabi spellings.
2) Slight difference between the pronunciation and spellings of some of the Punjabi words.
3) Naveen group elements.
4) Phonetic similarities of various consonants and vowels.
5) Borrowed words from other languages
6) Unnecessary Insertion, deletion of *addak*, *bindi* and *tippi*.

We have used the information gathered from error analysis to design a single error based suggestion list for a Punjabi spell checker.

4.2 Future Scope

This work can be enhanced for handwritten text pattern analysis and OCR generated error analysis. Analysis can also be extended to more number of misspellings in Punjabi text and on different types of data materials. Work can also be extended to context dependent error correction research that require information from the surrounding context for both detection and correction.

REFERENCES

[1] F.J. Damerau (1964),"A technique for computer detection and correction of spelling errors". Commun. ACM, pp. 171-176.

[2] V.I. Levenshtein (1966),"Binary codes capable of correcting deletions, insertions and reversals". Sov. Phys. Dokl. , pp. 707-710.

[3] Text Processing of National Languages
http://www.cicc.or.jp/english/hyoujyunka/af08/8-05.html

[4] Thomas N. Turba (1981),"Checking for spelling and typographical errors in computer based text", Proceedings of the ACM SIGPLAN/SIGOA symposium on text manipulation, pp. 51-60.

[5] Walker, D. E., and Amsler, R.A. (1986), "The use of machine-readable dictionaries in Sublanguage analysis", In analyzing Language ~n restricted domains: sublanguage Description and Processing, pp. 69-83.

[6] Karen Kukich (1992), "Spelling corrections for telecommunications network for the deaf ", Communications of the ACM, pp. 80-90.

[7] Roger Mitton (1987), "Spelling checkers, spelling correctors and the misspellings of poor spellers", Information Processing and Management: an International Journal, pp. 495-505.

[8] Jones, M. A., Story, G. A., and Ballard, B. W. (1991),"Integrating multiple knowledge sources in a Bayesian OCR post-processor", In Proceedings of IDCAR-91, pp. 925-933.

[9] Damerau, Fred J and Mays, Eric(1989),"An examination of undetected typing errors", Information Processing and Management, pp. 659-664.

[10] Pollock, J. J., and Zamora(1984),"Automatic Spelling Correction in Scientific and scholary text ",Communications of the ACM, pp. 358-368.

[11] Rhyne, J. R., and Wolf, C. G.(1993),"Recognition-based user interfaces",In Advances in Human-Computer Interaction, pp. 191-250.

[12] Zipf, G. K. (1935) ," The Psychobiology of Language". Houghton Mifflin, Boston.

[13] Landauer, T. K, and Streeter~ L. A. (1973),"Structural differences between common and rare words", pp. 119-131.

[14] Pollock, J. J., and Zamora, A. (1983), " Collection and characterization of spelling errors in scientific and scholarly text", J. Amer. Soc. Inf. Sci., pp. 51-58.

[15] Kukich, K. (1990), " A comparison of some novel and traditional lexical distance metrics for spelling correction", In Proceedings of INNC- 90-Paris, pp. 309-313.

[16] Yannakoudakis, E. J., and Fawthrop, D. (1983),"An intelligent spelling corrector", Inf. Process. Manage., pp. 101-108.

[17] Grudin, J. (1983), "Error patterns in skilled and novice transcription typing", In Cognitive Aspects of Skilled Typewriting, pp. 121-143.

[18] Church, K. W., and Gale, W.A. (1991)," Probability scoring for spelling correction", Stat. Comput. , pp. 93-103.

[19] Wing, A. M., and Baddeley, A.D. (1980), "Spelling errors in handwriting: A corpus and distributional analysis", In Cognitive Processes in Spelling, U. Frith, Ed. Academic Press, pp. 251-285.

[20] Van Berkel, B., and Desmedt, (1988) ," Triphone analysis' A combined method for the correction of orthographical and typographical errors", In Proceedings of the 2nd Applied Natural Language Processing Conference.Association for Computational Linguistics (ACL) pp. 34-38.

[21] Robert A. Wagner (1974), "Order n correction for regular languages", Communications of the ACM, pp. 265-268.

[22] Knuth, Donald E.(1973), "The Art of Computer Programming: Volume 3 Sorting and Searching" , Addison-Wesley, pp. 107-112.

[23] Davidson, Leon (1962), "Retrieval of misspelled names in an airlines passenger record system", Communications of the A.C.M, pp. 169-171.

[24] E.M. Riseman and A. R. Hanson, "A Contextual Post Processing System for Error Correction using binary n-grams", IEEE Transactions on Computer, pp. 480-493.

[25] K. Kukich (1992) ,"Techniques for automatically correcting words in text",ACM Computing Surveys, pp. 377-439.

[26] P. Kundu and B.B. Chaudhuri (1999), "Error Pattern in Bangla Text",International Journal of Dravidian Linguistics, pp. 49-88.

[27] Morris, Robert & Cherry, Lorinda L (1975)," Computer detection of typographical errors", IEEE Trans Professional Communication, pp. 54-64.

[28] Wagner, Robert A. & Fischer, Michael J (1974), "The string-to-string correction problem", Journal of the A.C.M., pp.168-173.

[29] R.E. Gorin (1971), "SPELL: A spelling checking and correction program", Online documentation for the DEC-10 computer.

[30] Durham, I, Lamb, D.A, & Saxe, J.B (1983), "Spelling correction in user interfaces", Communications of the A.C.M., pp.764-773.

[31] M.D. Kernighan, K.W. Church, and W.A. Gale. (1990),"A spelling correction program based on a noisy channel model", In Proceedings of the Thirteenth International Conference on Computational Linguistics, pp. 205-210.

[32] Gale and Church, (1991)," A program for aligning sentences in bilingual corpora" ,In Proceedings of the 29th Meeting of the ACL, pp. 177-184.