Laempel System
for
Intelligent Text
Processing

System Description

Spell Checking

The Laempel spell checking system is an intelligent spell checking system that, among other key features, learns user behavior. Based on that insight, the system with high likelihood suggests correct replacements for incorrect words and declares unknown but correct words to be correct. The system relies on three dictionaries, a so-called user history file, and two logic modules to carry out the learning and spell checking.

A key element that sets the system apart from other systems is learning of user behavior. Altogether, the system has the following features.

- Each word that the user considers to be incorrect is almost surely flagged as incorrect.
- For each word flagged as incorrect, a replacement word is suggested that very likely is the one desired by the user.
- Each word that the system does not know but that the user considers to be correct, is very likely declared to be unknown but correct.

This work was funded in part by the Office of Naval Research under grant N00014-93-1-0096.

Version 7.0 12 September 2005

- Each word that the user previously defined to be incorrect is always declared to be incorrect.
- The spell checking is done rapidly and in a way that places no undue burden on the user.

The spell checking system relies on the following approach.

When a user invokes the system in a given domain for the first time, the system relies on a general dictionary to determine spelling errors and suggests corrections. The dictionary has in excess of 200,000 entries and thus contains most common words of the English language. The spell checking process creates and maintains a second dictionary called user dictionary which lists all words the user has employed so far. There is a third, typically very small, excluded words dictionary that is defined and updated by the user. It contains words that the user will not accept as correct regardless of circumstances.

Suppose the system processes some text. For each word, it checks whether the word
- occurs in the excluded words dictionary, or
- occurs in the user dictionary, or
- occurs in the general dictionary, or
- can be constructed from a root word in the general dictionary, or
- likely is the result of a spelling or typing error, or
- likely is considered correct by the user.

The error detection and correction tasks are carried out as follows. Suppose a given word is not in the dictionaries and is not constructible from some root word. We define such a word to be unknown. Then, a logic formulation relates various characteristics of the given unknown word and the user's recent spelling and typing behavior, to spelling or typing errors at certain likelihood levels. Reasoning based on that logic formulation identifies applicable spelling and typing errors and associated likelihood values, and eventually produces up to three replacement words.

If the systems finds replacement words, it ranks them according to weights that express the likelihood of applicability. The user accepts

one of the proposed replacement words, or supplies another word, or declares the word in question to be actually correct.

If the system does not find any replacement words, it assumes that the unknown word is actually correct. The user confirms that assumption or supplies another word.

Regardless of which case applies, the system analyzes the user response, correspondingly updates the user dictionary and user history file, and thus learns user behavior and preferences. Goal of that learning is an improvement of the accuracy with which the system produces and ranks replacement words for incorrect words and declares unknown words to be actually correct.

The spell checking system can learn special vocabularies. Some text files, for example LaTeX files, contain special control words that the system might diagnose as incorrect. To speed up learning of such special vocabularies, one may declare a given text to be entirely correct. The system processes such a text without any user interaction, by assuming that any word determined to be unknown is correct.

Syntax Checking

The syntax checker of Laempel consists of three steps.

In the first step, the given text is cleaned up by a screening process that replaces special symbols and formulas by a special token called UNDEF. If a sentence after that change contains more UNDEF instances than word instances, then syntactic checking of that sentence is considered impossible, and the sentence is ignored. Otherwise, each string of consecutive UNDEF instances in a sentence is replaced by a single UNDEF instance. An UNDEF instance theoretically may be assigned any part-of-speech. But in most cases, UNDEF serves as a singular or plural noun or as an adjective.

In the second step, two logic modules check the cleaned text for local syntactic errors. A total of 27 different cases are considered. Examples are duplicated words such as "the the," duplicated auxiliaries such as "he'll will," "because of" followed by a nominative pronoun such as "because of she", improper use of prepositions such as "by ignore," and double negation such as "they can't hardly speak."

The third step is applied to each sentence that does not contain any local syntactic errors. A reasoning process involving 18 logic modules analyzes each such sentence for global syntactic errors. If no such error is determined, the process attempts to parse the sentence. We say "attempts" since the process gives up on parsing if the sentence is so complex that the 18 logic modules become bogged down in the parsing process. In tests, the percentage of sentences that were parsed by the syntax checker ranged from 100% for simple texts and 76% for a mathematical text to 61% for a TV network news text. For the sentences that have been parsed, Laempel records for each word the assigned part-of-speech. That information is utilized later to estimate whether a given word has a dominant part-of-speech. The 18 logic modules contain common-sense reasoning about syntax and may flag a sentence with awkward but correct structure as possibly incorrect. Thus, Laempel not only finds syntax errors, but also points out poorly constructed sentences. Analogously to the spell checker part, the syntax checker

learns from the user response and adapts its reasoning to the user style of writing.

System Files

This section describes the files required by the Laempel system. The file names are determined by the user in file params.dat. Almost all files are maintained by the system and cannot be changed by the user. The user can only edit two files: params.dat and nono.usr, which is a file of excluded words.

File params.dat

The directory where Laempel programs are executed must contain the file params.dat. A template of that file is in the subdirectory Laempel/Data. The file is listed and explained below.

```
*** Laempel System Version 7.0 Parameter File
*** (programs laempel.spl and laempel.stx)
***
*** CAUTION: Do not execute laempel.spl or laempel.stx
*** in the Workarea subdirectory of Laempel,
*** and do not specify any of the files below
*** to be in that directory.
*****************************************************
*** File names (include paths if needed)
user history file = history.usr
user dictionary = words.usr
user dictionary backup = words.bak
user phrase pattern = patterns.usr
user phrase pattern backup = patterns.bak
user suggested replacements = replace.usr
file of words excluded by user = nono.usr
file name storage = filename.sav
*****************************************************
*** Spell Checking Option
```

```
accept all unknown words as correct (yes, no) = no
*****************************************************
*** Syntax Checking Options
find syntactical interpretations (yes, no) = yes
interactive or batch syntax checking (int, bat) = int
*****************************************************
*** Directory of Data
***
directory of Laempel/Data = /home/Laempel/Data
*****************************************************
ENDATA
```

The first section of params.dat contains the names of the files required by the various Laempel programs.

The statement
`user history file = history.usr`
specifies that the file containing the user history of making spelling and typing mistakes is history.usr. This file is manipulated by the spell checker.

The statements
`user dictionary = words.usr`
`user dictionary backup = words.bak`
indicate that the user dictionary is in file words.usr and the backup of the dictionary is in file words.bak. The user dictionary includes information on all words the user has employed so far. This file is required by all Laempel programs.

The statements
`user phrase pattern = patterns.usr`
`user phrase pattern backup = patterns.bak`
specify that the file and its backup in which the syntax checker records special word patterns employed by the user are patterns.usr and patterns.bak, respectively.

The statement
`user suggested replacements = replace.usr`

tells that the file replace.usr is to include certain replacement words that the user specified in previous runs. This file is controlled by the spell checker.

The statement
`file of words excluded by user = nono.usr`
specifies that the file containing the words that the user considers to be incorrect, is nono.usr. The user may edit this file; details are given below. The file nono.usr is required by the spell checker.

The statement
`file name storage = filename.sav`
specifies that the file filename.sav includes the name of the most recent file processed by the Laempel system. This file is maintained by the spell checker and the syntax checker.

The next section in params.dat determines one spell checking option.

The statement
`accept all unknown words as correct (yes, no) = no`
tells the spell checker whether or not to accept unknown words as correct. This feature allows the spell checker to quickly learn special vocabulary that the user considers as correct. For example, a user can specify 'yes' at this option and run the system on a correct LaTeX document so that the system quickly learns the control words in LaTeX files.

The third section in params.dat specifies the syntax checking options.

The statement
`find syntactical interpretations (yes, no) = yes`
specifies whether the syntax checker is to find syntactical interpretations for sentences or only identifies syntactical errors.

The statement
`interactive or batch syntax checking (int, bat) = int`
tells the syntax checker to perform the syntax checking in an interactive mode or batch mode. In batch mode, the system outputs the error

messages into a file having the name of the input file plus the extension '.err' without any user interaction.

Finally, required path names are specified.

The path name for the Laempel/Data subdirectory is needed for all checking steps and is given by
`directory of Laempel/Data = /home/Laempel/Data`

File nono.usr

The name of the file is specified in params.dat on line
`file of words excluded by user = nono.usr`
A typical file that specifies British use of some words is listed below.

```
colour
analyse
labelled
labelling
guaranty
towards
```

Each line in the file includes a word that the user considers as incorrect regardless of circumstances. For example, if the user favors American English over British English, nono.usr might contain "colour". Updating of the file is controlled by the user. In a typical situation, the user discovers a word in a file that should not have been used but has been accepted by the spell checker. The user then adds that word to the file of excluded words to prevent future acceptance of the word.