Statistical Machine Translation as a Grammar Checker for Persian Language

Nava Ehsan, Heshaam Faili
Department of Electrical and Computer Engineering, University of Tehran
Tehran, Iran
{n.ehsan@ece.ut.ac.ir, hfaili@ut.ac.ir}

***Abstract*—Existence of automatic writing assistance tools such as spell and grammar checker/corrector can help in increasing electronic texts with higher quality by removing noises and cleaning the sentences. Different kinds of errors in a text can be categorized into spelling, grammatical and real-word errors. In this article, the concepts of an automatic grammar checker for Persian (Farsi) language, is explained. A statistical grammar checker based on phrasal statistical machine translation (SMT) framework is proposed and a hybrid model is suggested by merging it with an existing rule-based grammar checker. The results indicate that these two approaches are complimentary in detecting and correcting syntactic errors, although statistical approach is able to correct more probable errors. The state-of-the-art results on Persian grammar checking are achieved by using the hybrid model. The obtained recall is about 0.5 for correction and about 0.57 for detection with precision about 0.63.**

***Index Terms*—Natural Language Processing, Syntactic Error, Statistical Machine Translation, Grammar Checker, Persian Language**

I. Introduction

Proofreading tools for automatic detection and correction of erroneous sentences are one of the most widely used tools within natural language applications such as text editing, optical character recognition (OCR), machine translation (MT) and question answering systems [1]. The editorial assistance tools are useful in helping second language learners not only in writing but also in learning a language by providing valuable feedbacks [2]. Kukich [3] has categorized the errors of a text into five groups, 1. Isolated, 2. Non isolated or syntactic errors, 3. Real-word errors, 4. Discourse structure, and 5. Pragmatic errors. The first category refers to spelling errors. Detecting errors of second and third categories need syntactic as well as semantic analysis. The last two hierarchies cannot be considered as spelling or grammatical error. In this article we just focus on correcting syntactic errors and presuppose that the text is spell checked correctly. This paper is going to describe a statistical grammar checker approach within the framework of phrasal statistical machine translation. SMT has the potential to solve some kind of errors occurring in the sentences [4]. We will show that training statistical model would be helpful in detecting and correcting grammatical errors which were not addressed in the rule-based grammar checker [5] especially those errors which need contextual cues for recognition. We will also introduce a hybrid of statistical and rule-based approaches for grammar checking and achieved the state-of-the-art results on Persian grammar checking. Grammar checkers cannot check the whole syntactic structure of the text [6]. In the proposed model, frequently occurred error types have been identified for evaluating both error detection and correction of the system in terms of precision and recall metrics.

The remainder of the paper is organized as follows: Section 2 outlines related works of grammar checking. In Section 3, the limitation of previous Persian grammar checker is discussed. Section 4 describes the use of SMT framework for grammar checking followed by preparing training and test data set. Finally, the evaluation results for each approach individually and the hybrid model are reported in Sections 5 and 6, respectively.

II. Related Work

Grammar checkers deal with syntactic errors in the text such as subject-verb disagreement and word order errors. Grammar checking entails several techniques from the NLP research area such as tokenization, part-of-speech tagging, determining the dependency between words or phrases and defining and matching grammatical rules. Grammar checking techniques are categorized into three groups: syntax-based, statistical or corpus-based and rule-based [2]. In syntax-based approach the text is parsed and if parsing does not succeed the text is considered as incorrect. It requires a complete grammar or mal-rules or relaxing constraints which are obviously difficult to obtain due to complex nature of natural languages. Mal-rules allow the parsing of specific errors in the input and relaxing constraints redefine unification so that the parse does not fail when two elements do not unify [2]. The existing grammar checkers [7][8] fall into rule-based category in which a collection of rules describe the errors of the text, while [9][10][11] use statistical analysis for grammar checking. Although, rule-based grammar checkers have been shown to be effective in detecting some class of grammatical errors, manual design and refinement of rules are difficult and time-consuming tasks. Deep understanding of the linguistics is required to write non-conflicting rules which cover a suitable variety of grammatical errors. Although there have been some prior works on Persian spell checking, [12][13] from the best of our knowledge, the only work on Persian grammar checking is a rule-based system which is introduced in [5]. The limitations of this grammar checker are described in detail in next section.The ALEK system

ISBN: 978-1-61208-139-7

developed by [9] uses an unsupervised method for detecting English grammatical errors by using negative evidence from edited textual corpora. It uses TOEFL essays as its resource. Integrating pattern discovery with supervised learning model is proposed by [11]. A generation-based approach for grammar correction is introduced by [10] which checks the fluency of sentences produced by second language learners. The N-best candidates are generated using n-gram language model which are reranked by parsing using stochastic context-free grammar. A pilot study of [4] presents the use of phrasal SMT for identifying and correcting writing errors made by learners of English as a second language and the focus was on countability errors associated with mass nouns. The statistical phase of grammar checking procedure introduced in this paper also relies on phrasal SMT framework for detecting and correcting syntactic errors. To overcome the negative impact of some types of errors on recall metric, the system is augmented with the rule-based procedure.

III. Limitations of Persian Rule-based Grammar Checker

The proposed rule-based grammar checker [5] faces some limitations. It is based on regular expression patterns and detects errors which can be matched by regular expressions, thus it cannot detect those patterns which are difficult or impossible to be modeled by regular expressions. The other problem is having pre-defined pattern and suggestion for each type of error. For example whenever it detects two repeated words, it shows an error although not all two repeated words are incorrect and one of them is deleted due to pre-defined suggestion. Our method is an SMT-based approach which does not follow any specific pre-defined rule or suggestion. For example by detecting repeated words, in some cases one of the words may be eliminated or sometimes a preposition is added between duplicate words and in some cases it does not recognize any error. In addition regular expressions cannot detect any recursive pattern. The errors which need context free grammar or statistical or semantic analysis or disambiguation are also undetectable by regular expressions. Existing techniques for Persian, based on hand-crafted rules or statistical POS tag sequences [5] are not strong enough to tackle the common incorrect preposition or conjunction omission errors due to lack of information about language model. In our experiments not only all the syntactic errors described in the rule-based grammar checker are included but also, the following errors are added. The errors are illustrated with an example.

(1) Omission of prepositions: Some words need special prepositions to complete their meanings. Prepositions depend on nouns and can complement the other words. Since the lexical information is important to correct omission of prepositions, we need to define large number of rules which include all the phrases containing prepositions and the words around it. Thus, it is not feasible to define the patterns by regular expressions. For example, بحث سر تفاوتها است/ *bahs sar tafaavotha ast* (the discussion is differences) should be corrected as بحث بر سر تفاوتها است/ *bahs bar sar tafaavotha ast* (the discussion is about differences).

(2) Omission of را/ *ra* (definite object sign): Object is a mandatory argument of transitive verbs. The meaning of transitive verb is incomplete and unclear without the object. The direct object should be addressed in the sentence by preposition را/ *ra*. Finding the object of the sentence requires semantic analysis and it cannot be detected by regular expressions. Since this is an important preposition, the rule has been considered separately. For example, کار شروع کردید/ *kar shooroo kardid* (you started work) should be corrected as کار را شروع کردید/ *kar ra shooroo kardid* (you started the work).

(3) Omission of conjunctions: The omission of conjunctions is not always incorrect, but there are some cases that the omission makes the sentence grammatically wrong. This usually happens when a clause appears in the middle of the sentence. Lexical information is also important in this case. For example, تعریفی خود شما دارید چیست/ *tarifi khod shoma darid chist* (what is the description you have) should be converted to تعریفی که خود شما دارید چیست/ *tarifi ke khod shoma darid chist* (what is your description).

(4) Using indefinite noun when a demonstrative pronoun is used: Demonstrative pronouns are independent words that precede the noun. After demonstrative pronouns such as این/ *in* (this) and آن/ *an* (that) a definite noun should be used, unless a description is given within a phrase, like آن کتابی که خواندم/ *an ketaabi ke khaandam* (the book that I have read). Since regular expressions cannot identify to which word of the sentence the descriptive phrase belongs, defining this rule with regular expression may result in many false alarms. For example,آن کتابی را خواندم/ *an ketaabi ra khaandam* (I read that a book) should be changed to آن کتاب را خواندم/ *an ketaab ra khaandam* (I read that book).

(5) Connecting indefinite postfix to the first noun in possessive nouns (ezafe construction [14]): Persian is a dependent-marking language [15] and tends to mark the relation on the non-head. In case of having possessive nouns indefinite postfix ی/ *i* should be connected to the last word. The postfix ی/ *i*, is not only used as indefinite sign, but also can be used as a copula for the second singular person like آزادی/ *azaadi* (you are free) and it may also belong to its own word like آزادی/ *azaadi* (freedom). The morpheme ی/ *i* is used in forming various lexical elements in derivational morphology and will cause ambiguities [14]. Since regular expressions deal with the surface of the word without semantic analysis these ambiguities are not distinguishable by regular expressions. For example, کتابی داستان/ *ketaabi daastaan* should be corrected as کتاب داستانی/ *ketaab*

daastaani (a story book).

(6) Using adjective before noun: In Persian, adjectives usually follow the nouns. This rule was omitted from the rule-based grammar checker [5] according to its low precision. Adjectives can sometimes stand as the adverb of the sentence and they can precede the noun. The part-of-speech tagger used in the rule-based grammar checker could make mistakes in recognizing adjectives. Also there is no chunking process before applying regular expressions, thus the rule-based system cannot understand if the adjective belongs to the same phrase as nouns or not. For example, جالب کتاب/ *jaaleb kettab* (book interesting) should be converted to کتاب جالب/ *ketaab jaleb* (interesting book) but دانشمند مرد را دید/ *daaneshmand mard ra did* (scientist saw the man) is correct.

(7) Using wrong plural morpheme: Morphologically, Persian falls into polysynthetic languages in which a single word may have many morphemes and also several morphemes exist for marking plurality in Persian, like ان/ *an*, ها/ *ha* , یون/ *yun* and ات/ *at* [16]. Some words like درخت / *derakht* (tree) can become plural with ان/ *an* and ها/ *ha*. Both words درختان/ *derakhtan* (trees) and درختها/ *derakhtha* (trees) are correct but, some words like انسان/ *ensaan* (human) or میز/ *miz* (table) can become plural with ها/ *ha* but not with ان/ *an* and unfortunately there is no special rule for that and it cannot be defined by regular expressions. For example, انسان + ان (انسانان) *ensaanan* should be converted to انسانها/ *ensaanha* (humans).

In brief, the mentioned errors can be detected either by using probabilistic context free grammar (PCFG) as a modeling formalism, or by using statistical or semantic analysis or by defining too many lexical rules. Due to the discussed limitations of regular expressions, we used a statistical approach based on SMT framework for grammar checking to overcome the problems of the existing rule-based grammar checker.

IV. SMT Framework for Grammar Checking

Machine translation refers to usage of computer to automate some or all of the process of translating from one language into another [17]. Automatic grammar checking is modeled as a machine translation where the erroneous sentence is translated to the correct sentence. Machine translation is considered as a hard task [15] in general due to differences between the languages which are referred to as translation divergences [15]. Translation divergence could be structurally, like differences in morphology, argument structure, ordering, referential density and linking of predicates with their arguments or it could be lexically like homonymous, polysemy, many-to-many translation mappings and lexical gaps. The less divergence between source and target languages leads to better translation. Unlike machine translators that the input sentence belongs to a language other than output sentence and could have many differences structurally and lexically, in the proposed model for grammar checking both input and output sentences belong to the same language except the input sentence has some syntactic errors. As it will be described later the syntactic errors considered in this paper could cause lexical gap or divergences in morphology, argument structure or ordering, between source and target sentences. Stylistic and cultural differences which are another source of difficulty for translators do not appear in this model. The noisy channel model is the foundation of statistical machine translation [18]. In this article we explore its application to grammar checking. The noisy channel is used whenever the received signal does not identify the sent message. Grammar checking could be modeled as a noisy channel where the intended message is the correct sentence while the received signal is the erroneous sentence. We assume grammar checking from an incorrect sentence to correct sentence. The suggested correct sentence is the one whose probability is the highest:

$$\hat{C} = \arg\max_{c} P(C|E) = \arg\max_{c} \frac{P(E|C)P(C)}{P(E)} \quad (1)$$

The probability in the denominator of equation 1 is ignored since we are choosing the correct sentence for a fixed erroneous sentence, thus is a constant. Equation 1 shows that we need to compute $P(E|C)$ and the language model $P(C)$. We assume that the noisy sentence is the result of applying syntactic errors on the correct sentence. The syntactic error rules considered in this paper are classified in Table I and we refer to them as $R = r_1, r_2, ..., r_n$. The relationship can be expressed as follows:

$$P(E|C) = \sum_{i=1}^{n} P(E, r_i|C) \quad (2)$$

The conditional probability $P(E|C)$ is computed as follows:

$$P(E|C) = \sum_{i=1}^{n} P(E|C, r_i) * P(r_i|C) \quad (3)$$

Two assumptions have been made, although we will later show in our experiments that these assumptions will not really affect the system's accuracy regarding to precision and recall metrics.

(1) Each sentence could just have one syntactic error. In other words, just one of the error rules could be applied on the sentence.

(2) The condition probability $P(r_i|C)$ has uniform distribution. It means that each rule is equally likely to be applied on a correct sentence. That is, the probability of appearing each error mentioned in Table I, is the same.

Thus, equation 1 is defined as follows:

$$\hat{C} = \arg\max_{c} P(C|E) = \arg\max_{c} P(E|C, r_i)P(C) \quad (4)$$

where r_i is the error rule which was applied on the correct sentence, (C).

TABLE I
PERSIAN SYNTACTIC ERROR RULES

ID	Error description
1	Omission of preposition
2	Omission of conjunction
3	Using plural noun after cardinal numbers
4	Using a verb or preposition after a genitive noun ending with sign ی/ *i*
5	Using a verb before copulative verbs
6	Using a superlative adjective before preposition از/ *az* (than)
7	Using a preposition or conjunction at the end of the sentence
8	Omission of را/ *ra* (definite object sign)
9	Using را/ *ra* (definite object sign) after verb or preposition or in the beginning of the sentence
10	Using two consecutive adverbs of question or pronouns without و/ *va* (and)
11	Double plural noun
12	Using adjective before noun
13	Disagreement between the subject and the verb
14	Repeating a word
15	Using wrong plural morpheme
16	Using indefinite noun when a demonstrative pronoun is used
17	Connecting indefinite postfix to the first noun in ezafe constructions

In order to use the phrase-based translation model, we need a training data. Training data construction is described in detail later in this paper. Here, we just mention that we have parallel corpora of correct and erroneous sentences in which correct sentences are infected by one of the error rules to produce the erroneous sentences. If more than one rule are applicable on a sentence, separate sentences are produced each containing only one error. Phrasal translation model uses phrases as well as single words as the fundamental units. Phrase translation probability, $\phi(\overline{e_i}, \overline{c_i})$, is defined as the probability of generating phrase $\overline{c_i}$ from incorrect phrase $\overline{e_i}$. Distortion refers to a word having a different position in the input and output sentences. The more reordering the more expensive is the translation. The distortion is parameterized by $d(a_i - b_{i-1})$, where a_i is the start position of the erroneous phrase generated by the i-th correct phrase, and b_{i-1} is the end position generated by the $(i-1)$-th correct phrase. The phrase translation probability and distortion probability are computed as follows [15]:

$$\phi(\overline{e}, \overline{c}) = \frac{count(\overline{e}, \overline{c})}{\sum_{\overline{e}} count(\overline{e}, \overline{c})} \tag{5}$$

$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|} \tag{6}$$

where, α is a small constant.

Similar to the translation model for statistical phrasal MT the conditional probability in equation 1 is decomposed into:

$$P(E|C) = \prod_{i=1}^{I} \phi(\overline{e_i}, \overline{c_i}) d(a_i - b_{i-1}) \tag{7}$$

Moses [17] is a statistical machine translation system for automatically training translation models and decoding for any language pair in which GIZA++ [19] is used for word-alignments and SRILM [20] is used as the language model toolkit. As in phrase-based models, factored translation model which is used in Moses can be seen as the combination of several features $h_i(C|E)$. These features are combined in a log-linear model. If there are N features, then the log-linear translation model is:

$$P(C|E) = \frac{1}{Z} \prod_{i=1}^{N} \alpha_i^{h_i(C|E)} \tag{8}$$

Here, Z is a normalizing constant and α_i is the weight assigned to feature In practice, the noisy channel model factors (the language model $P(C)$ and translation model $P(E|C)$), are still the most important feature functions in the log-linear model, but the architecture has the advantage of allowing for arbitrary other features as well [15]. If two weights and features are used, and set them relative to the language model $P(C)$ and the conditional probability $P(E|C)$ as follows:

$$h_1(C|E) = \log_{\alpha_1}^{P(C)} \tag{9}$$

$$h_2(C|E) = \log_{\alpha_2}^{P(E|C)} \tag{10}$$

We can see by replacing equations 9 and 10 in equation 8 that the fundamental equation 1 is the special case of the log-linear model [21]. Language model is used to score the fluency of the output. Phrase translation table saves the extracted phrases. The construction of phrase table and learning phrases are described in [22]. Distortion model allows reordering of the input sentence, but at a cost. The main features of Moses are distortion model, language model, translation model and word penalty [22].

A. Preparing Training and Test Data

Training data is a collection of aligned sentences in two files, one for ungrammatical sentences and one for correct sentences. In order to prepare the erroneous corpus, the set of specific error types mentioned in Table I are used. These errors include all the patterns defined in [5] and those defined in Section 3. A Persian part-of-speech tagged corpus named Peykareh [23] containing collection of formal news and common well-formed texts is used. A set of example sentences for each of various error types should be collected, thus, the error rules are injected to the sentences of this corpus automatically. The underlying assumption that each sentence has one syntactic error has to be held. If more than one type of error are possible in a sentence, erroneous sentences are made each containing only one type of error and the corresponding correct sentence is placed for each sentence in another file. The sentences larger than 25 words are pruned to make the training phase more practical. The prepared training set contains about 340,000 erroneous sentences; each corresponds to a correct sentence in another file. The language model is also created from the correct corpus by SRILM toolkit [20]. We used some parts of Peykareh, other than those used in training phase, as test set, to evaluate the result of the system. For each type of

TABLE II
CATEGORIES OF DEFINED ERROR RULES

Category	Rule number
An unnecessary word	7,9,14
A missing word	1,2,4,7,8,9,10
A word or phrase that needs replacing	9,12
A word used in the wrong form	3,4,5,6,11,13,15,16,17

error mentioned in Table I, a set of 20 samples are injected in the test set. These sentences are considered as the input of the system. The results are illustrated in Figure 1. In each case if the created error leaves the sentence meaningful, that sentence is not evaluated. Thus, the assessment set are those sentences that contain real grammatical errors which contains 321 erroneous sentences. Dealing with null subject and pro-drop feature of Persian language, error number 13 the subject-verb disagreement error, is examined when the subject is a pronoun and it is not dropped.

V. EXPERIMENTAL RESULTS

Error rules are classified and results are evaluated for each rule individually. Correction is referred to those sentences which the output is a correct sentence and detection is referred to those sentences which system made changes to the input sentence, in the scope of the error, but the suggested change may not be correct. The error rules are those shown in Table I. Nicholls [24] identifies four error types: an unnecessary word, a missing word, a word or phrase that needs replacing and a word used in the wrong form. Table II shows that to which categories our defined error rules belong, the error rules are referred by their number defined in Table I. As shown in Table II some error rules could belong to more than one category. The first and second category cause lexical gap between the erroneous and correct sentences, the third category causes ordering differences and the category fourth causes morphological differences while error number 17 will cause difference in argument structure which emphasizes on dependant marking feature of the language. A similar classification is also introduced by [25] which does not contain the third category and includes a separate category for agreement errors. The recall results of 20 samples of each error rule are demonstrated in Figure 1. The horizontal numbers in Figure 1 indicate the error numbers described in Table I.

The recall of the system was 0.44 for correction and 0.48 for detection with precision 0.61. The results of Figure 1 indicate that this approach is successful in detecting some rules which were not detectable by regular expressions as discussed before, rules number 1,2,8,12,15,16 and 17, on the other hand it cannot detect the rules number 4,7 and 9 which are detectable by regular expressions. These rules are those that could be belonged to different categories. Since there could be different ways for correcting these errors, it seems that the translation probabilities of possible solutions are distributed in the training set. The rule-based grammar checker proposed in [5] is tested on this test set which the recall results are demonstrated in Figure 2. Here, detection means flagging the error with or without suggestion. The recall was 0.23 for correction and 0.35 for detection on the defined rules with precision 0.94.

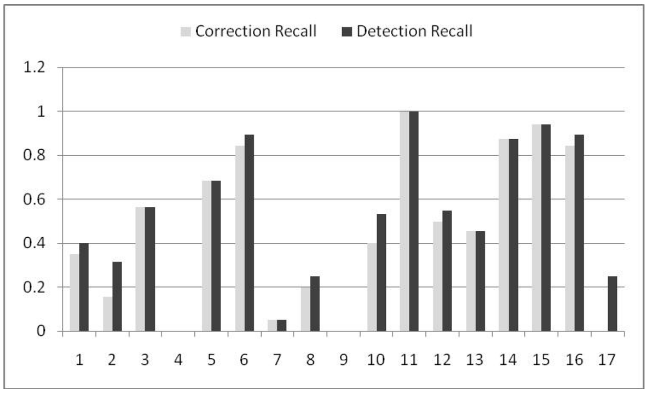

Fig. 1. Results for SMT-based Grammar Checker

VI. COMBINING SMT AND RULE-BASED GRAMMAR CHECKER

The proposed SMT procedure performed as an error corrector. It does not contain error detection in first stage and all sentences are regarded as erroneous for correction. The error detection defined in previous section actually refers to incorrect correction. Some rules have negative impact on recall of the SMT-based technique which are detectable by rule-based approach. We would expect to see a greater improvement by combining these two techniques. In this case, errors are detected either by SMT-based or rule-based grammar checker or both. If the correction differs in two systems, both could be shown to the user as suggestions. If one of the suggestions is correct we assume that the system corrected the sentence. The results are illustrated in Figures 3 and 4 for detection and correction recall respectively. The recall improved to 0.53 for error correction and 0.66 for error detection while the precision was 0.67.

VII. DISCUSSION

In the previous experiments, we relied on the assumption that each error is equally likely to be applied on a correct sentence and in the test set the errors were uniformly distributed. The likelihood occurrences of the errors were not considered.

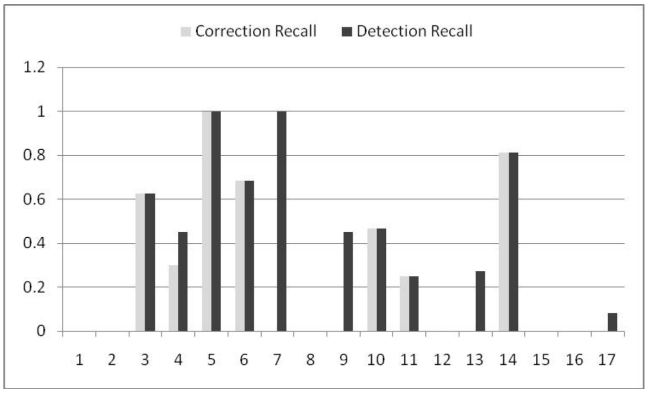

Fig. 2. Results for Rule-based Grammar Checker

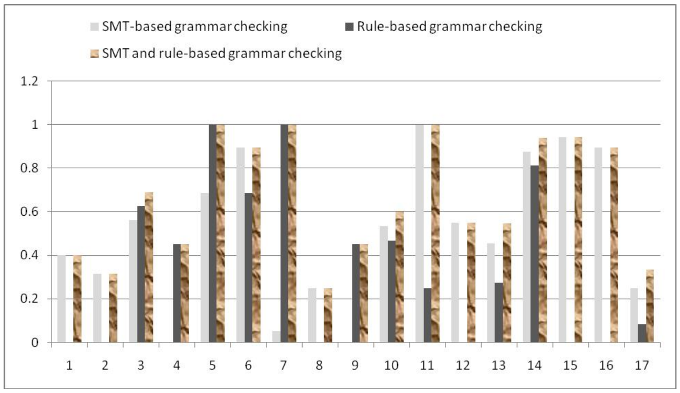

Fig. 3. Combining SMT and rule-based results for grammar detection

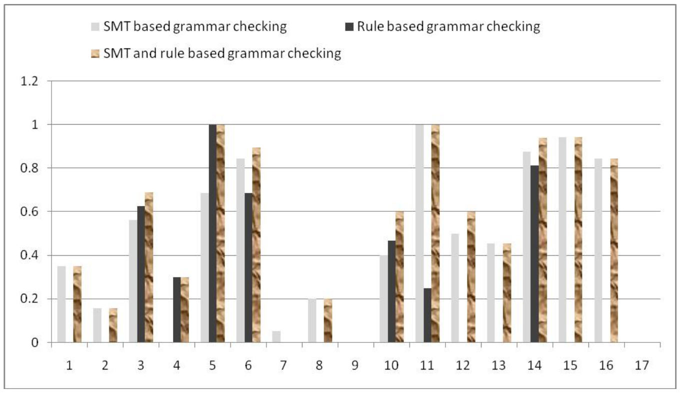

Fig. 4. Combining SMT and rule-based results for grammar correction

In the real world some types of errors are more probable to happen than the others. The conditional probability $P(r_i|C)$ is defined as the probability of making the error r_i when it is applicable to a correct sentence. Although there are learner corpora for some languages like ICLE (Cambridge Learner Corpus) and JLE (Japanese Learners of English Corpus) that contain annotated errors, there is no such annotated corpora available to date for Persian language to compute the probability occurrence of each type of error. The new method to process the free resource of revision histories of Wikipedia to create error corpora, have shown in experiments that even large revision histories contain rather scarce information about errors [26]. In this survey we asked from 19 native speakers the probability of making each type of error in the texts. The answers were classified to high, medium and low. Computing the weighted average of the answers the likelihood occurrence of each error in the texts is estimated. The weights are 80, 50 and 20 percent for high, medium and low classes respectively. The results are given in Table III. The information given in Table III indicates that those errors which could not be recognized by SMT-based grammar checker are the less probable errors of the language.

TABLE III
OCCURRENCE PROBABILITY FOR EACH TYPE OF ERROR

Rule Number	Probability (%)
1	67.36
2	35.78
3	34.21
4	35.78
5	40.52
6	27.89
7	31.05
8	40.52
9	26.31
10	51.57
11	61.05
12	26.31
13	43.68
14	43.68
15	48.42
16	37.36
17	45.26

Similar to the work of [27] which appended the difficult-to-translate phrases with human translations to the training set to reduce the negative impact of these phrases, this time we made a training set by considering the occurrence probability of errors. If an error was applicable to a sentence it is injected regarding to the relevant probability which results in about 220,000 pair sentences. We refer to our previous train set as train set1 and to our newly produced train set as train set2 which will result in statistical grammar checker1 (SGC1) and statistical grammar checker2 (SGC2). In order to test the results, another experiment is done on 500 erroneous sentences from the test set in which we used the likelihood occurrence information of errors where the error is applicable to the sentence. This test set is evaluated with both SGC1 and SGC2. The language model is same for both models. We refer to this test set as probable test set. The all results are summarized in Table IV. In order to consider the importance of precision for grammar checker [28] we have also evaluated the $F_{0.5}$ measure to weight precision twice as much as recall. In the test sets used so far, all sentences contained just one grammatical error. In order to realize that whether the existence of more than one error would affect the grammar checking process, 20 sentences are tested each containing more than one type of error. The results indicated that the grammar checker may not be able to recognize an error (the error which was previously recognized) if two types of errors happened in the same phrase.

VIII. CONCLUSION

This paper proposed a hybrid model of statistical and rule-based approaches for identifying grammatical errors for Persian language. The statistical part is based on phrasal SMT and the principles are language independent. The studies show that employing SMT framework for grammar checking has the ability to correct some class of errors which are the most probable errors in the sentences. To overcome the negative impact of some types of errors on the recall metric, the system is augmented with the rule-based procedure. The obtained recall was 0.53 for error correction and 0.66 for error detection while the resulted precision is 0.67 without considering the likelihood of occurrences of errors in the text. The likelihood of occurrences of each error type is estimated to be able to

TABLE IV
SUMMARIZED RESULTS OF GRAMMAR CHECKERS

		Correction recall	Detection recall	Precision	F-1	F-0.5
Uniform test set	SGC1	0.44	0.48	0.61	0.53	0.566
	Rule-based grammar checker	0.23	0.35	0.94	0.51	0.581
	SGC1 + Rule-based grammar checker	0.53	0.66	0.67	0.66	0.636
Probable test set	SGC1	0.46	0.5	0.61	0.54	0.572
	SGC2	0.48	0.5	0.62	0.55	0.585
	Rule-based grammar checker	0.25	0.31	0.91	0.46	0.595
	SGC2 + Rule-based grammar checker	0.5	0.57	0.63	0.59	0.599

evaluate the grammar checker more accurately. In this case, the obtained recall is 0.5 and 0.57 with augmentation of rule-based approach. This is the state-of-the-art results on Persian grammar checking so far.

IX. FUTURE WORKS

There are still number of tasks to improve the grammar checking system. We would like to collect grammatical errors from non-native learners which allow us to expand the grammar checker to better distinguish correct and erroneous sentences for language learners. It can also help to find better training examples for the system. Some errors in the sentence are result of real word errors. The SMT-based framework seems to be able to detect one word among the sentence that does not fit. The real-word error detection is going to be tested with this approach. Since the statistical approach is language independent it can be trained and tested on the other languages such as English, considering the errors of the language.

REFERENCES

[1] M. Bhagat, “Spelling error pattern analysis of punjabi typed text,” Master’s thesis, Computer Science and Engineering Department Thapar Institute of Engineering and Technology Deemed University Patiala, 2007.

[2] C. Leacock, M. Chodorow, M. Gamon, and J. Tetreault, “Automated grammatical error detection for language learners,” *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–134, 2010.

[3] K. Kukich, “Techniques for automatically correcting words in text,” *ACM Computing Surveys (CSUR)*, vol. 24, no. 4, pp. 377–439, 1992.

[4] C. Brockett, W. Dolan, and M. Gamon, “Correcting ESL errors using phrasal SMT techniques,” in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006, pp. 249–256.

[5] N. Ehsan and H. Faili, “Towards grammar checker development for Persian language,” in *in 6th IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE’10)*, 2010, pp. 150–157.

[6] D. Kies, “Evaluating Grammar Checkers: A Comparative Ten-Year Study,” in *in 6th International Conference on Education and Information Systems, Technologies and Applications: EISTA*, 2008.

[7] D. Naber, “A rule-based style and grammar checker,” Master’s thesis, Diplomarbeit.Technische Fakultat Universitat Bielefeld, 2003.

[8] F. Bustamante and F. León, “GramCheck: A grammar and style checker,” in *Proceedings of the 16th conference on Computational linguistics-Volume 1*, 1996, pp. 175–181.

[9] M. Chodorow and C. Leacock, “An unsupervised method for detecting grammatical errors,” in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, 2000, pp. 140–147.

[10] J. Lee and S. Seneff, “Automatic grammar correction for second-language learners,” in *Ninth International Conference on Spoken Language Processing*, 2006, pp. 1978–1981.

[11] G. Sun, X. Liu, G. Cong, M. Zhou, Z. Xiong, J. Lee, and C. Lin, “Detecting erroneous sentences using automatically mined sequential patterns,” in *Annual Meeting-Association for Computational Linguistics*, vol. 45, no. 1, 2007, pp. 81–88.

[12] M. Shamsfard, H. Jafari, and M. Ilbeygi, “Step-1: A set of fundamental tools for persian text processing,” in *In 8th Language Resources and Evaluation Conference*, 2010.

[13] O. Kashefi, M. Nasri and K. Kanani Towards Automatic Persian Spell Checking. *Tehran, Iran: SCICT*, 2010.

[14] K. Megerdoomian, “Finite-state morphological analysis of Persian,” in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, 2004, pp. 35–41.

[15] D. Jurafsky, J. Martin, A. Kehler, K. Vander Linden, and N. Ward, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* MIT Press, 2010, vol. 163.

[16] B. Sagot and G. Walther, “A morphological lexicon for the Persian language,” in *in Proceedings of the 7th Language Resources and Evaluation Conference (LREC’10)*, 2010, pp. 300–303.

[17] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 2007, pp. 177–180.

[18] P. Brown, V. Pietra, S. Pietra, and R. Mercer, “The mathematics of statistical machine translation: Parameter estimation,” *Computational linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[19] F. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Computational linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[20] A. Stolcke, “SRILM-an extensible language modeling toolkit,” in *Seventh International Conference on Spoken Language Processing*, vol. 3, 2002, pp. 901–904.

[21] A. Axelrod, “Factored Language Model for Statistical Machine Translation,” Master’s thesis, Institute for Communicating and Collaborative Systems, Division of Informatics, University of Edinburgh, 2006.

[22] P. Koehn, “MOSES, Statistical Machine Translation System, User Manual and Code Guide,” 2010.

[23] M. Bijankhan, “Naghshe Peykarehaye Zabani dar Neveshtane Dasture Zaban: Mo’arrefiye yek Narmafzare Rayane’i [the Role of Corpus in generating grammar: Presenting a computational software and Corpus],” *Iranian Linguistic Journal*, vol. 19, pp. 48–67, 2006.

[24] D. Nicholls, “The Cambridge Learner Corpus-error coding and analysis for lexicography and ELT,” in *Proceedings of the Corpus Linguistics 2003 conference*, 2003, pp. 572–581.

[25] J. Wagner, J. Foster, and J. Van Genabith, “A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors,” *Proceedings of EMNLP-CoNLL-2007*, 2007.

[26] M. Miłkowski, “Automated building of error corpora of Polish,” *Corpus Linguistics, Computer Tools, and ApplicationsState of the Art. PALC 2007*, pp. 631–639, 2008.

[27] B. Mohit and R. Hwa, “Localization of difficult-to-translate phrases,” in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, pp. 248–255.

[28] A. Arppe, “Developing a grammar checker for Swedish,” in *The 12th Nordic Conference of Computational Linguistics*, 2000, pp. 13–27.