IMPLEMENTATION OF SPELLING ERROR WORDS CORRECTING SYSTEM USING BI-GRAM MODEL AND APPROXIMATE STRING MATCHING ALGORITHM

Phyo Hai Mar Wai
Computer University (Monywa)
phyohaimarwai@gmail.com

ABSTRACT

Natural language processing is a subfield of AI. Natural languages are human languages such as English and Chinese, etc. NLP enables computer systems to understand written or spoken utterances made in human languages. There are many steps in NLP. This system uses morphological analysis steps for spelling checking. The system uses bi-gram model to reduce search space and then approximate string matching is used to suggest correct word. This system will assume the words in the sentences, containing in the lexicon or dictionary. Even the word is correct, however, if the word is not containing in the dictionary, the known word part of speech tagging process will determine the word as an misspelled word. Thus, the system assumes the words of the input sentences are containing in the dictionary. The system also provides to update dictionary to add new vocabulary.

1. INTRODUCTION

Natural Language Processing (NLP) is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. Natural language generation systems convert information from computer databases into readable human language. Natural language understanding systems convert samples of human language into more formal representations such as parse trees or first-order logic structures that are easier for computer programs to manipulate. Many problems within NLP apply to both generation and understanding; for example, a computer must be able to model morphology (the structure of words) in order to understand an English sentence, and a model of morphology is also needed for producing a grammatically correct English sentence [8].

Natural-language understanding is sometimes referred to as an AI-complete problem, because natural-language recognition seems to require extensive knowledge about the outside world and the ability to manipulate it. The definition of understanding is one of the major problems in natural language processing [8] [9].

In computing, a spell checker (or spell check) is an application program that flags words in a document that may not be spelled correctly. Spelling checkers may be stand-alone capable of operating on a block of text, or as part of a larger application, such as a word processor, email client, electronic dictionary, or search engine. Spelling suggestion is a feature of many computer software applications.

Spelling is the writing of a word or words with the necessary letters and diacritics present in an accepted standard order. It is one of the elements of orthography and a prescriptive element of alphabetic languages. Most spellings attempt to approximate a transcribing of the sounds of the language into alphabetic letters, however, completely phonetic spellings are often the exception , due to drifts in pronunciations over time and irregular spellings adopted through common usage.

Simple spell checkers operate on individual words by comparing each of them against the contents of a dictionary, possibly performing stemming on the word. If the word is not found it is considered to be an error, and an attempt may be

made to suggest a word that was likely to have been intended [8].

This thesis paper is intended to make suggestions for misspell words for English language. The system use bi-gram model to reduce search space for suggested word area and also us approximate string matching algorithm to find the most appropriate suggested words.

2. RELATED WORKS

N-gram models are widely used in statistical natural language processing. In speech recognition, phonemes and sequences of phonemes are modeled using a n-gram distribution. For parsing, words are modeled such that each n-gram is composed of n words. The most common application of approximate matchers until recently has been spell checking. Aminul Islam [Department of Computer Science University of Ottawa, Canada] has been proposed the paper "Real-Word Spelling Correction using GoogleWeb 1T 3-grams" for detecting and correcting multiple real-word spelling errors using N-gram model [1]. Mohammad Ali Elmi and Martha Evens [Department of Computer Science, Illinois Institute of Technology] have been proposed the paper "Spelling Correction Using Context* " to propose a method for finding unknown words using syntatic and semantic knowledge [6].

3. NATURAL LANGUAGE PROCESSING

The most common way that people communicate is by speaking or writing in one of the natural languages, like English. On the other hand, computer programming languages seem awkward to humans.

These artificial languages are designed so that the sentences have a rigid format, or syntax, making it easier for compiler to parse a program and convert it into the proper sequence of computer instruction.

Natural language processing (NLP) is a wide spread part of Artificial Intelligence, looking for utilization all in computer both as a communication between human and computer [8]. In theory, natural language processing is a very attractive method of human-computer interaction.

There are five steps in NLP:

Morphological analysis: Individual words are analyzed into their components and non-word tokens such as puntuation are separated from the words [4].

Syntactic analysis: Linear sequences of words are transformed into structures that show how the words relate to each other [2].

Semantic analysis: The structures created by the syntactic analyzer are assigned meanings.

Discourse integration: The meaning of an individual sentence may depend on the sentences that precede it and may influence the meanings of the sentences that follow it.

Pragmatic analysis: The structure representing what was said is reinterpreted to determine what was actually meant.

3.1. Level of analysis

- Major Morphology - word
  ⇩
- Syntax- sentence structure
  ⇩
- Semantics - meaning
  ⇩
- Discourse/Pragmatics - context, user intention

3.2. Morphological analysis

Morphological analysis is the first analysis stage in a NLP system. Morphology is the components that make up words. Often these components have grammatical significance, such as '-es', '-ed', 'ing'. It is useful in identifying which part of speech (noun, verb, etc.) a word is.

This level deals with the componential nature of words, which composed of morphemes- the smallest units of meaning. For example, the word stems like "preregistration" can be morphologically analyzed into three separate morphemes: the prefix -pre, the root -registra, and the suffix -tion.

3.3. Roles of morphological analysis

There are three roles of morphological analysis:

Tokenization: The early step of processing is to divide the input text into units called tokens where each is either a word or something else like a number or punctuation mark. So, the tokenizer takes as input text and splits it into its tokens. A token is used to separate by morphology analysis.

Spelling correction: A spell checker customarily consists of two parts:

1. A set of routines for scanning text and extracting words
2. An algorithm for comparing the extracted words against a known list of correctly spelled words (ie. , the dictionary) [3].

Part of speech tagging: Part of speech tagging is the process of marking up the words in a text as corresponding to a particular part of speech, based on both its definition, as well as its context -ie. Relationship with adjacent and related words in a phrase, sentence, or paragraph.

4. N-GRAM

An *n*-gram is a subsequence of *n* items from a given sequence. The items in question can be phonemes, syllables, letters, words or base pairs according to the application [8].

An *n*-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram"; and size 4 or more is simply called an "*n*-gram". Some language models built from n-grams are "($n-1$)-order Markov models".

An *n*-gram model is a type of probabilistic model for predicting the next item in such a sequence. *n*-gram models are used in various areas of statistical natural language processing and genetic sequence analysis.

4.1. Bi-gram

Bi-grams or di-grams are groups of two written letters, two syllables, or two words, and are very commonly used as the basis for simple statistical analysis of text. They are used in one of the most successful language models for speech recognition. Bi-grams are word pairs which allow gaps bi-grams help provide the conditional probability of a word given the preceding word, when the relation of the conditional probability is applied:

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})}$$

That is, the probability $P()$ of a word W_n given the preceding word W_{n-1} is equal to the probability of their bigram, or the co-occurrence of the two words $P(W_{n-1},W_n)$, divided by the probability of the preceding word.

5. APPROXIMATE STRING MATCHING

Approximate string matching is the technique of finding approximate matches to a pattern in a string.

The closeness of a match is measured in terms of the number of primitive operations necessary to convert the string into an exact match [7]. This number is called the edit distance between the string and the pattern. The usual primitive operations are:

- insertion: *cot* → *co**a**t*
- deletion: *co**a**t* → *cot*
- substitution: *co**a**t* → *co**s**t*

These three operations may be generalized as forms of substitution by adding a NULL character (here symbolized by λ) wherever a character has been deleted or inserted:

- insertion: *co**λ**t* → *co**a**t*
- deletion: *co**a**t* → *co**λ**t*
- substitution: ***coat*** → ***cost***

Some approximate matchers also treat transposition, in which the positions of two letters in the string are swapped, to be a primitive operation.

Changing *cost* to *cots* is an example of a transposition. Different approximate matchers impose different constraints. Some matchers use the total number of primitive operations necessary to convert the match to the pattern.

For example, if the pattern is *coil*, *foil* differs by one substitution, *coils* by one insertion, *oil* by one deletion, and *foal* by two substitutions. If all operations count as a single unit of cost and the

limit is set to one, *foil*, *coils*, and *oil* will count as matches while *foal* will not.

5.1. Edit distance or Levenshtein distance

In information theory and computer science, the Levenshtein distance is a metric for measuring the amount of difference between two sequences (ie an edit distance). The term edit distance is often used to refer specifically to Levenshtein distance [9].

The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character.

For example, the Levenshtein distance between "kitten" and "sitting" is 3, since the following three edits change one into the other, and there is no way to do it with fewer than three edits:

1. **k**itten → **s**itten (substitution of 's' for 'k')
2. sitt**e**n → sitt**i**n (substitution of 'i' for 'e')
3. sittin → sittin**g** (insert 'g' at the end).

6. PROPOSED SYSTEM

In a traditional spelling checking system, the system found the suggested corrected words comparing the misspelled unknown word with all words in the dictionary. So, it is very time comsuming and cann't display suggested correct words with nearest possible form. This system use context of the mispelled unknown word and determine the possible part of speech of the mispelled unknown word by using bi-gram token sequence and then search the possible suggested corrected words within the scope of the determined part of speech not all words in the dictionary. When searching in the dictionary, the system calculates the edit distance of each word by using approximate string matching Algorithm. After then, the system displays the suggested correct words in the form of ascending order of the edit distances. So, the system can reduce the time comsuming for searching the correct words and can display with the nearest possible form by using edit distance.

7. DESIGN OF THE PROPOSED SYSTEM

User must input text to the system. Then the text is divided into word (token). After tokenization process, search each word in the wordlist whether it exists in the dictionary or not. If found, classify Part Of Speech of the input token and if not, determine spelling error word and the system assume the word as unknown word.

After search process completed the word list is converted into Part Of Speech token list. Before the spelling error word suggestion, create the bi-gram sequence of the tokens to reduce search space. Then by using the approximate string matching, the system can suggest the spelling error word in the reduced search space.

Finally, the system displays the suggestion of the correct words of the misspelled unknown words that does not exist in the lexicon.The suggested correct words are displayed ascending order of the edit distances of the words.

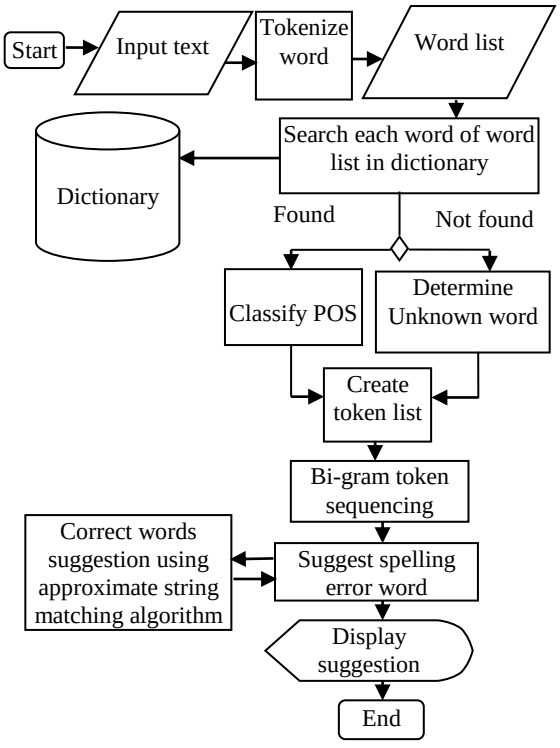

Figure 1. System flow diagram

8. IMPLEMENTATION OF THE SYSTEM

Suggest menu represents the important tasks of spelling checking process and suggestion process. When Create Bi-gram Sequence menu is choose, the system creates bi-gram sequence for the current sentence and shows in bi-gram sequence list.

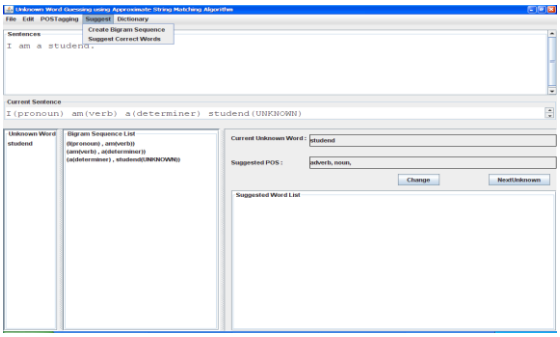

Figure2. Creation of bi-gram sequence

After creating bi-gram sequence for the current sentence is completed, the system shows the possible POS of the unknown word is displayed on the right pane. When user choose Suggest Correct words from suggest menu, the system calculates the edit distance of the words in the noun group and shows the 5 possible corrected words with least Edit distance value.

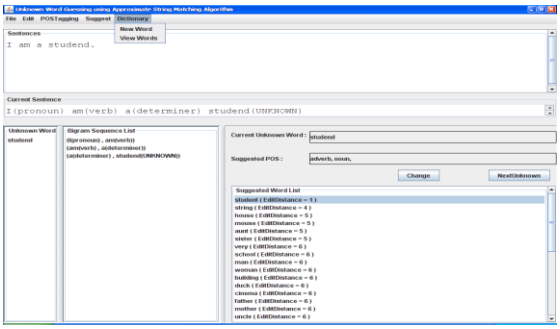

Figure3. Suggestion of correct word

9. CONCLUSION

Natural language processing is a very interesting topic in computer science. This system is to check spelling for English sentence. This system uses bi-gram model to predict and reduce search space of words. Then, approximate string matching algorithm is used to search a correct word. Suggested words are sorted using edit distance value in ascending order. The system displays the most top 5 words as same as the misspelled words.

This system is implemented as a standalone desktop application. Also, the system is developed by Java and therefore the system can be used in windows platform. The system can be extended to a web-based system. In addition, displaying the meaning of the suggested words is a good feature to be added in the system.

REFERENCES

[1] Aminul Islam, "Real-Word Spelling Correction using GoogleWeb 1T 3-grams", Deparment of Computer Science, University of Ottawa, Canada, 2002.

[2] Diana McCarthy, "Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations", Ph.D.thesis, University of Sussex, 2001.

[3] F. Damerau ,"A Technique for Computer Detection and Correction of Spelling Errors", 1999.

[4] Haruo Kubozono, "Phonological constraints on blending in English as a case for phonology-morphology interface", 2003.

[5] Janine Toole, "Categorizing unknown words: using decision trees to identify names and misspelling", 2000.

[6] Mohammad Ali Elmi and Martha Evens, "Spelling Correction Using Context*", Department of Computer Science, Illinois Institude of Technology, 2003.

[7] Z. Galil and K. Park "An improved algorithm for approximate string matching", *SIAM Journal of Computing*, 2003.

[8] http://www.en.wikipedia.org/wiki/Natural Language Processing/Spell Checker/N-gram Model/Bi-gram

[9] http://www.enclylclopedia.com/Natural Language Understanding/Edit Distance

.