Words Reordering based on Statistical Language Model

Theologos Athanaselis, Stelios Bakamidis, and Ioannis Dologlou

***Abstract*—**There are multiple reasons to expect that detecting the word order errors in a text will be a difficult problem, and detection rates reported in the literature are in fact low. Although grammatical rules constructed by computer linguists improve the performance of grammar checker in word order diagnosis, the repairing task is still very difficult. This paper presents an approach for repairing word order errors in English text by reordering words in a sentence and choosing the version that maximizes the number of trigram hits according to a language model. The novelty of this method concerns the use of an efficient confusion matrix technique for reordering the words. The comparative advantage of this method is that works with a large set of words, and avoids the laborious and costly process of collecting word order errors for creating error patterns.

***Keywords*—**Permutations filtering, Statistical language model N-grams, Word order errors

I. Introduction

AUTOMATIC grammar checking is traditionally done by manually written rules, constructed by computer linguists. Methods for detecting grammatical errors without manually constructed rules have been presented before. Atwell [1] uses the probabilities in a statistical part-of the speech tagger, detecting errors as low probability part of speech sequences. Golding [2] showed how methods used for decision lists and Bayesian classifiers could be adapted to detect errors resulting from common spelling confusions among sets such as “there”, “their” and “they’re”. He extracted contexts from correct usage of each confusable word in a training corpus and then identified a new occurrence as an error when it matched the wrong context. Chodorow and Leacock [3] suggested an unsupervised method for detecting grammatical errors by inferring negative evidence from edited textual corpora. Heift [4],[5] released the German Tutor, an intelligent language tutoring system where word order errors are diagnosed by string comparison of base lexical forms. Bigert and Knutsson [6] presented how a new text is compared to known correct text and deviations from the norm are flagged as suspected errors. Sjobergh [7] introduced a method of grammar errors recognition by adding errors to a lot of (mostly error free) unannotated text and by using a machine learning algorithm.

Unlike most of the approaches, the proposed method is applicable to any language (language models can be computed in any language) and does not work only with a specific set of words. The use of parser and/or tagger is not necessary. Also, it does not need a manual collection of written rules since they are outlined by the statistical language model.

The paper is organized as follows: the architecture of the entire system and a description of each component follow in section 2. The language model is described in section 3. The 4th section shows how permutations are filtered by the proposed method. The 5th section specifies the method that is used for searching valid trigrams in a sentence. The results of using TOEFL’s experimental scheme are discussed in section 6. Finally, the concluding remarks are made in section 7.

II. System Architecture

It is straight forward that the best way for reconstructing a sentence with word order errors is to reorder the words. However, the question is how it can be achieved without knowing the attribute of each word. Many techniques have been developed in the past to cope with this problem using a grammar parser and rules. However, the success rates reported in the literature are in fact low. A way for reordering the words is to use all the possible permutations. The crucial drawback of this approach is that given a sentence with length N words the number of all permutations is N!. This number is very large and seems to be restrictive for further processing. The novelty of the proposed method concerns the use of a technique for filtering the initial number of permutations. The process of repairing sentences with word–order errors incorporates the followings tools:

1. a simple, and efficient confusion matrix technique
2. and language model’s trigrams and bigrams.

Consequently, the correctness of each sentence depends on the number of valid trigrams. Therefore, this method evaluates the correctness of each sentence after filtering, and provides as a result, a sentence with the same words but in correct order.

Manuscript received January 9, 2006. Theologos Athanaselis is with the Institute for Language and Speech Processing, Artemidos 6 and Epidavrou, Maroussi, Athens, Greece, GR-15125, (phone: +302106875416; fax:+302106854270; e-mail: tathana@ilsp.gr).

Stelios Bakamidis, is with the Institute for Language and Speech Processing, Artemidos 6 and Epidavrou, Maroussi, Athens, Greece, GR-15125, (e-mail: bakam@ilsp.gr).

Ioannis Dologlou is with the Institute for Language and Speech Processing, Artemidos 6 and Epidavrou, Maroussi, Athens, Greece, GR-15125, (e-mail: ydol@ilsp.gr).

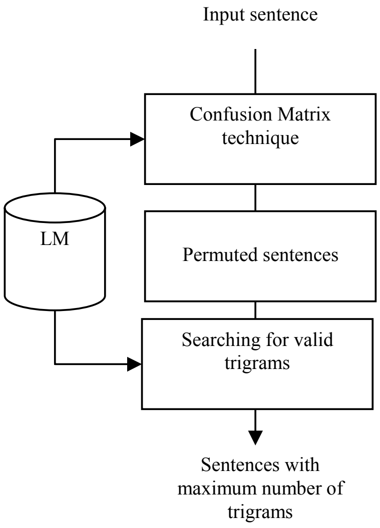
Input sentence
Confusion Matrix technique
LM
Permuted sentences
Searching for valid trigrams
Sentences with maximum number of trigrams

Fig. 1 The architecture of the proposed system

III. Language Model

The language model (LM) that is used subsequently is the standard statistical N-grams [8]. The N-grams provide an estimate of $P(W)$, the probability of observed word sequence W. Assuming that the probability of a given word in an utterance depends on the finite number of preceding words, the probability of N-word string can be written as:

$$P(W)=\prod_{i=1}^{N}P(w_i \mid w_{i-1},w_{i-2},...,w_{i-(n-1)}) \quad (1)$$

One major problem with standard N-gram models is that they must be trained from some corpus, and because any particular training corpus is finite, some perfectly acceptable N-grams are bound to be missing from it. That is, the N-gram matrix for any given training corpus is sparse; it is bound to have a very large number of cases of putative “zero probability N-grams” that should have some non zero probability. Some part of this problem is endemic to N-grams; since they can not use long distance context, they always tend to underestimate the probability of strings that happen no tot have occurred nearby in their training corpus. There are some techniques that can be used in order to assign a non zero probability to these zero probability N-grams. In this work, the language model has been trained using BNC and consists of trigrams with Good-Turing discounting [9] and Katz back off [10] for smoothing. BNC contains about 6.25M sentences and 100 million words.

IV. Filtering Permutations

Considering that an ungrammatical sentence includes the correct words but in wrong order, it is plausible that generating all the permuted sentences (words reordering) one of them will be the correct sentence (words in correct order). The question here is how feasible is to deal with all the permutations for sentences with large number of words. Therefore, a filtering process of all possible permutations is necessary. The filtering involves the construction of a confusion matrix NxN in order to extract possible permuted sentences.

Given a sentence $a=[w[0],w[1],...w[n-1],w[n]]$ with N words, a confusion matrix $A\in R^{NXN}$ can be constructed.

TABLE I
CONSTRUCTION OF THE CONFUSION MATRIX NXN FOR A GIVEN SENTENCE
$a=[w[0],w[1],...w[n-1],w[n]]$

WORD	**w[0]**	**w[1]**	**.......**	**w[n]**
w[0]	P[0,0]	P[1,0]	....	P[n,0]
w[1]	P[0,1]	P[1,1]	....	P[n,1]
.	.	.		.
.	.	.		.
.	.	.		.
w[n]	P[0,n]	P[1,n]	....	P[n,n]

The size of the matrix depends on the length of the sentence. The objective of this confusion matrix is to extract the valid bigrams according to the language model. The element $P[i,j]$ indicates the validness of each pair of words $(w[i]w[j])$ according to the list of language model’s bigrams. If a pair of two words $(w[i]w[j])$ cannot be found in the list of language model bigrams then the corresponding $P[i,j]$ is taken equal to 0 otherwise it is equal to one. Hereafter, the pair of words with $P[i,j]$ equals to 1 is called as valid bigram. Note that, the number of valid bigrams is M lower than the size of the confusion matrix which is N^2, since all possible pairs of words are not valid according to the language model. In order to generate permuted sentences using the valid bigrams all the possible words’ sequence must be found. This is the search problem and its solution is the domain of this filtering process.

As with all the search problems there are many approaches. In this paper a left to right approach is used. To understand how it works the permutation filtering process, imagine a network of N layers with N states. The factor N concerns the number of sentence’s words. Each layer corresponds to a position in the sentence. Each state is a possible word. All the states on layer 1 are then connected to all possible states on the second layer and so on according to the language model. The connection between two states (i,j) of neighboring layers

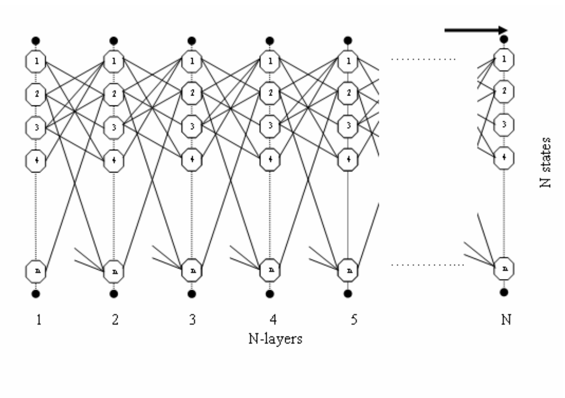

Fig. 2 Illustration of the lattice with N-layers and N states

$(N-1, N)$ exists when the bigram $\left(w[i]w[j]\right)$ is valid. This network effectively visualizes the algorithm to obtain the permutations. Starting from any state in layer 1 and moving forward through all the available connections to the N-th layer of the network, all the possible permutations can be obtained. No state should be “visited” twice in this movement.

V. Searching Valid Trigrams

The prime function of this approach is to decompose any input sentence into a set of trigrams. To do so, a block of words is selected. In order to extract the trigrams of the input sentence, the size of each block is typically set to 3 words, and blocks are normally overlapped by two words. Therefore, an input sentence of length N, includes N-2 trigrams. The second step of this method involves the search for valid trigrams for each sentence. In the third step of this method the number of valid trigrams per each permuted sentence is calculated. Considering that the sentence with no word-order errors has the maximum number of valid trigrams, it is expected that any other permuted sentence will have less valid trigrams. Although some of the sentence’s trigrams may be typically correct, it is possible not to be included into the list of LM’s trigrams. The plethora of LM’s trigrams relies on the quality of corpus. The lack of these valid trigrams does not affect the performance of the method since the corresponding trigrams of the permuted sentence will not be included into LM as well. The criterion for ranking all the permuted sentences is the number of valid trigrams. The system provides as an output, a sentence with the maximum number of valid trigrams. In case where two or more sentences have the same number of valid trigrams a new distance metric should be defined. This distance metric is based on the total probability of the trigrams. The total probability is computed by adding the probability of each trigram, whereas the probability of non valid trigrams is assigned to zero. Therefore the sentence with the maximum probability is the the system’s response.

VI. Experimentation

A. Experimental Scheme

The experimentation involves a test set of 100 sentences of 847 words. These sentences have been selected randomly from the section “Structure” of TOEFL past exams [11],[12]. The TOEFL test refers to the Test of English as a Foreign Language. The TOEFL program is designed to measure the ability of non-native speakers to read, write and understand English as used at college and university in North America. The Structure section focuses on recognizing vocabulary, grammar and proper usage of standard written English. There are two types of questions in the Structure section of the TOEFL test [13]. One question type presents candidates with a sentence containing a blank line. Test-takers must choose a word or phrase that appropriately fills in the blank. The other question type consists of complete sentences with four separate underlined words. Candidates must choose which of the four underlined answer choices contains an error in grammar or usage. For experimental purposes our test set consists of sentences for TOEFL’s word order practice. These sentences are selected from the list of the answer choices but are not the correct ones. Note that the test sentences are not included into the training set of the statistical language model that is used as tool for the proposed method. The goal of the experimental scheme is to confirm that the outcome of the method (sentence with best score) is the TOEFL’s correct answer. It is shown that the corpus contains sentences of length between 4 and 12 words.

B. Results

The evaluation of this method was conducted by comparing the output of the system with the correct answer choice that is indicated by TOEFL. The findings from the experimentation show that 93 sentences (93% in total) have been repaired using the proposed method (True Corrections).

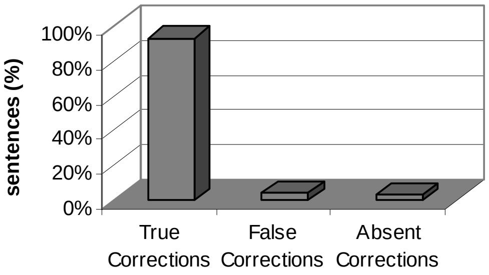

Fig. 3 The percentage of sentences with True, False and Absent corrections

On the other hand, the result for 4 sentences (4% in total) were false (False Corrections) and for 3 sentences (3% in total) the system was unable to rank the sentences (all scores were very close), (Absent Corrections). In case of "False Corrections" the system's response is different from the correct sentence. The incorrect output of the system can be explained considering that some TOEFL words are not included into the BNC vocabulary, hence some of the sentences' trigrams are considered as invalid.

The number of permutations that are extracted with the filtering process is significantly lower than the corresponding value without filtering, especially for large sentences. For sentences with length up to 8 words, the number of permutations is slightly lower when the filtering process is used, while for sentences with length greater than 8 words the filtering process provides a drastical reduction of permutations. It is obvious that the performance of filtering process depends mainly on the number of valid bigrams. This implies that the language model's reliability affects the outcome of the system and especially of the filtering process.

TABLE II
THE MEAN VALUE OF PERMUTATIONS FOR TOEFL SENTENCES

words	No filtering	With filtering
7	5040	968
8	40320	6293
9	362880	57890
10	3628800	429600
11	39916800	3127840
12	479001600	11378400

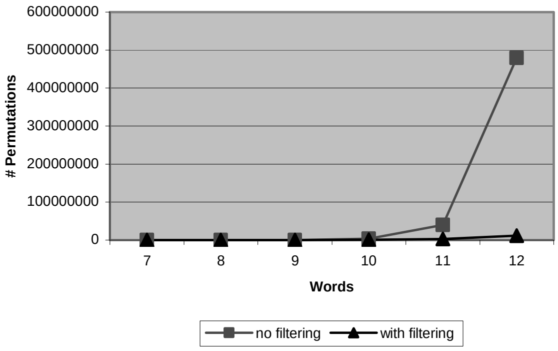

Fig. 4 The number of permutations with and without filtering in logarithmic scale for sentences with length from 5 to 12. The symbol (◊) denotes the number of the sentence's permutations without filtering while the symbol (▲) presents the number of the permutations extracted from the filtering method.

VII. Conclusions

The findings show that most of the sentences can be repaired by this method independently from the sentence's length and the type of word order errors. By the permutation's filtering process, the system takes advantage of better performance, rapid response and smaller computational space. One of the key questions is whether the use of language model can correct other grammatical errors such as subject- verb disagreement. The issue certainly invites research.

References

[1] E.S., Atwell, How to detect grammatical errors in a text without parsing it. In Proceedings of the 3rd EACL, 38–45, 1987.

[2] A., Golding, A Bayesian hybrid for context-sensitive spelling correction. Proceedings of the 3rd Workshop on Very Large Corpora, 39--53. 1995

[3] M.,Chodorow, C., Leacock. An unsupervised method for detecting grammatical errors. In Proceedings of NAACL'00, 140–147. 2000.

[4] T. Heift, Designed Intelligence: A Language Teacher Model, Unpublished Ph.D. Dissertation, Simon Fraser University,1998

[5] T. Heift, Intelligent Language Tutoring Systems for Grammar Practice. Zeitschrift für Interkulturellen Fremdsprachenunterricht (Online), 6 (2), 15 pp. 2001

[6] J., Bigert, O., Knutsson. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In Proceedings of Robust Methods in Analysis of Natural language Data, (ROMAND 2002), 10–19, 2002.

[7] J., Sjöbergh, Chunking: an unsupervised method to find errors in text, Proceedings of the 15th Nordic Conference of Computational Linguistics, NODALIDA 2005, 2005

[8] S.J., Young,. Large Vocabulary Continuous Speech Recognition, IEEE Signal Processing Magazine 13, (5), 45-57, 1996.

[9] I.J., Good, The population frequencies of species and the estimation of population parameters. Biometrika, 40(3 and 4):237-264, 1953.

[10] S.M., Katz, Estimation of probabilities from sparse data for the language model component of a speech recogniser. IEEE Transactions on Acoustics, Speech and Signal Processing, 35(3):400-401, 1987

[11] K.S., Folse, Intermediate TOEFL Test Practices (rev. ed.). Ann Arbor, MI: The University of Michigan Press, 1997.

[12] C. M., Feyton, Teaching ESL/EFL with the internet. Merill Prentice-Hall, 2002.

[13] J. A., Hawkins, A Performance Theory of Order and Constituency. Cambridge, Cambridge University Press, 1994.