Combining Methods for Detecting and Correcting Semantic Hidden Errors in Arabic Texts

Conference Paper · May 2007

DOI: 10.1007/978-3-540-70939-8_56 · Source: DBLP

3 authors, including:

Chiraz Ben Othmane Zribi
Université de la Manouba
46 PUBLICATIONS **144** CITATIONS
SEE PROFILE

Mohamed Ben Ahmed
Université de la Manouba
197 PUBLICATIONS **1,042** CITATIONS
SEE PROFILE

Combining Methods for Detecting and Correcting Semantic Hidden Errors in Arabic Texts

Chiraz Ben Othmane Zribi, Hanene Mejri, and Mohamed Ben Ahmed

RIADI laboratory, National School of Computer Sciences, 2010,
University of La Manouba, Tunisia
{Chiraz.benothmane,Hanene.mejri,Mohamed.benahmed}@riadi.rnu.tn

Abstract. In this paper, we address the problem of semantic hidden errors in Arabic texts. These are spelling errors occurring in valid words and causing semantic irregularities. We first expose the different types of these errors. Then, we present and argue the adopted approach, which is based on the combination of several methods. Next, we describe the context of our work and show the multi-agent architecture of our system. Finally we present the testing framework used to evaluate the implemented system.

1 Introduction

Hidden errors are spelling errors that occur in valid words. The presence of such a word within an incorrect syntactic or semantic (even pragmatic) context makes the whole sentence incomprehensible. For instance:

Example: تطلع الشّمس علينا من الشّوق (the sun shines from desire).

In this example, the writer intended to write *الشّرق* (east) not *الشّوق* (desire) but a typographical error yielded a sentence that does not make sense.

Statistics mentioned in [1] show that hidden errors count for 40% of all spelling errors. This high number demonstrates the need for studying this kind of errors. In Arabic this problem is much more present because of the proximity of words. According to [2] the probability to encounter a hidden error is 14 times larger than in English and 10 times larger than in French.

Several researchers have taken an interest in this problem. Golding studied this kind of errors for the English language and proposed multiple correction methods such as the Bayesian method [3], the Trigram-based method [4] and the Winnow method [5]. Chinese was also studied by [6]. Swedish was the subject of a similar study by [7]. Bolshakov et al. studied a type of semantic errors called malapropisms. They were first interested in English [8] and Russian [9]. Then they studied and proposed a solution for the detection and the correction of these kind of errors for Spanish [10] [11].

Even though Arabic has characteristics that increase the probability of such errors occurring, there is only one research that we carried out earlier [12]. This work was concerned by the problematic of hidden errors in general and by the syntactic level in

A. Gelbukh (Ed.): CICLing 2007, LNCS 4394, pp. 634 – 645, 2007.

particular. Thus, the problem of hidden errors in Arabic is not yet resolved, and we hope to have a part in the solution by proposing a system for the detection and the correction of semantic hidden errors occurring in Arabic texts.

Due to the complexity of the problem, we made some assumptions to restrict the scope of our investigation: we first did not take into account the vowel marks in words. This is argued by the fact that the majority of texts (except of the didactic ones) are without vowels in spite of the importance of these marks in the process of reading. Second we assumed that there is only one hidden error at the most per sentence and that error results from one elementary typographical error such as: character insertion, deletion, substitution or transposition. Statistics have showed that only one of theses operations are at the origin of a spelling error in 90% of all cases [13].

The remainder of this paper is organized as follows: First, we present the concept of semantic hidden errors. Then we present our approach for detecting and correcting these errors. Next we explain the MAS (Multi –agent System) architecture of the implemented system. Finally, we describe the method we used to evaluate the efficiency of our system and the obtained results.

2 Semantic Hidden Errors

We call “semantic hidden” error every single-consonant spelling error which results in a correctly-spelled, but semantically incorrect word in a given context. This class of typographic mistakes cannot be detected by a simple spell checker, which is concerned only by the erroneous spelled words.

Semantic hidden errors can cause incomprehensive sentences or unfinished sentences. In the first case, the sentence is misinterpreted or completely absurd. The second case concerns sentences having a partial or incomplete meaning. We take an interest in this work only in errors giving incomprehensive sentences.

يعرضون عليه أموالا كبيرة (كثيرة)
They give him big (much) money

In this erroneous sentence, the adjective كبيرة (big) takes the place of the correct word كثيرة (much) due to the substitution of ب by ث

3 Detecting Semantic Irregularities

To understand the meaning of a word, the computer (like human) must know the different representations of this word and its different contexts of use. This knowledge can be obtained by different resources as: thesaurus, ontologies, semantic networks or textual corpora.

In this work, we chose a method that obtains the words’ meaning from textual corpora. This direction is based on the principle of the distributional linguistic that stipulates: "*the word’s meaning can be determined statistically, from contexts (i.e., paragraphs, sentences, texts in which this word occurs* "[14]. For example, the word *plane* occurs often with words as: *take off*, *wing*, *airport*, and rarely with *lion* or *forest*.

For detecting semantic hidden errors, we propose to check the semantic validity of each word in the text. To this purpose we combine four methods (statistical and linguistic) making possible the representation of a word according its near and distant context and the comparison of this representation with the ones obtained from the textual corpora. The idea behind this combination is to profit from the advantages of each method. In addition, this involves the selection of only one error if there is a conflict. This decision is then taken by a process of voting that takes into account the results of the application of each method and chooses the most probable error.

A training phase is needed to obtain from the textual corpora all data which are used by the different proposed methods. These data are presented as linguistic information and statistical measures.

3.1 Co-occurrence-Collocation Method

This method verifies the contextual validity of a word by calculating its frequency of appearance in a given context using the following measures:

First, let:
$S = \{w_1 ,.., w_i,..., w_n\}$ be the input sentence to the semantic checker .
$L = \{l_1,..., l_i,..., l_n\}$ be the set of the words lemmas of the sentence.
$C = \{c_{-k}, ..., c_{-1}, c_{1,} ..., c_k\}$ be the set of k words surrounding the word to be analyzed.

- **Frequency of occurrence:** This frequency is calculated for each word w_i in the sentence to analyze, in a window of 10 words[1]. This is achieved by using Bayes' inversion formula:

$$p(w_i|C) = \frac{p(C|w_i) \times p(w_i)}{p(C)}. \quad (1)$$

The word w_i *i*s closer to its surrounding words as the value of $p(w_i|C)$ is higher.

- **Coefficient of collocation:** To determine this coefficient we first identify all collocations in a sentence by referring to a list of collocations obtained during the training phase. For this purpose, we used and adapted a part of the system accomplished by [15]. When a collocation is found in a sentence, a coefficient is given to each word in this expression. This coefficient is the Kulczinsky measure *(KUC)*, which is a criterion of association that identifies the degree of correlation between two lemmas l_i et l_j using the following formula:

$$KUC = \frac{a}{2}\left(\frac{1}{a+b} + \frac{1}{a+c}\right). \quad (2)$$

Where:
a: number of occurrences of the pair (l_i, l_j)
b: number of occurrences of the pairs where l_i is not followed by l_j
c: number of occurrences of the pairs where l_j is not followed by l_i

[1] This value can be adjusted easily.

The value of this coefficient varies from 0 to 1. When it is equal to 0.5 l_i is usually observed with l_j. Thus, an expression is considered as a collocation if the *KUC* coefficient is greater than 0.5.

- **Frequency of repetition:** This measure is used to know whether the lemma of the textual form to check repeats itself in the text. In fact, if a word is rare, one can suppose that it hides an error. This idea is based on the assumption that "Words (or more precisely lemmas of words) of a given text tend to repeat themselves" [16]. Indeed, according to research carried out by [16] on an Arabic textual corpus, it seems that a textual form can appear 5.6 times on average, whereas a lemma can appear 6.3 times on average in the same text.
  For each lemma we calculate its frequency of occurrence within all text, using the following formula:

$$p(l_i) = \frac{\text{number of occurences of } l_i}{\text{total numbers of lemmas}}. \quad (3)$$

The word w_i whose lemma is l_i is closer to its distant context as the value of $p(l_i)$ is higher.

Finally, these three measures are combined by the following linear formula:

$$F(w_i) = \alpha * p(w_i|C) + \beta * KUC(w_i) + \delta * p(l_i). \quad (4)$$

Where *F(wi)* is the total frequency of appearance of the word w_i in the text, and α, β and λ are three coefficients related to the three calculated contextual probabilities cited above. The values associated with these coefficients cannot be predicted, but must be obtained through several tests and comparisons of relevance. However, we estimate[2] that the value of α and β should be more important than that of λ because the context close to the target word is more relevant than its remote context. For each word, we calculate the $F(w_i)$ value which, compared with a threshold value, will validate the relevance of this word in its context.

3.2 Context-Vector Method

In this method we represent each word in a sentence by a vector representing its context. Therefore, a vector Vw_i is a vector representation of the probability of co-occurrence of a word w_i with all the words in the same sentence. If we consider the following sentence:

شرب الرجل كلبا(كأسا)

The men drunk a dog (glass)

The matrix below shows the co-occurrence probability of each word w_i in a sentence with its neighbours in the same context. The columns represent the words w_i and the rows represent the elements of the vector Vw_i. Thus, a cell contains the co-occurrence probability of the word w_i with the word w_j.

[2] After many tests we chose $\alpha = 2$ $\beta = 1$ and $\lambda = 0.5$.

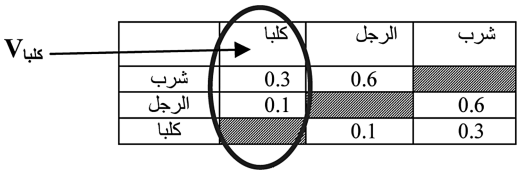

Fig. 1. Matrix of words' co-occurrences in a sentence

To represent the degree of correlation of each word w_i with the other words in the sentence, we propose to calculate the norm of each vector. Consequently, we evaluate the norm of each word's vector and we compare them to a threshold. The words having a norm lower to the threshold will be added to the list of probable errors.

In the last example, the norms of the words' vectors شرب, الرجل ,كلبا are respectively equals to 0.67 ; 0.6 et 0.31 the word having the lower norm is كلبا (dog), it will be probably then suspected.

3.3 Vocabulary-Vector Method

The vocabulary relating to a text is a representative element for this later and a good indicator of its coherence. Consequently, we can study the semantic validity of a sentence by using the vector representation cited previously. Thus we propose to represent each word in the sentence using a vector according to its probability of occurrence with each word in the vocabulary. To evaluate the proximity between two vectors, we use the measure of angular distance expressed as following:

$$\mathrm{Dist}(\mathrm{Vw}_i, \mathrm{Vw}_j) = \arccos(\mathrm{Sim}(\mathrm{Vw}_i, \mathrm{Vw}_j)) \,. \tag{5}$$

$$\mathrm{Sim}(\mathrm{Vw}_i, \mathrm{Vw}_j) = \cos(\mathrm{Vw}_i, \mathrm{Vw}_j) = \frac{\mathrm{Vw}_i \mathrm{Vw}_j}{\|\mathrm{Vw}_i\| * \|\mathrm{Vw}_j\|} = \frac{\sum_{t=1}^{k} \mathrm{Vw}_{it}\, \mathrm{Vw}_{jt}}{\sqrt{\sum_{t=1}^{k} \mathrm{Vw}_{it}^2 \sum_{t=1}^{k} \mathrm{Vw}_{jt}^2}} \,.$$

We calculate the angular distance for each word's vector *Vwi* regarding to all the words' vectors Vw_j of the sentence. The most distant vector to the context is the one which appears rarely with the words in the vocabulary. To select this vector, the sum of angular distances of each word's vector is calculated and then compared to a threshold. Those having a sum higher than the threshold will be suspected.

3.4 Latent Semantic Analysis Method

"*LSA* (Latent Semantic Analysis) is a method that makes possible the acquisition of knowledge by an automatically analysis of big textual corpora" [14]. Particularly, this method identifies the semantic similarity of two words, two textual segments or their combination even though these words or textual segments don't appear together.

The principle of *LSA* method consists on representing the words (terms) called lexical unities and the textual segments (documents: sentences, paragraphs or texts) called textual unities by vectors in a vector space of reduced dimensions in regards to the original space. The original space is represented by a matrix of co-occurrence (or matrix of words by context) *X(t, d)* which represents the corpora of training, where the *t*

rows correspond to the lexical unities, and the *d* columns to the textual unities. A cell in this matrix contains the number of occurrences of a lexical unity in a textual unity.

The next step of *LSA* method consists, on expressing this matrix in a product of three other matrixes *T(t,r)*, *S(r,r)* et *D(r,d)* thanks to a sort of factorial analysis called *Singular Value Decomposition (SVD)*. The matrix *T* is orthogonal and represents the original term vectors*, S* is a diagonal matrix called also singular value matrix and *D* is an orthogonal matrix of original document vectors.

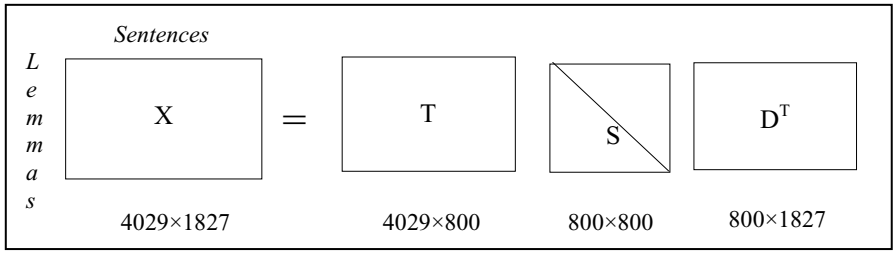

Fig. 2. Singular value decomposition of the matrix of co-occurrence

In our case, the matrix *X* is built during the training phase. The rows correspond to the lemmas of this corpus (we count **4029** lemmas), the columns represent the sentences (the corpus is composed of **1827** sentences).

The reduction of dimensions consists on the choice among the *n* dimensions the *k* ones that are the most pertinent and the most representative of the original space. This is done from the diagonal matrix *S* sorted according to the rank of its singular values. In this way, we obtain three matrixes *T (t,k)*, *S(k,k)* et *D(k,d)* of reduced dimensions (for us k=300, determined after some tests).

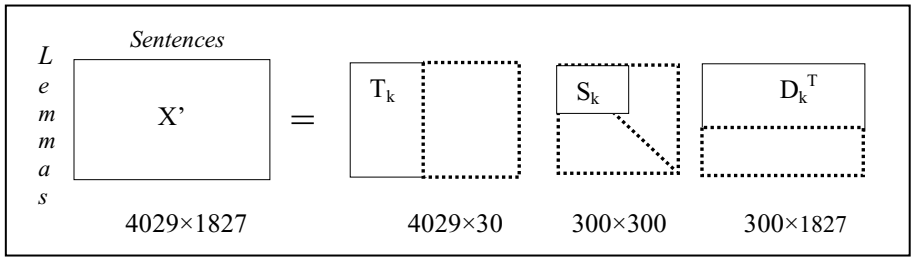

Fig. 3. Dimension reduction of the matrix of co-occurrence

However, before its decomposition in singular values, the initial matrix of co-occurrence undergoes a whole of transformations which consists in weighting each cell in order to highlight the importance of a word in a particular passage and its importance in the field of the speech in general. Therefore, we chose to apply the entropy to our initial matrix. This measurement is especially used in the field of the knowledge extraction and it qualifies the state of disorder of a source of information. It is thus calculated for each word of the matrix by the following formula:

$$\text{Entropy}(w_i) = 1 \quad \frac{\sum_{j=1}^{n} f_{ij} \text{Log} f_{ij}}{\text{Log} n}. \tag{6}$$

Where n is the number of sentences in which the term w_i appears at least only once, f_{ij} is the frequency of appearance of the word w_i in the sentence.

The variant of the *LSA* method that we propose here verifies the semantic validity of words in a given sentence by comparing their semantic vectors which are extracted from the reduced matrix of co-occurrence which is obtained during the training stage. To measure the semantic proximity of two vectors in this matrix, we use, in the way of *Vocabulary-Vector* method, the measure of the angular distance. Thus, each semantic vector Vw_i of the word w_i is compared to all the others vectors Vw_j in the sentence using the angular distance. The sum of these distances is then calculated for each word and is compared to a threshold. If this value is higher than the threshold, the correspondent word is suspected.

3.5 Voting Method

Since our system is based on the assumption that there is one error at the most in a sentence and the suspected errors are sorted by a decreasing order of probability in each method we chose to apply a voting procedure of type *uninominal with classification* (the candidates are sorted and only one among them will be the winner). We present here the principle applied by this procedure.

1. We calculate the number of occurrences of the different hypothetical errors ranked first in each list, given by each method.
2. We select the errors having the biggest number of occurrences. If only one error obtains the biggest number of occurrences, this one is selected as being the most probable error in the sentence. Otherwise, we calculate once again the number of occurrences for errors but using the next rank.
3. We repeat this process until one error obtains the biggest number of occurrences.

However, this voting method can induce sometimes to a blocking situation when the number of occurrences of selected errors in the first rank never changes. In this case, we use a *confidence degree* that is attributed to each method, in order to select among the list of the retained errors, that one detected by the method having the highest confidence degree.

4 Correcting Semantic Errors

To correct semantic errors we proceed by generating all the forms close to the error. These forms are obtained through one editing error. They are then all added to a list, which contains the candidates for the correction. Because of lexical proximity of Arabic words, the number of these candidates can be excessively high and one could estimate that an average of 27 forms will be suggested for the correction of each error. In extreme cases, this number can reach 185 forms [2].

To reduce the number of candidates, we propose to substitute the erroneous word with each suggested correction and form thus a set of candidate sentences. These sentences are processed once more by the detection part of the system and sentences containing semantic anomalies are eliminated from the list. The remaining sentences are then sorted using the combined criteria of classification presented below:

- **Typographical distance criterion:** It measures the degree of resemblance between an erroneous word and a candidate correction of the word. [17] confers weights to the various operations of edition according to their relevance: 1.5 for adding a character, 1 for substituting two characters and 0.5 for deleting a character.
- **Proximity value criterion:** According to [2], there is a correlation between the classification of candidate corrections and the proximity values between character strings. Candidate corrections can be classified according to the value of their proximity to the form they aim to correct. This value is defined as: "*the sum of the squares of the sizes of the common maximum sub-strings*".
- **Position of error criterion:** It gives more significance to the principal word of the sentence. This word is rarely incorrect since the writer is supposed to be more attentive when writing the beginning of the sentence rather than its end [18].

5 Context of Work

This work comes to complete the previous one [12] that was interested by the problem of hidden errors in Arabic texts. The system that we proposed for the treatment of these errors is based on a Multi-Agent-System (MAS). This system is composed principally of an agent for the correction and two groups of agents for detection: a group of syntactic agents responsible of the analysis of the syntactic anomalies and a group of semantic agents for the semantic inconsistencies. Only the agent correction and the group of syntactic agents were well studied and implemented, we thus supplement by this research the semantic part.

Accordingly, we implemented our semantic checker as a group of semantic agents, where each method suggested will be applied by a specific agent. Moreover, one *Supervisor* agent of the group is in charge of the activation of the different semantic analyzer agents. The semantic agents work, therefore, in parallel and communicate their results to the Supervisor which selects the most probable error among the lists of errors (given by each analyzer agent) thanks to the voting procedure.

The following figure illustrates the global architecture of the system using the two groups of syntactic and semantic agents in the two phases: detection and the correction. Because of the need of various linguistic information about the input text, a morpho-syntactic analysis [2] is performed at the beginning.

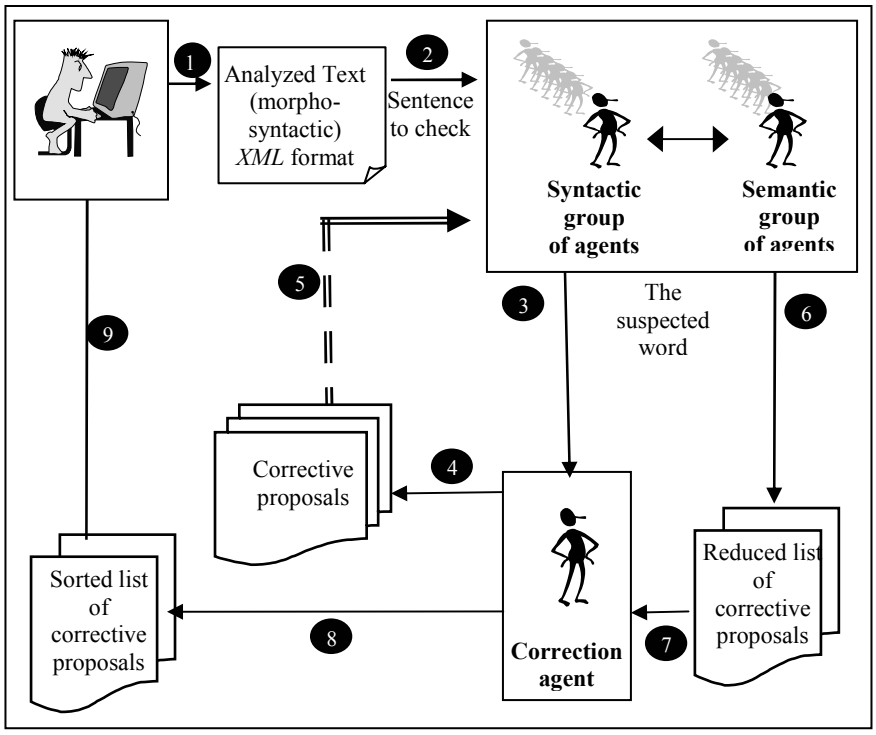

Fig. 4. A multi-agent system for the detection-correction of hidden errors

6 Testing and Results

We have built our own corpus of training in order to extract the data used by the various agents. This corpus is made of 30 economic texts (29 332 words) available on the Net, and which come from the corpus of contemporary Arabic, collected, treated and classified by category by [19]. We also chose a corpus of test of the same field counting 1 564 words and 50 hidden errors in 100 sentences.

6.1 Evaluation of the Detection Component

The following figure illustrates the performance of each agent and that one of the global system in term of accuracy.

The highest rate of accuracy for the semantic group agents is that of the Co-occurrence-Collocation agent with a value of 89.18%. This performance is explained by the complementarities of the phenomena of co-occurrence, collocation and repetition. On the other hand, the rate provided by *LSA* agent (82.92%) is weaker; this is certainly due to the modesty of our data of training which cause a high rate of sur-detection of errors. However, we think that *LSA* method remains always promising regarding the methods only based on the co-occurrences of the words. Indeed, the rate

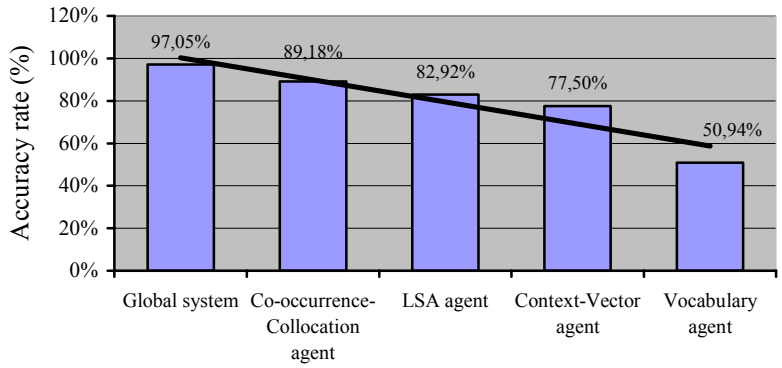

Fig. 5. Evaluation of the semantic group of agents

of precision of the agent *Context-Vector*, is low (77.5%) and that of the *Vocabulary-Vector* agent is not excellent (50.94%). The improvement of the results of the latter would require better corpora of training and strategy of extracting the vocabulary. Regarding the result of the evaluation of the total system, we can say that the rate of precision which is equal to 97.05% is very satisfactory. The performance of the voting system and its contribution for the selection of the most probable error in the sentence are thus confirmed.

6.2 Evaluation of the Correction Component

This phase was tested on two levels; initially after obtaining all the proposals for a correction, then after the reduction of the number of proposals. The obtained results are illustrated in the table hereafter.

Table 1. Evaluation of the correction agent

	Coverage	Accuracy	Ambiguity	Proposal	Rank
Initially	100%	100%	100%	46.67	13.82
After reduction	100%	80%	80%	5.98	3.43

We notice that our method of minimization of proposals decreases considerably, the average number of the proposals of 98% (from 46.67 to 5.98 proposals on average). Although, this reduction has reduced the ambiguity of our corrector of 20%, it did not occur without damage. In fact, it caused the fall of the precision (reduction of 20%).

7 Conclusion

Our system of detection of semantic hidden errors gave satisfactory results (97.05% of accuracy) in spite of the constraints and the restrictions related to the size and the non-diversity of our training corpora. We point out, also, the contribution of the process of

correction which made possible the reduction of the number of proposals by 98% and advanced the correct form in the first ranks. However, because our solution uses a training stage and its performance depends on the quality of this training, we estimate that the results obtained can be furthermore improved specially by more tests and bigger training corpora. Other prospects are also in sight, we think indeed of integrating the two groups of syntactic and semantic agents unit in order to test and evaluate the global system devoted to the treatment of hidden errors in Arabic texts.

References

1. Verberne S.: Context sensitive spell checking based on word trigram probabilities. Master thesis Taal, Spraak & Informatica, University of Nijmegen, (2002)
2. Ben Othman C.: De la synthèse lexicographique à la détection et la correction des graphie fautives arabes. Thèse de doctorat, Université de Paris XI, Orsay, (1998)
3. Golding A.: A Bayesian hybrid method for context-sensitive spelling correction. In Proceedings of the third Workshop On Very Large Corpora, Cambridge, Massachuses, USA, (1995), 39-53
4. Golding A. and Schabes Y.: Combining trigram based and feature based methods for context sensitive spelling correction. In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, (1996), 71-78
5. Golding A. R. et Dan Roth. A winnow-based approach to context-sensitive spelling correction. Machine Learning, (1999), 34(1-3), 107-130
6. Xiaolong W., Jianhua L. Combine trigram and automatic weight distribution in Chinese spelling error correction, Journal of computer Science and Technology, Volume 17 Issue 6, Province, China, (2001)
7. Bigert J., Knutsson O. 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'02), Frascati, Italie, (2002)
8. I. Bolshakov, A. Gelbukh.: On Detection of Malapropisms by Multistage Collocation Testing. NLDB-2003. Lecture Notes in Informatics, Bonner Kllen Verlag, (2003), 28–41
9. I. A. Bolshakov, A. Gelbukh. Paronyms for Accelerated Correction of Semantic Errors. International Journal on Information Theories and Applications, Vol.10, (2003), 11–19
10. A. Gelbukh, I. Bolshakov. On Correction of Semantic Errors in Natural Language Texts with a Dictionary of Literal Paronyms. AWIC-2004. Lecture Notes in Artificial Intelligence, N 3034, Springer (2004), 105–114
11. I.A. Bolshakov, S.N. Galicia-Haro, A. Gelbukh. Detection and Correction of Malapropisms in Spanish by means of Internet Search. TSD-2005. Lecture Notes in Artificial Intelligence, N 3658, Springer (2005), 115–122
12. Ben Othmane Z. C., Ben Fraj F., Ben Ahmed M.: A Multi-Agent System for Detecting and Correcting "Hidden" Spelling Errors in Arabic Texts. NLUCS 2005: 149-154
13. Ben Hamadou A. Vérification et correction automatique par analyse affixale des textes écrits en langue naturelle : le cas de l'arabe non voyellé. Thèse d'état en informatique, Faculté des Sciences de Tunis, (1993)
14. Landauer T.K., Foltz P.W. et Laham D., An introduction to Latent Semantic Analysis.Discourse Processes, Vol. 25, (1998), 259-284
15. Mlayeh I. Extraction de collocations à partir de corpus textuels en langue arabe. Mémoire de mastère, Ecole nationale des sciences informatiques, Université de la Manouba, 2004.

16. Ben Othmane Z. C. and Ben Ahmed M., le contexte au service des graphies fautives arabes. TALN 2003, Nantes, (2003), 11-14
17. Aloulou, C. Utilisation de l'approche multi-critère pour orienter un processus de correction des erreurs d'accord dans des phrases de la langue arabe non voyellée. Mémoire de DEA, Institut Supérieur de Gestion, Université de Tunis III, (1996)
18. Courtin J., Genthial D. et Menézo J. Intégration de strategies de correction dans un système de detection/correction d'erreurs, Colloque Informatique et Langue Naturelle (ILN93), Nantes, (1993)
19. Sulaiti L. Designing and Developing a Corpus of Contemporary Arabic. Master of Science, School of Computing, University of Leeds, United Kingdom, (2004)