Syntactic Word Prediction for Hindi

Ms. Niti Shah[1], Mr. Nirav Khetra[2]
[1]Silver Oak College of Engineering and Technology, Ahmedabad, Gujarat
[2]Apollo Institute of Engineering and Technology, Anasan, Gujarat

Abstract-*Word prediction is very effective technique for improving efficiency of entering text. Current word prediction systems predict a word if and only if a user has not made mistake in the starting of some characters of the word. This is more applicable for Indian languages, which have a large set of characters, alphabets, words with complex characters and inflections, phonetically similar sets of characters, etc. Therefore, there is a requirement for development of better word prediction. For existing systems, N-Gram approach considers only sequence of words in given sentence. It doesn't consider structure and grammar of the sentence. New approach is to use "Syntactic N-Gram" approach. Sn-Grams are differing from traditional n-grams in the way of which elements are considered as the neighbors. Sn-Grams consider Grammar in making prediction. So they are less arbitrary in making predictions.*

Keywords-Hindi Word Prediction System; Hindi Keyboard; Indian Languages; Syntactic N-Grams; Sn-Grams.

I. INTRODUCTION

Natural Language Processing:

Natural language processing (NLP) is a study of excellence, a field of computer science concerned with the human – computer interactions.

When you want an intelligent system like machines or robots to perform as per your instructions, Natural language is required.

Word Prediction:

Many of us can have problems with correct spelling, or would not like to type more. Software that completes words by showing some words using predictive text on keyboards of mobile phones or from Webpages can help.

Word prediction is an "intelligent" feature of word processing that can reduce writing for a range of users by reducing the number of key pressing necessary for typing words. [3]

As the number of Internet users are increasing very quickly, the number of email users or the number of mobile internet users are also increasing. The increasing need leads to the fast typing system for the users to save their time. [8]

Further, there are some characters which look same, raise the possibility of getting confused when making choice of the correct character and it leads to pressing the wrong characters. [9]

It is very important to see that the virtual keyboard works properly so that the users can type their routine life words very quickly and accurately.

According to Internet World Stats 2015, numbers of Internet Users are growing very rapidly. There is a growth of 1,267.6% of Internet users in 2000 to 2015. [8] (See Figure 1)

Following chart shows the Internet Usage

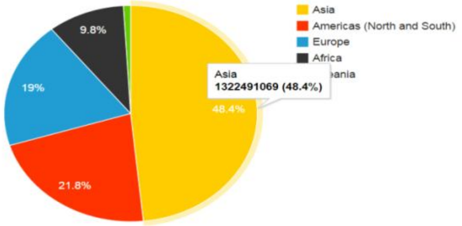

Figure 1 Internet Users in Asia[8]

The Hindi language follows Devanagari script, contains a very rich set of characters including:

- 11 vowels,
- 35 consonants [18]
- 12 matras
- Some special symbols and complex characters called "conjuncts". [9]

Following are the major objectives: [2] [3] [8]

- To give better word predictions for Indian Languages like Hindi
- To help the Indian people who want to use the Data available on Internet but cannot access it due to Lack of knowledge in English Language

- To help people who wants to communicate in Hindi Language but finds difficulties in typing due to large character sets in Hindi

- To give ease of access to Indian people to communicate in their National Language as Hindi is 4th most widely used language.

The uses of virtual keyboards with word prediction:

- Mobile Applications
- Desktop Applications
- Search engine feature
- Hardware Keyboard feature
- Software keyboard feature
- Microsoft Office extension
- G-mail plug-in, etc.

II. LITERATURE SURVEY

2.1 Detection and Correction of Non Word Spelling Errors in Hindi Language [1]

The research shows that more development in Hindi language will be useful for those people who have no or little understanding about other languages like English but they want to use resources which are available in the world.

Hindi is a highly complicating language, because the matras and many of the other symbols are present, which confuse the user in selecting the right word

.

For example, (**दिन – दीन**) (**सामान - समान**). These words are similar but they have discrete meaning. A very small change in word may lead to an error.

If incorrect word is chosen by the user and it is not identified and corrected by the spelling checker, then it can maximize the chance of irrelevant outcome.

For ex, **वस्तुकाज्ञान** and **वास्तुकाज्ञान** have discrete interpretation. Processing of text in Hindi is a very big issue. [1]

Table 1: Levels of Errors for Hindi

	Levels of Errors in Hindi	**Example**
Level-1	Non-word errors	वहजलवीतीहै।
Level-2	Syntactic errors	रामसीताबुलाए।
Level-3	Real-word errors	कलमपानीमैंउगताहै।
Level-4	Discourse errors	अलमारीमेंतीनपुस्तकेहैजिनकेरंगलालऔरनीलाहै।
Level-5	Pragmatic errors	मेराघरबायीओरहै।

In this paper, researchers suggested the approach to identify and correct the non-word spelling mistakes for Hindi Language.

It is examined that procedures followed by English error identification and correction could not be straight away used for Hindi.

2.2 Word Prediction System for Text Entry in Hindi [2]

In this paper, authors made a keyboard named "*hIndiA". The analysis presents that correction rate of *hIndiA is roughly 89.75% for typing errors on average compared to text entry without prediction using the same texts.

Authors have analyzed 43.05% Keystroke save rate, 92.46% hit rate and 93.84% utilization of Prediction with the word prediction system.

Table 2: Comparison of Features of Different Hindi Keyboards

Description	Lipik	Google	Trigram	*hIndiA
Layout (Alphabetic)	N	N	Y	Y
Layout (QWERTY)	Y	Y	N	N
Human typing error handling for Hindi	N	N	N	Y
Auditory clue support	N	N	Y	Y
Multiple word prediction	N	Y	N	N
Prediction window size	10	10	7	7

There is an absence of calculating errors while composing text through the word prediction system proposed by the authors. A new measure can be evolved to decide the performance for text composition with word prediction.

Swarachakra Keyboard for Indic Scripts (Tutorial) [3]

Authors have developed a Hindi Keyboard named “Swarachakra”. Swarachakra is made with an alphabetical design based on the structure of Indic scripts. Swarachakra displays the order of consonants according to the formation of Indic scripts, orally gathered and ordered in a grid like a text book.The following figure shows the layout of the Swarachakra keyboard for Hindi Language.

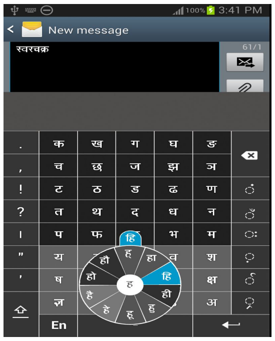

Figure 1 Swarachakra Hindi Keyboard

Syntactic Dependency-Based N-grams as Classification Features [4]

The authors said that techniques based on N-gram are leading in modern Natural Language Processing and its applications. N-grams are series of pieces as they are visible in texts. In most of the word prediction system till now, n-gram model is used for the prediction of the next word.

This paper introduces syntactic dependency based n-grams (sn-grams). Sn-grams are unique from normal n-grams by the idea of which elements are considered as the neighbors.

In sn-grams, syntactic relations in the syntactic trees are considered to take neighbors, and not the sequence of words. Any of the NLP task can use Sn-Grams in the place of normal n-grams.

The main benefit of sn-grams is that they are considered by the syntactic connections of words, thus, each and every word is linked to its “original” neighbors, ignoring the randomness that is found by the structure of sequence of text.For example, let’s take following two phrases. See following Figures.

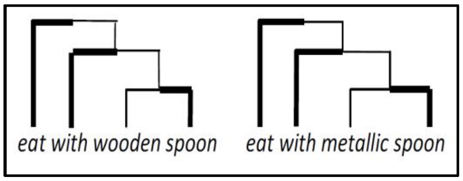

Figure 2 Representation of Syntactic Relations

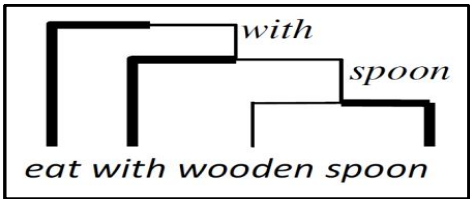

Figure 3 Promoted Head Nodes

If normal (traditional) bigrams are generated from above example, then only one bigram is common: “eat with”.

If Syntactic N-Grams (sn-grams) are taken into account, then two bigrams are caught common: “eat with” and “with spoon”.

Skip N-grams and Ranking Functions for Predicting Script Events [5]

The techniques examined and measured by the authors for generating models that, by giving a part of a chain of events, it predicts some other part of events that belong to the same script.

For script event prediction, authors used the technique of Skip-Gram.

They have used

- 1-skip gram bigrams and
- 2-skip gram bigrams, instead of the traditional bigrams.

Authors have presented skip-grams and shown that skip bigrams have scored steadily higher on event prediction. Use of Skip-gram instead of traditional n-gram is better proved in form of results.

One major issue with Skip-gram is that: Skip-gram does not provide Linguistic phenomenon.

III. HINDI WORD PREDICTION WITH SYNTACTIC N-GRAM – THE NEW APPROACH

The New approach to introduce Syntactic N-Gram for Hindi Word Prediction is as shown here. It states that when the user types the word, the word is checked with the parser.

The parser uses WordNet for reference and then it generates the predictions based on whether the word is available in the dictionary or not.

Following are the steps:

Step 1: Generate Hindi Keyboard for Android Devices
Step 2: Type the word from the Hindi Android Keyboard
Step 3: Check the input word with Parser with the use of WordNet
Step 4: Check whether the word is available or not in the WordNet
Step 5: Display the predicted words in Prediction Window.

Working of Proposed System:

Step 1: Generate Hindi Keyboard

Language	:	Java
OS	:	Android
Keyboard type	:	InScript (Unicode)
Keyboard Layout	:	Alphabetical Order

The New approach to introduce Syntactic N-Gram for Hindi Word Prediction.

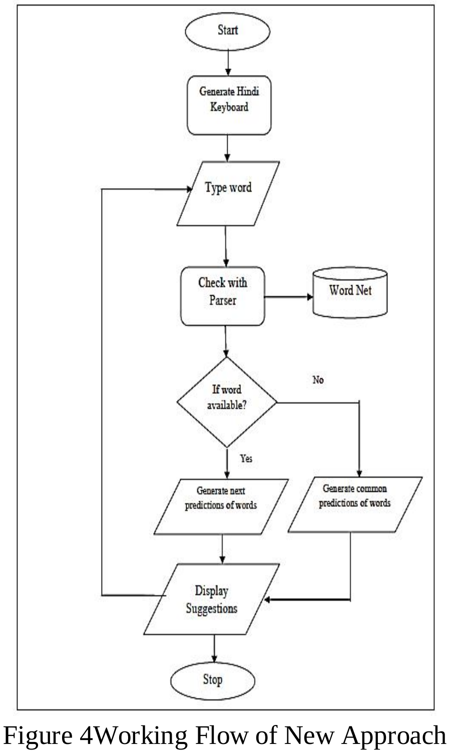

Figure 4Working Flow of New Approach

Step 2: Type the word from the Hindi Keyboard

Suppose the user wants to type: “हिन्दी”

The user input sequence for composing the text should be:“ह + ि + न + ् + द + ी”. When the user enters or removes every character, the combined string is checked by the parser every time.

Step 3: Check the input word with parser

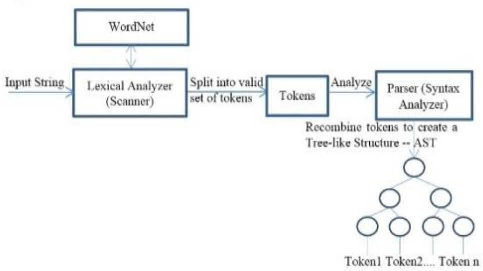
WordNet
Input String
Lexical Analyzer (Scanner)
Split into valid set of tokens
Tokens
Analyze
Parser (Syntax Analyzer)
Recombine tokens to create a Tree-like Structure -- AST
Token1 Token2.... Token n

Figure 5 Working of Parser for Word Prediction

In this example, when user starts typing, the Scanner checks the input letter by letter.

When Scanner gets “ह + ि + न + ्”,

The Scanner checks the input string and Generates Tokens using Regular Expressions.

Finally, it will predict words like: “हिंदी”, “हिन्दी” using Grammar.

Step 4: Check whether the word is available or not

If word is available:

- Print the word
- Generate next Predictions of words

If word is not available:

- Print the word
- Generate common Predictions of words

Step 5: Display the predicted words in Prediction Window

Finally, it will predict words like: “हिंदी”, “हिन्दी”, “हिंदु”, “हिन्दु”, etc. in the prediction window.

IV. CONCLUSION

The survey shows that the current word prediction system for Hindi Language should be improved more so that better words can be predicted. This way there will be no more need of lots of typing. The current algorithms used for most word prediction are based on N-Gram approach. If the algorithms are modified or changed then the word prediction system will be improved. The improvement can be done by including grammar in prediction. The layout of keyboard is another issue faced by users. The layout should be user friendly to improve text typing speed. The new approach to use word prediction with Syntactic N-Gram will work better than the existing systems.

REFERENCES

[1] Amita Jain, Minni Jain, “Detection and Correction of Non Word Spelling Errors in Hindi Language”, 2014 © IEEE, 978-1-4799-4674-7/14

[2] Manoj Kumar Sharma, DebasisSamanta, “Word Prediction System for Text Entry in Hindi”, 2014 © ACM Transactions on Asian Language Information Processing, 1530-0226/2014/06-ART8, DOI:http://dx.doi.org/10.1145/2617590

[3] Manjiri Joshi, AnirudhaN.Joshi, NagrajEmmadi, NiravMalsattar, “Swarachakra Keyboard for Indic Scripts”, 2014 © Association for Computational Linguistics, 978-1-4503-2878-4/14/06, DOI: http://dx.doi.org/10.1145/2593902.2593905

[4] GrigoriSidorov, Francisco Velasquez, EfstathiosStamatatos, Alexander Gelbukh, Liliana Chanona-Hernández, “Syntactic Dependency-Based N-grams as Classification Features”, 2013 © Springer-Verlag Berlin Heidelberg, pp. 1–11.

[5] Bram Jans, Steven Bethard, Ivan Vuli´c, Marie Francine Moens, “Skip N-grams and Ranking Functions for Predicting Script Events”, 2012 © Association for Computational Linguistics.

[6] What is Word Prediction, http://www2.edc.org/NCIP/LIBRARY/wp/What_is.htm

[7] A Guide to Hindi, http://www.bbc.co.uk/languages/other/hindi/guide/facts.shtml

[8] Internet World Stats, http://www.internetworldstats.com/stats.htm

[9] Hindi Speaking and Writing, https://en.wikibooks.org/wiki/Hindi/Speaking_and_Writing

[10] Languages of the World, http://www.ethnologue.com/statistics/size

[11] Number of Native Speakers in India, https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India

[12] National Mission on Education through ICT, http://textofvideo.nptel.iitm.ac.in

[13] Share and Discover Knowledge, http://www.slideshare.net

[14] Gujarat Samachar, http://www.gujaratsamachar.com

[15] Linguistic Data Consortium, Hindi WordNet, https://catalog.ldc.upenn.edu/LDC2008L02