6. Indian Language Spell Checker Design Workshop

Indian Language Spell-Checker Design Workshop at CVPR Unit of ISI, Kolkata on 18-19th July, 2002

During July 18-19, 2002 a two-day workshop entitled "Indian language Spell-checker Design" was organised by the CVPR Unit of ISI, Kolkata at its seminar room. The workshop was sponsored by Ministry of Information Technology, Govt. of India.

The main theme of the workshop was to present the work on Spell-checker done by various Resource Centres and groups in different Indian languages. The participants were also requested to demonstrate the Spell-checker software developed by them. Also, it was hoped that a benchmarking method will be evolved so that the spell-checkers can be tested against that benchmark data.

The participants included representatives from ISI, Kolkata, Dept. of IT (Govt. of India), Central University of Hyderabad, Thaper Institute of Engineering & Technology, Patiala, IISC, Bangalore, IIT, Bombay, C-DAC, Pune, M.S. University, Baroda, IIT, Guwahati, Anna University, Chennai, ER&DCI, Trivendrum, Jadavpur University, Kolkata. In addition, some students of Dept of Linguistics, University of Calcutta also participated in the workshop.

Lectures

Following are the titles and abstracts of some important talks:

1. Title: Detection of Word Error Position and Correction Using Reversed Word Dictionary

Speaker: B. B. Chaudhuri
Computer Vision & Pattern Recognition Unit
Indian Statistical Institute
203, Barrackpore Trunk Road
Kolkata 700108
e-mail: bbc@isical.ac.in

Abstract

A new novel technique of localization and correction of non-word error is described. In this technique a candidate string S of n characters is searched in the conventional dictionary Dc. If S is a non-word, its first k1 (n characters will match with a word in Dc. (If k1 = n then the word in Dc must be longer than n). A reversed word dictionary Dr is also generated where the characters of the word are maintained in a reversed order. If the last k2 characters of S match with a word in Dr then, for single error, it is located within the intersection region of first k1 + 1 and last k2 + 1 characters of S. We observed that this region is very small compared to word length for most cases and the number of suggested correct words can be drastically reduced using this information. We have used our approach in correcting Bangla text, where the problem of inflection is cleverly tackled.

2. Title: Spell Checker Related Work at University of Hyderabad

Speaker: K. Narayan Murthy
University of Hyderabad
Telugu Resource Center
e-mail: knmcs@uohyd.ernet.in

Abstract

We do not have a spell checker as of now but some groundwork is being carried out towards that end. Telugu is an agglutinating language. It is in fact one of the most complex languages of the world as far as structure of words is concerned. A good spell-checker for Telugu is planned to be developed as part of the Resource Center project during the current year.

We have as of today a dictionary of Telugu root words of about 65,000 entries, proving a very good coverage of Telugu language. We also have a Morphological analyzer for Telugu. Tests on available corpora have given 97% plus performance for the morph. analyzer. Given the nature of Telugu language, this is indeed a major achievement. The morphological analyzer, as it was originally developed, was a tightly coupled component of a very large system. After substantial engineering effort, a stand-alone version has been extracted. More work is required in this direction for integration and adaptation for spell checker and other such applications.

Preliminary studies show that it is practically impossible to work directly at the level of full words in the case of Telugu. It is essential to work at the level of smaller units. A practical via-media approach was suggested by me in paper in the Third International Conference on South Asian Languages wherein I had suggested the use of a two level root+combined suffix approach. A thorough and detailed morphological analysis is avoided but at the same time the highly productive nature of Telugu suffixation processes are taken care of. The scheme needs to be implemented and tested.

There is a preliminary version of a Telugu Morphological generator too. Further work is required to adapt this for spell checker development. Going further down the level, basic statistics for Markov Model at syllable level have been obtained. Preliminary studies have shown that the observation probability for invalid words is generally lower than for valid words and this can be exploited to build spell error detection and correction systems. More systematic testing and evaluation of the scheme is required.

A dictionary of 15,000 root words and a morphological analyzer and generator are available for Kannada language. Kannada and Telugu being very similar in structure, work done in one can be useful in the other. The dictionary is being expanded and the morph. will be tested against the corpus and refined. Given all this, a detailed plan of action is being worked out to ensure that a working spell checker can be developed and tested during the current year.

3. Title: Design & Development of Malayalam Spell Checker

Speaker: Santosh Varghese, and
V. N. Shaji
ER&DCI, Trivendrum
Malayalam Resource Center
Tel : 0471-325897

Abstract

A Spell Checker is a tool that will check the spelling of the words in a given text file, validate them and in case the checker has doubts, list out the right spelling(s) in the form of suggestions. The Malayalam Spell Checker being developed by RCILTS-Malayalam at ER&DCI, Thiruvananthapuram, consists of two basic modules namely the Language Module and Spell Checker Engine.

Language module

The Language module for the Malayalam Spell Checker is structured to suit the rule cum dictionary based approach adapted in spell checking. It Consists of three text files (i) A word list (ii) A suffix list (iii) A post-positions list. In order to facilitate easy manipulation of Strings/suffixes, the Roman transliteration of the data is stored. The word list consists of the base forms of words. Different word classes in the word list (verbs, nouns, pronouns, adjectives, adverbs, postpositions etc.) are assigned different paradigm numbers. The suffix list consists of all valid suffixes in the language along with different paradigm numbers for different class of suffixes. The postpositions list have all the available postpositions in Malayalam.

The information regarding all suffixes, their order, the base forms to which they are added on, the changes that the suffixes undergo when they are linked up to each other and to the base forms (sandhi rules) are documented. While coding these rules are used to validate a word.

Spell Checker Engine

The Engine module covers the programming part of the spellchecker. The engine takes word by word from the input file and checks its validity. This mainly covers three types of checking

i) Word Checking checks whether the string is a valid base form.

ii) Suffix checking which will check whether the input string ends with a valid suffix/suffixes.

If so the suffix is stripped off, sandhi rules are applied to reconstruct the base form and the base form subjected to word checking. (The "Sandhi" rules listed in the Language module is coded. The rules are applied on the input string in order to check validity of a word.). Since multiple suffixes are possible in the language it is also being taken care of in programming.

Post-position checking checks whether the input string ends with a valid post- position and if found the post- position is taken off. The remaining word is reconstructed using the sandhi rules. This word is then subjected to steps ii &i. After checking, if the engine recognises the input as a valid word, it goes on to check the next word. If the input word is found invalid, the closest matching word/words will be proposed as suggestion.

4. Title : Tamil Spell Checker

Speaker: T. Dhanabalan
Anna University, Chennai
Tamil Resource Center
Tel : 044-2351723

Abstract

Spell checking application presents valid suggestions to the user based on each mistake they encounter in the user's document. The user then either makes a selection from a list of suggestions or chooses to ignore the suggestions and accepts the current word as valid. Spell checking program is often integrated with word processing software that checks for the correct spelling of words in a document. Each word is compared against a dictionary of correctly spelt words. The user can usually add words to the spell checker's dictionary in order to customize it to his or her needs.

How does the Spell Checker work?

Initially the Spell Checker reads extracted words from the document, one at a time. Dictionary examines the extracted words. If the word is present in the Dictionary, it is interpreted as a valid word and it seeks the next word. If a word is not present in dictionary, it is forwarded to the Error correcting process. The spell checker comprises three phases namely text parsing, spelling verification and correction, and generation of suggestion list. To aid in these phases, the spell checker makes use of the following.

(i) Morphological analyzer for analyzing the given word

(ii) Morphological generator for generating the suggestions.

In this context, the spell checker for Tamil needs to tackle the rich morphological structure of Tamil. After tokenizing the document into a list of words, each word is passed to the morphological analyzer. The morphological analyzer first tries to split the suffix. It is designed in such a way that it can analyze only the correct words. When it unable to split the suffix due to mistake, it passes the word to spelling verification and correction phase to correct the mistake.

Spelling verification and correction

Correcting similar sounding letters

Similar sounding letter can cause incorrect spelling of words. For example consider the word 'Thaalam'. Here the letter 'La' may be misspelled as 'la'. Suggestions are generated by examining the entire possible similar sounding letters for the erroneous word.

Checking the Noun

Tasks in noun correction include Case marker correction, plural marker checking, postposition checking, adjective checking and root word correction.

Checking the Verb

Verb checking tasks include Person, Number & Tense marker checking and root word checking.

Correcting the adjacent key errors

User can mistype one letter instead of one letter. So we have to consider all the possible adjacent keys of that particular letter. If any adjacent key of the mistyped letter matches with the original letter then that letter is replaced instead of mistyped one and the dictionary is checked.

When the correction of errors is completed, root word and all components are sent to morphological generator (for word forming), which then generate the possible corrected words as suggestions.

Dictionary Look Up

Dictionary contains all the root words like noun, verb, adjective, adverb and participles. If there is any spelling mistake in the root word, that word has to be corrected or all the nearest matching dictionary entries are collected and sent to the suggestion generation phase.

Conclusion

The Spell checker for Tamil helps the user to identify most of errors, which may occur while typing. The tasks implemented in Tamil Spell checker are Case marker, postposition checking and adjective checking for nouns, PNG marker checking for verbs, Adverb checking, and adjacent key error checking.

5. Title : A Spell checker for Punjabi

Speaker: G. S. Lehal
Thapar Institute of Engg. & Technology, Patiala
Punjabi Resource Center
e-mail: gslehal@mailcity.com

Abstract

A spell checker for Punjabi has been developed at RC-ILTS Punjabi, TIET. The spell checker uses a database of Punjabi words taken from Punjabi corpus, Punjabi dictionary and general knowledge books of Punjabi. As there is no standardization of spelling of Punjabi words so all the spellings of each Punjabi word are take. The size of the database is 0.15 million words. The database is split into fifteen files based on the word length. The files correspond to word lengths from 2 to 15 and for word lengths greater than 15.

When a word is sent to the spell checker it searches for it in the file corresponding to its word length. In case the word is not found, a suggestion list is generated based on the four common tying errors a) insertion of extra character b) deletion of a character c) swapping of characters and d) wrongly typed characters.

6. A Study of Spellchecker for the Urdu Language

Speaker: Munawwar Ali, DRM,
Urdu RC Project, CDAC Pune,
e-mail: munawwar. ali @ cdac. ernet. in

Abstract

Description of the Urdu Language

Urdu has its origin in Arabic and Persian languages but is also influenced by Hindi and Sanskrit. Its alphabet is a super set of Arabic and Persian and contains 39 characters. The characters of Urdu also need diacritics to help in the proper pronunciation of the constituent word. The diacritics appear above or below a character to define a vowel or emphasize a particular sound.

The Vowel and Diacritic Marks

The earliest Arabic script had no vowels, as the structure of the language and the context served to make the passage clear. Later on, however, a set of diacritics fcabar, zer, and pesh) was developed, written above or below the consonant symbol, which they follow. However, the vowel diacritics are almost never written in materials intended for adult speakers of the language.

Homonyms

The language has a lot of consonants that have similar sounding pronunciation. For example consonant “te” (as in “tarbooz” [watermelon]) and consonant “toe” (as in ‘tota” [parrot]). Similarly consonants “se” (as in “samar” [fruit]), consonant “seen” (as in “sailaab” [flood]), consonant “suad” (as in “sabr” [patience]). And similarly consonant “ze” (as in zebra), consonant “zaal” (as in ‘^kr”), consonant “zoe” (as in ‘^aroor”), etc.

Prefixes/Suffixes

The language tends to have a lot of prefixes and suffixes derived from Arabic and Persian languages. A large number of words can have any of these suffixes and prefixes. It is hard to segregate these into separate

groups of suffixes/prefixes and associate to a fixed set of words.

Furthermore, a lot of word can be derived from a single word or root word. For example “jaana” is a word (meaning to go). Unlike English where (only goes and going are the two derived forms), the words that can be derived from it can be many e.g. “jaataa”, “jaati”, “jaate”, “jaao”, “jaayenge”, “jaaiye”, etc. All these must/should be identified as correct words by the Spellchecker. But all of them cannot be put in the dictionary, as it will enormously increase the size of the dictionary and data entry work

A Well Formed Dictionary (The Spellchecker Dictionary)

Considering the complexities of the language, only a dictionary of plain word would not be that useful for providing good suggestions. The dictionary for the Spellchecker would be the most important part of the Spellchecker, as it would give complete details of a word and its language specific data.

What should go in the dictionary?

What shall give more options for suggestions?

What would make the search better?

Only a well-designed dictionary could give answers to above questions.

Storing Root Words

Instead of storing the word as it is in the dictionary, if the word is saved in a reduced form, it would give more options for a search that can give suggestions. The information to get back the original word would also be stored in the dictionary.

Some sort of rules can be made to reduce a word into root word. For example vowels may be removed, homonyms may be replaced with a chosen letter representing that group of letters, etc.

Storing prefix/suffix Information

The prefix and suffix information would be stored separately in the dictionary and rules may be formed so that using use these rules, all sort of derived words can be regenerated.

The Dictionary Structure File Dictionary Structure

The structure that is how the data will be stored on the disk and read from there into memory is also important. The dictionary can be stored as a collection of files having words as per their lengths (number of letters in it), OR the entire dictionary may be stored in one file.

The File Dictionary may be reproduced with the memory dictionary and the memory data structures can be directly saved in it so that they can be read from it to reduce the load time.

Memory Dictionary Structure

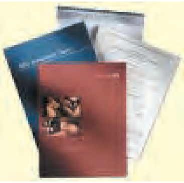
The memory data structures to represent the dictionary would again be important, as the search algorithm would be dependant on it. The dictionary may be stored in memory in the forms of relational tables. Indexing table may be generated for faster location of root word and other entries. The words may be loaded in word tables (different length word in different tables).

The Search Algorithm

One may have to search the dictionary a number of times with different conditions to get suggestions. The first thing would be to find whether the input word is stored as correct word in the dictionary. This would mean converting the input word to the equivalent root word and then searching it in the dictionary for a matching root word.

A separate search may be needed to find whether a suffix or prefix is there in the input word. Suffix stripping algorithms may be applied to separate the suffix or prefix before searching the root word.

One may also use fuzzy search so that if the exact root word is not found in the dictionary, a percentage-wise comparison result may be used to select suggestions. But this search, off course, cannot be applied to the entire dictionary. The likely word range would have to be selected. This may be done by considering the length of the word, for example, if the length of input word in N, only words having



lengths in the range N-I to N+I can be searched, where I can be 1 or 2 or 3 depending on the value of N.

Furthermore, techniques like comparing adjacent letters by swapping two letters, or comparing letters by shifting one position may be use to get away with mistakes of one letter or two letters swapped.

End Notes

It is important to have a dictionary with information such as root word, information to regenerate root word, prefix and suffix information. At the same time a well designed data structure that would represent this dictionary. The spellchecker dictionary would further be used as a lexical resource by adding tags and grammar information, so that it may be extended for a grammar checker.

Secondly, the search algorithms have to be a mixture of searches like string matching, fuzzy search, and suffix stripping, etc. This would probably increase the correct suggestion rate and number of suggestions.

7. Title : Detection And Correction Of Phonetic Errors In Alphabetic Languages

Speaker: Sivaji Bandyopadhyay
Computer Science & Engineering Department
Jadavpur University
Kolkata – 700 032.
e-mail: sivaji_ju@vsnl.com

Abstract

The Phonetic and the Homophone Error problem in a language have been characterized as a symbol substitution problem. Phonetically equivalent symbols or symbol combinations in the language are grouped together. Each group or a number of related groups give(s) rise to a dictionary or a number of dictionaries. A new design methodology for Orthographic dictionaries in alphabetic languages has been described. The dictionaries include the root words. The meanings are stored only in case of Homophone words. Words are sorted on the basis of a Phonetic Ordering Scheme. The dictionaries are being used to detect and correct the Phonetic Error and the Homophone Error in isolated words of Bengali. The Orthographic dictionaries can be used to detect Cognitive Errors and suggest a possible set of corrections. Work has now started for context dependent word correction.

Demonstrations

- RC-Bangla of ISI, Kolkata demonstrated Bangla Spell-checker.
- RC-Tamil of Anna University, Chennai demonstrated Tamil Spell-checker.
- RC-Punjabi of Thapar Institute of Engg. & Technology, Patiala demonstrated Punjabi spell-checker.

Discussion Session

The following is the Language-wise summary of discussion session:

Malayalam

- Root and suffix rules formation for morphological processor
- Spell-checker work not yet started. Dictionary would be completed by 2002

Punjabi

- Words, suffix are stored in a dictionary
- Corpus collection from CIIL
- Total size of lexicon from corpus is 1.35 lakhs
- Two dictionaries used:

  (a) Dictionary of words having same size

  (b) Dictionary of words having single error
- Start work for errors due to insertion/deletion of characters in words
- Apply reverse word-dictionary
- Engine for spell-checker

Assamese

- Will get an Assamese corpus from CIIL, sort it, collect the vocabulary, and store it into the dictionary
- Going to start Spell-checker work

Urdu

- Plain dictionary of 50,000 words for Urdu language
- Spell-checker work is about to start

Tamil

- Morphological analyser, generator
- Root word dictionary of 50,000 words
- Initial prototype of Spell-checker will be made within 3 months

Bangla

- Single error correction
- Multi-error detection and correction is started
- Software is under distribution

Telugu

- Telugu morphological analyser
- Dictionary
- Spell-checker for error detection work started
- Only prototype by 2002

Gujarati

- Standard format for character encoding
- Spell-checker work yet to start

Marathi

- Just initiated work

Kannada

- Participants from RC-Kannada were absent during the 2nd day of the workshop.

MIT

- It can evaluate all systems and products

For all Centres

- 1 year more will be given for each centre
- Each product should have minimum price
- Uniform and unbiased way of judgement
- If a valid word is stopped, spell-checker system is to pay a penalty
- If a wrong word is not stopped spell-checker system is to pay a penalty
- More marks for more inflectional languages
- Simulated error pattern both for one and two character position
- Start work for real word error data collection and analysis
- Real word error can be passed through the large dictionary
- Testing evaluation benchmark
- Centre should prepare test-data for validation
- Texts should be taken randomly for testing
- Story books (it may have specialised words, dialects, etc.)
- Devise something for foreign words, e.g., English, Hindi etc.
- Proper names (recurrent proper names should be passed by spell-checker)
- The scientific terms and names can be collected from the corpus and stored in an auxiliary dictionary. For the time we can ignore it.
- Are spell-checkers able to give suggestions
- Number of alternatives, less the number the better, the order of priority (*at)
- Use total vocabulary from the DOE corpus as test-beds
- Compare individual scores and put into together
- Put all dictionaries in XML so that they can be uniform and self-evident