KASHMIRI SPELL CHECKER AND SUGGESTION SYSTEM

Aadil Ahmad Lawaye*
Bipul Syam Purkayastha**

Abstract

The aim of this research paper is to present the complete design and implementation of a Kashmir Spell Checker. Although all major word processors offer spell checking for English and in various European languages including many Indian languages like Hindi, Urdu, Sanskrit, Tamil and Kannada, but no work has been done for Kashmiri language.

Key words: Kashmiri Language, English, Script, Challenges.

Introduction

Spelling checkers 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. According to O'Neill, et. al., 2003, "spelling checkers have looked for four possible errors: a wrong letter ("wird"), an inserted letter ("woprd"), an omitted letter ("wrd"), or a pair of adjacent transposed letters ("wrod")". This process can be resolve by means of a simple dictionary lookup. However, the notion of having languages with high degree of inflection requires additional computational work such as morphological analysis and stemming. Developing a Spelling checker for Kashmiri poses many new challenges, which complicates the design of the spell checker. Kashmiri language is far different from Western languages in phonetic properties and grammatical rules. Syntactically, from a word order perspective, Kashmiri shows both verb medial and verb final characteristics. Kashmiri also shows strong V2 features like Germanic, Yiddish, Dutch and Icelandic. In the root clause, the finite verb may be preceded not only by the subject, as in English, but also by other clause constituents, as is the case in the verb-second languages. 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 Kashmiri; rather it needs different algorithms and techniques for expected efficiency.

Brief Description about Kashmiri Language

Kashmiri language is primarily spoken in the Kashmir province and some parts of the Jammu province of the state of Jammu and Kashmir State and by migrant populations in the rest of India and abroad. The earliest script used for writing Kashmiri is the Sharda script which is now only used by some Kashmiri pundits for writing horoscopes. Presently, the official script of Kashmiri is the modified Persio-Arabic script with additional diacritic marks to represent Kashmiri specific sounds. Alternative scripts like the modified Devanagri and Roman script are also used for writing Kashmiri. Regarding the modified Persio-Arabic script, it is written from right to left. It has two modes: *nasakh* or the type script, and *nastalikh*, the handwritten version.

Spelling Errors

Detecting whether or not a word is correct seems simple-why not to look up the word in a set of all words? Unfortunately, there are some problems with this simple strategy. Firstly, a lexicon containing

*Department of Computer Science, Assam University, Silchar.

** Department of Computer Science, Assam University, Silchar.

all correct words could be extremely large, which entails space and time inefficiency. Secondly, in some languages it is practically impossible to list all correct words, because they are highly productive. Thirdly, making a spelling error can sometimes result in a real word, which belongs to the lexicon—such an error is called a *real-word error*. It is impossible to decide that this word is wrong without some contextual information. Fourthly, the bigger the lexicon, the more esoteric words it contains, making real-word errors more likely. The appropriate lexicon size is dependent on the language. In only slightly inflective languages as English, lexicons of size 50,000 – 2,00,000 words were recommended (Damerau, 1990; Damerau and Mays, 1989; Peterson, 1986). For highly inflective languages, the lexicon has to be much larger, and typically contains millions of words.

The classic data structure offering a fast search is a hash table (Knuth, 1973). Its disadvantage is the need to properly choose the hash function and the size of the hash table to mitigate the problem of collisions. Minimal perfect hashing (Czech *et al.*, 1997) eliminates collisions but requires storing the hash table of a size equal to the number of words in the lexicon and the whole lexicon (possibly compressed by some method). Another popular data structure used for lexicon storage is a trie (Knuth, 1973). It is a character-oriented tree, in which every path from a root to a leaf corresponds to a key, and branching is based on successive characters. A trie offers fast lookup and some compression of the lexicon. Its size, however, is typically comparable to the lexicon size due to the need of storing pointers to the nodes. There are many works on reducing trie sizes; some of the alternative versions of tries are the C-trie (Maly, 1976), PATRICIA (Morrison, 1968), and Bonsai (Darragh *et al.*,1993).

Kashmiri Spell Checker Architecture

There are two different applications in the system. One creates the dictionary, which is already been developed and executed once to create the dictionary. Second is designed for Spell Checking of Kashmiri Text where user gives the input Kashmir text and the system detects the errors by looking up for that word in dictionary and provide the suggestions for those errors. Then user can select the suggestion from the list and make changes accordingly and the final output is the corrected text without errors. Figure 1, shows the architecture of the system. The system is divided into three modules. Creation of Kashmri Dictionary (Lexicon), Error Detection and Error correction & Replacement. Various techniques and methods are used to develop these modules.

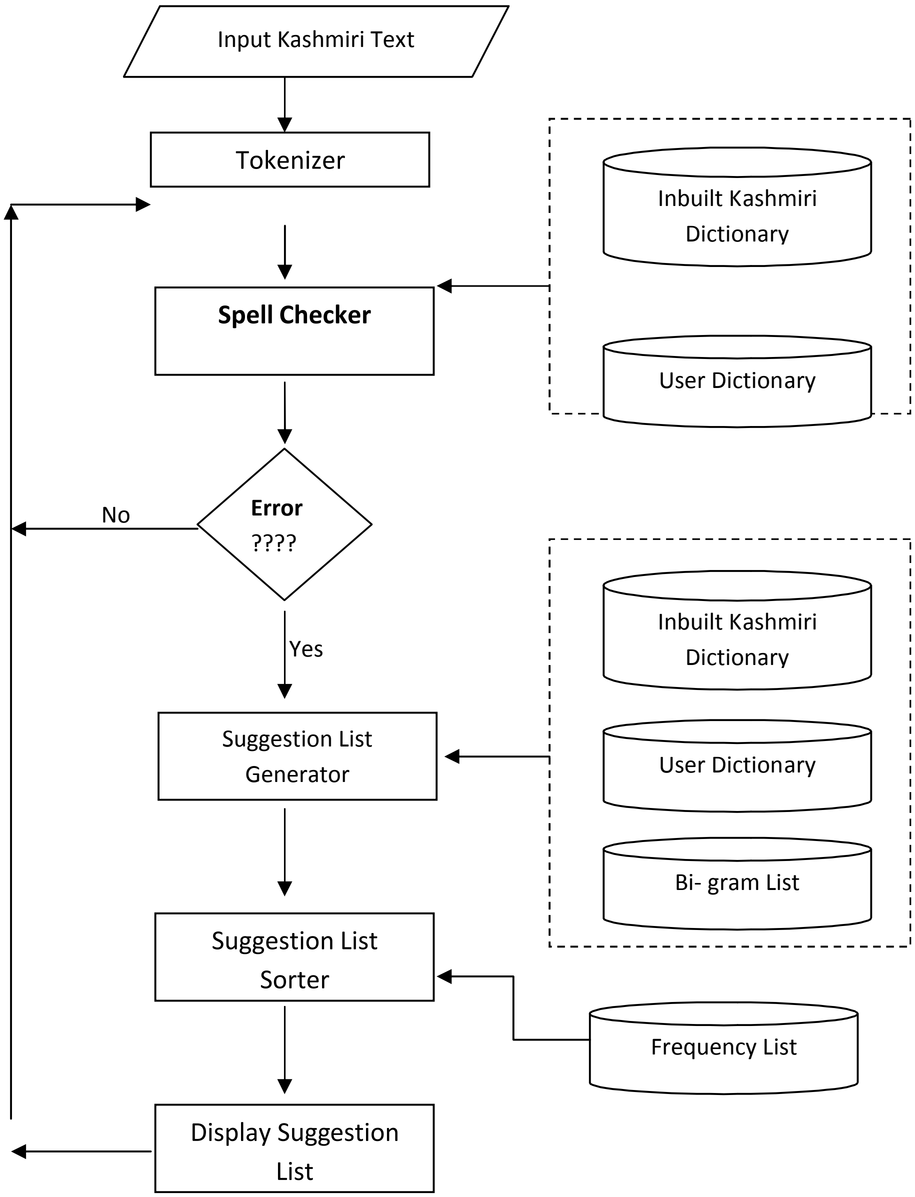

Fig. 1: Architecture of Kashmiri Spell Checker

Dictionary Creation/ Lexicon

This application is a part of Spell Checker but is not needed to be executed every time because a dictionary that acts as database for Spell Checker is created using this tool. A corpus of around one million words, which contains unique Kashmiri correct words, has been created using this application, which will act as Dictionary for the Spell Checker. An application is developed in java to create that dictionary where a large Kashmiri text is given as an input. The dictionary creation tool tokenizes each word and calculates unique hash code of each word. A Hash Set holds a set of words, but in a way that it allows you to easily and quickly determine whether an object is already in the set or not. It does so by internally managing an array and storing the object using an index which is calculated from the hash code of the word, thus resulting in a set of unique words.

Error Detection

It detects the errors present in the input text. There are many techniques available which can be used for error detection. The inbuilt Kashmiri dictionary has been partitioned into sixteen sub-dictionaries based on the word length and at execution time each of these sub-dictionaries 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. 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

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

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:

- Phonetic similarity of the suggested word with the related mis-spelt word
- Frequency of occurrence of the suggested word

The smallest number of substitutions, insertions and deletions required in that order to change the mis-spelt word to the suggested word.

Conclusion and Future Scope

The developed Spell Checker for Kashmir is a Standalone application which is not a part of any word processor. In this system, we have taken care of only non real word errors. The system detects approximately 80% of the errors and provides 85% of the correct suggestions. Real word error detection and correction is a subject of future research. This system can be used for other languages also but dictionary for other languages will have to be created for its use and we will implement it as add on for open office.

References

Golding, R. (1995). A Bayesian hybrid method for context- sensitive spelling correction. *Proceedings of the Workshop on Very Large Corpora*, pp. 39-53.

Czech Z.J., Havas G. and Majewski B.H. (1997). Perfect hashing. *Theoret. Comput. Sci*., Vol. 182, No. 1–2, pp. 1–143.

Brill, E. & Moore R. (2000). An improved error model for noisy channel spelling correction," *Proceedings of the ACL 2000*, pp. 286-293.

Riseman, E.M. & Hanson, A. R. (1994). A Contextual Post Processing System for Error Correction using binary n-grams", IEEE Transactions on Computer, pp. 480-493.

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

Gurpreet Singh Lehal (2007). Design and implementation of Punjabi spell checker. *International journal of systemic cybernetics and informatics*, pp.70-75.

Knuth, D.E. (1973). *The Art of Computer Programming.* MA: Addison–Wesley.

Maly K. (1976). Compressed tries. *Comm. ACM,* Vol. 19, No. 7, pp. 409–415.

Morrison D.R. (1968). PATRICIA—Practical Algorithm to Retrieve Information Coded in Alphanumeric. *J. ACM,* Vol. 15, No. 4, pp. 514–534.

O'Neill, M.E. & Connelly, C.M. (2003). Spell Checking Using Hash Tables. http://www.cs.hmc.edu/courses/mostRecent/cs70/homework/cs70ass9.pdf

Tanveer Siddiqui, U.S.Tiwary, (2008). *Natural Language Processing and Information Retrieval.* Oxford University Press.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl*, Pp. 707-710.