Hindi Spell Checker

Amit Sharma, Pulkit Jain
Instructor : Dr. Amitabha Mukerjee
{asharma,pulkitj,amit}@cse.iitk.ac.in

April 17, 2013

Abstract

Spell checker applications are important part of fundamental applications such as editors or search engines. They are important for flow of correct information in form of text. Hindi is one of the commonly typed languages. A lot of text in the form of newspapers, books, novels, magazines, web pages and other documents is typed in Hindi. We, in this project have made an attempt towards building a spell checker application for Hindi.

Introduction

The task of spell checking is primarily divided into two parts :

(i) Error Detection
(ii) Error Correction

The first part consists of identifying the errors in the typed text. This part uses a language model which accounts for the words allowed in the language. Language models may vary from a simple list of permitted words to finite state graphs that accept words with valid spellings in the language.

The second part consist of rectifying the spelling mistakes made by the user. This requires an error model which tries to find out the candidate replacements of a mis-spelled word. This part also include ranking of the candidate replacements. Ranking may be done on the basis of edit distances, string similarity measures, phonetic measures or word frequency.

Error Classes

Errors in spell checking literature are broadly classified into two categories. These are :

1. Non Word Errors

Those spelling mistakes that arise due to the word not complying with the language model are categorized as non word errors. These are those words that are considered as mis-spellings of some other word in the language.

For example:	बस्तु	for	वस्तु
	ग्यान	for	ज्ञान

These mistakes generally arise due to a wrong key press or lack of knowledge of spelling of the correct word.

2. Real Word Errors

This class of errors include those errors where the mis-spelled word fits into the language model but, occurs as a mis-spelling of some other correct word. In other words, the word does not fit into the context of the sentence.

For Example : दुकान उस और है

for

दुकान उस ओर है

कलम पानी में उगता है

for

कमल पानी में उगता है

Both the words और, कलम are correct words in Hindi language, but occur as mis-spellings of ओर and कमल respectively in above sentences.

Previous Work

A lot of research has been done in the field of spell checking. By and large, non word errors have been dealt with by the use of simple dictionaries as language models. In [1], the authors make use of finite state graphs for identifying and correcting non word errors for languages where the morphology of language permits potentially infinite words in the dictionary, and they claim their method to work for real word errors also. In [2], the authors deal with spelling errors in tagging by constructing co-occurrence graphs, where two tags are connected by an edge, if they have some non-zero probability of occurring together in the language. [3] presents the use of 3-grams to use contextual information to deal with real word errors. We have primarily followed the work done in [3].

Our Implementation

We use a dictionary with word, frequency pairs as our language model. A lookup into the dictionary categorises a word as correct or erroneous. To produce candidate corrections, we calculate strings at edit distance one and two from the identified erroneous string and further filter out those strings that are not present in the dictionary.The edit distance used is Damerau-Levenshtein edit distance. This gives us a set of words that are possible corrections for the erroneous word. To produce a ranking among these words, we sort these candidates in increasing order of edit distance. Words at same edit distance are sorted in order of their frequencies.

To deal with real word errors we create 2-grams and 3-grams along with their frequencies of occurence. To check for real word errors, every 2-gram. 3-gram and 4-gram of the sentence is checked in the created set. If the frequency of the gram is low, it is raised as an error. To produce corrections for the erroneous gram, we calculate edits of each of the word in the gram and construct valid candidate grams from these. Again ranking is done on the basis of edit distance and frequency of the grams.

The corpus that we use is made available as “Hindi Corpus (tar)” at
http://www.cfilt.iitb.ac.in/Downloads.html

We have also used the code made available at
http://norvig.com/spell-correct.html

Clearly, our methodology is highly dependent upon the corpus that we use. There were nearly 30M words in the corpus with around 1.17 Lac unique words. The corpus is noisy i.e. it contains mis-spelled words. So we try to eliminate noise by not considering with words with low frequencies. We present few numbers from this corpus.

	#Grams (freq > 2)	#Grams (freq > 5)	#Grams (freq > 10)	#Grams (freq > 20)
1-Grams	37181	22226	14437	4755
2-Grams	151899	63504	31282	14692
3-Grams	107995	29029	10619	3756
4-Grams	42245	6629	1851	489

We have also implemented a basic GUI in which users can type in Hindi and check for non word and real word errors. All our code and implementation has been done in python.

Results

We collected a set of 291 mis-spelled words along with their intended correct words. Following were the results produced when we used various sets of Grams (from the above table). Let d_i denote the unigrams with frequency > i . Following are the statistics we obtained.

	Mis-classified as correct	Intended word in top10 suggestions	Intended word not in dictionary	Detection rate	Correction rate
d_2	54	199	21	81.4%	68.3%
d_5	35	201	37	87.9%	69.1%
d_{10}	28	188	56	90.3%	64.6%
d_{20}	14	173	85	95.2%	59.4%

Since using d_5 gives the best correction rate and second best detection rate, we continued to use d_5 to test our application over a set of 15 articles collected from several online newspapers including dainik jagran and navbharat times. Following are the results obtained for non-word and real word errors detection are as follows.

Non Word Errors : 4086/19219 words were raised as errors. This is attributed to use of English words written in hindi multiple times and use of abbrevations such as 'बसपा सुप्रीमो'

The spell checker is also able to detect and correct real word errors also, although the results are not very good.

Conclusion and Future Work

In above results, we see that mis-classification reduces as we eliminate more words out of the dictionary, and the number of words that are corrected are reduced. This is bound to happen. The database is noisy. Errors in the corpus are too large to be fixed manually. When we try to eliminate wrong words to increase the detection rae, we also land up in eliminating many correct words resulting a decrease in correction rate. Further there are not many correct words in the corpus. From the above table, there are as many as 21 words that do no occur even twice in the corpus.

A better implementation of GUI and a better corpus and bigger corpus to use would be of great help in increasing the accuracy of the spell checker.

REFERENCES

[1] Tommi Pirinen and Krister Linden. *Finite-state spell-checking with weighted language and error models.* Proceedings of LREC 2010 Workshop on Creation and use of basic lexical resources for less-resourced languages [2010]

[2] Francesco Bonchi, Ophir Frieder, Franco Maria Nardini, Fabrizio Silvestri and Hossein Vahabi. *Interactive and Context-Aware Tag Spell Check and Correction* [2012]

[3] Suzan Verberne. *Context-sensitive spell checking based on word trigram probabilities* [2002]

[4] Neha Gupta, Pratistha Mathur. *Spell Checking Techniques In NLP: A Survey* [2012]

[5] Peter Norvig. How to write a spelling corrector. http://norvig.com/spell-correct.html