Detection of semantic errors in Arabic texts

Article *in* Artificial Intelligence · February 2013

DOI: 10.1016/j.artint.2012.07.002

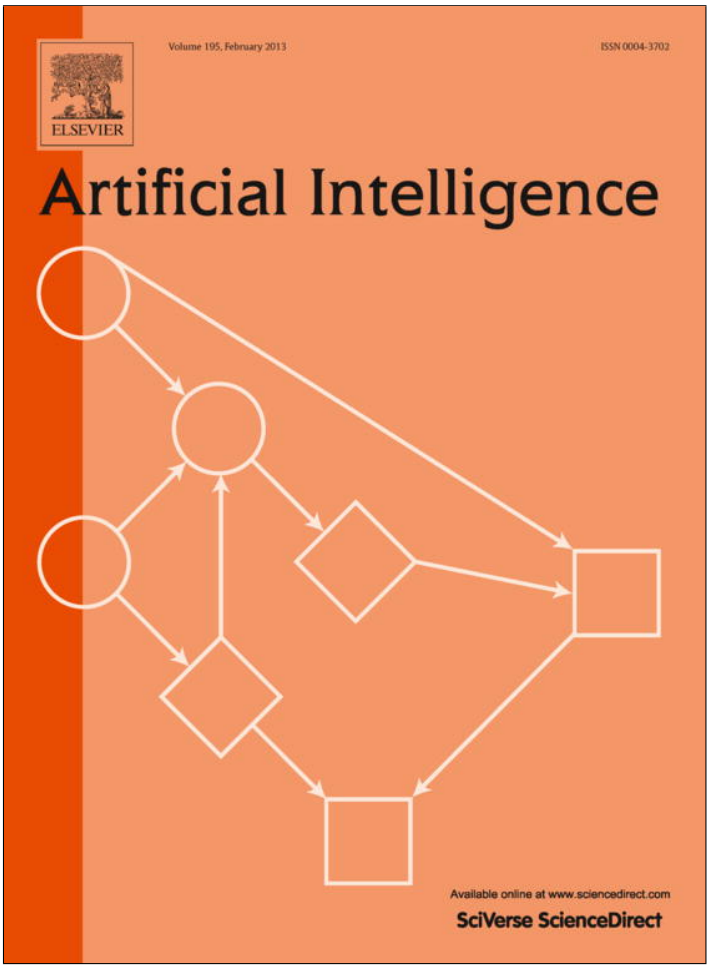
CITATIONS
10

READS
1,047

2 authors:

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



Artificial Intelligence 195 (2013) 249–264

Contents lists available at SciVerse ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Detection of semantic errors in Arabic texts

Chiraz Ben Othmane Zribi *, Mohamed Ben Ahmed

RIADI Laboratory, Manouba University, Tunisia

ARTICLE INFO

Article history:
Received 11 August 2011
Received in revised form 9 July 2012
Accepted 10 July 2012
Available online 16 July 2012

Keywords:
Semantic error
Detection
Statistical method
Linguistic method
Combining methods
Co-occurrence
Collocation
Latent Semantic Analysis (LSA)
Multi-Agent System (MAS)
Arabic

ABSTRACT

Detecting semantic errors in a text is still a challenging area of investigation. A lot of research has been done on lexical and syntactic errors while fewer studies have tackled semantic errors, as they are more difficult to treat. Compared to other languages, Arabic appears to be a special challenge for this problem. Because words are graphically very similar to each other, the risk of getting semantic errors in Arabic texts is bigger. Moreover, there are special cases and unique complexities for this language. This paper deals with the detection of semantic errors in Arabic texts but the approach we have adopted can also be applied for texts in other languages. It combines four contextual methods (using statistics and linguistic information) in order to decide about the semantic validity of a word in a sentence. We chose to implement our approach on a distributed architecture, namely, a Multi Agent System (MAS). The implemented system achieved a precision rate of about 90% and a recall rate of about 83%.

1. Introduction

Semantic errors result in morphologically and syntactically valid words whose use in context is senseless or absurd. Typically, writers make such errors through ignorance or keyboard slips. An ignorant writer may confuse the intended word with another one with similar orthography or pronunciation. When the writer's knowledge of the meanings of words is imprecise, he or she may choose a word whose meaning seems appropriate but which is in fact incorrect. The following sentence shows two examples of semantic errors that can be caused by the writer's ignorance of the word "piece":

Can I have a **peace**/**member** (piece) of cake?

In the first case, the writer mistypes "piece" as "peace" because it is similar in sound, while in the second case he uses the word "member" because it is a synonym of the intended word. Both errors are morphologically and syntactically correct but semantically incorrect in the context "cake".

When the semantic error results from writing slips the erroneous word is frequently similar in letters (e.g. insertion of a letter, substitution of a letter by another, etc.) to the correct word.

Example:

Her mother prepared a delicious **desert** (dessert)

* Corresponding author.
E-mail addresses: ChirazBenOthmane@riadi.rnu.tn (C. Ben Othmane Zribi), MohamedBenAhmed@riadi.rnu.tn (M. Ben Ahmed).

0004-3702/$ – see front matter
http://dx.doi.org/10.1016/j.artint.2012.07.002

In this example, the deletion of a letter produces the word “desert” which disturbs the text cohesion.

All modern text editors have tools for error detection. They focus on orthographic errors, and the lists of correction suggestions are similar to the suspicious word in letters and/or sounds. Grammatical errors are not always detectable because of deficiencies of syntactic analyzers, and then suggested syntax corrections are rare or imperfect. Semantic errors are not detected at all. In fact, handling this type of errors is more difficult. It requires information at higher levels than morphology and syntax. Detecting and correcting semantic errors is still the subject of ongoing research in the field of natural language processing.

This work is an extension of our previous research [4,5] which dealt with the treatment of hidden errors in Arabic. These are errors resulting in words lexically correct but syntactically and/or semantically incorrect. In the current research, we especially focus on the detection of semantic errors in Arabic texts. The approach we propose combines four methods using the context of the word to be checked. We chose to implement it on a distributed architecture, namely, a Multi-Agent System (MAS) because it offers various benefits such as cooperation, interaction and parallelism, giving enhanced results.

To date, most Natural Language Processing (NLP) work in this sub-area has been devoted to English and other European languages and no other work (that we are aware of) has treated the problem of semantic errors in Arabic texts, in spite of its importance. Indeed, Arabic words are lexically very similar and this lexical proximity increases the risk of semantic errors. In Arabic texts, one is likely to come across a real word by making a mistake or just by changing the spelling of the intended word. In addition, Arabic is morphologically rich and has specificities such as agglutination and vowelization, which makes it more difficult to handle than the Indo-European languages.

The reminder of the paper is organized as follows: in the first part, we present related work. In the second part, we explain our motivation and continue with an exposition of the difficulties regarding Arabic and the types of semantic errors to detect. In the third part, we describe the proposed approach for detecting semantic errors and we present after the MAS architecture and the implementation details. The fourth and last part is devoted to the description of the performance evaluation of the resulting system and it offers comparison to related work.

2. Related work

In literature, the problem of semantic errors has been seen through two different perspectives. The first group of researchers has considered this problem as the resolution of lexical ambiguity. They used pre-established sets of words named “confusion sets”, containing ambiguous words similar in sound (i.e. {stationary, stationery}), in letter (i.e. {dessert, desert}) and in usage (i.e. {between, among}). According to this approach, a word is simply suspected when a member of its confusion set better fits its context. It is then corrected by selecting the most likely confusable alternative with regard to the context. The second group of researchers was not restricted to predefined confusion sets. They used the context to detect semantic errors by applying methods based on semantic or probability information.

Golding [14] is the originator of the confusion set based method. He identified 18 confusion sets for the English language. He proposed with his colleagues several machine learning methods (for the same data set), presented here in chronological order: the Bayesian hybrid method based on probabilities as well as collocations [14], the Tribayes method combining a part-of-speech trigram method with the Bayesian hybrid method [15] and the Winnow algorithm using nearby and adjacent words as feature with weighted-majority voting [16]. These methods gave respectively a precision rate, on average, of 83%, 89% and 93.5%. The best result was obtained later by Carlson et al. [9]. They proposed a method based on the SNOW (a multi-class classifier) learning architecture and scaled the correction system up to 265 confusion sets with 99% accuracy. Other researchers joined them. They proposed new methods and tested their system on the same confusion sets. Examples include Jones and Martin [23] who applied the latent semantic analysis and obtained a precision rate of 83%, it is also the case for Mangu and Brill [26] who chose a rule-based method which achieved a precision of 88%. More recently, the Winnow method was tested on the Icelandic language for 11 confusion sets and gave a precision of 80.9% [19]. According to the authors the decline in performance compared to the English language is justified mainly by the nature of the Icelandic language which is morphologically rich. In a similar vein, some works have developed correction systems based on the web-scale N-gram models. In these systems, the word choice depends on how frequently each candidate (a member of the confusion set) has been seen in the given context in web-scale data, such as the Google N-gram Corpus. We can cite for example: Bergsma et al. [6,7] who improved the accuracy (95.7% on average) for 5 confusion sets where the reported performance in [16] is below 90%. Xu et al. [37] built their system on the basis of the model of Bergsma et al. [7] and used dependency parse features combined with distributional information to improve the performance. They tested their experiments on 14 common and rare confused sets and the obtained results achieved an accuracy ranging from about 92% to 99%.

As regards the approaches not based on confusion sets, they achieved less effective results since the solution to this problem is more difficult. We can quote Verberne [32] who applied a trigram based method and tested it on 5,500 words of the British National Corpus (a subset of the training data) with 606 errors introduced by inserting all possible instances from a pre-compiled list of 134 error types. This achieved a detection recall rate of 72% and a precision rate of 98%. When tested on data set out of the training data the results for the detection were largely inferior with a recall rate of 51% and a precision

Table 1
Example of words graphically close. Real-words with an edit distance of 1 from كتب/ktb/(write/books).

Insertion	Deletion	Replacement	Transposition
فكتب/fktb/(then he wrote)	تب/tb/(it ceased)	عتب/Etb/(doorsills)	تكب/tkb/(she spills)
بكتب/bktb/(with books)	كب/kb/(he spilled)	يتب/ytb/(he repents)	كبت/kbt/(a repression)
يكتب/yktb/(he writes)		رتب/rtb/(he arranges)	
أكتب/Oktb/(I write)		كثب/kvb/(close to)	
تكتب/tktb/(she writes)		كعب/kEb/(a foot)	
نكتب/nktb/(we write)		كحب/kHb/(as cereals)	
مكتب/mktb/(a desk)		كسب/ksb/(he gains)	
ككتب/kktb/(as books)		كذب/k*b/(he lies)	
وكتب/wktb/(and books)		كلب/klb/(a dog)	
...		...	

rate of 5%. Hirst and Budanitsky [18] used semantic distance measures in WordNet to detect and correct malapropisms.[1] An error is flagged when a spelling variation (any word with an edit distance of 1 from the original word; the insertion, deletion, or replacement of a single character or the transposition of two adjacent characters) results in a new word that is semantically closer to the context. This method achieved a precision of about 23% when tested on approximately 300,000 words from the 1987–1989 Wall Street Journal corpus, with about 1,400 malapropisms randomly induced at a frequency of approximately one word in 200. Wilcox-O'Hearn et al. [36] applied a method using a large word-trigram probability model to detect and correct real-word[2] errors. Islam and Inkpen [21] proposed a method to detect and correct real-word errors that relies on using a very large set of trigrams of English words (about 977 million trigrams) with their frequencies, collected by Google in 2006 (Google Web 1T data set). A word in a sentence is suspected if there is a candidate word (the most similar to the error) in the set of trigrams which has higher frequency within the same context. Wilcox-O'Hearn et al. [36] and Islam and Inkpen [21] were not limited to semantic errors and more restrictively to malapropisms, they, however, evaluated their system on the same test data used by Hirst and Budanitsky [18]. For the detection task, the method proposed by Wilcox-O'Hearn et al. [36] achieved the best result with a precision of about 53%, while the one of Islam and Inkpen [21] was better in recall (89%). Whitelaw et al. [35] also made use of the web. They implemented a language independent system that performs spellchecking and auto-correction. This system used statistical models (error model, N-gram language model and list terms) inferred from the web. The authors claim that they can detect and correct real-word substitutions (word usage and grammar) as well as non-word errors. However, the experiments tested on human typed errors for English and German did not detail the nature of these errors.

Moreover, errors in texts produced by automatic speech recognizers were detected by identifying words which are semantic outliers with respect to other words in the transcript. Sarma and Palmer [30] used co-occurrence statistics to analyze the context of words in a dialogue query and to identify and correct errors. Inkpen and Désilets [20] used Point-wise Mutual Information (PMI), a statistical measure of the semantic independence of two terms, to determine errors in transcripts. Voll et al. [33] applied a statistical error detection technique based on co-occurrence relation to post-speech recognition radiology report detection. However, for these works, all incorrectly decoded words were similarly considered as "semantic outliers" and then the reported results did not specify if the detected errors were only semantic.

3. Why is detecting semantic errors more crucial for processing of Arabic texts?

Arabic words are graphically very similar to each other. This increases the risk of getting semantic errors in texts, since a typing/spelling error could result in a valid word.

Table 1 shows an example of the word كتب/ktb/(write/books) being changed into several different real words by the insertion, deletion, or replacement of a single character, or the transposition of two adjacent characters.

The study we conducted previously [3], proved this phenomenon. In this experiment, we applied four editing operations (adding one letter, substitution of a letter by another, deleting a letter, interchanging two adjacent letters), to all dictionary words. For each word, we calculated the number of correct forms obtained among the automatically built forms, called "the number of lexically neighboring words". These counts gave us a clear idea about the degree of similarity between words of a given language. We found that the words in Arabic are much closer to each other. As shown in Table 2 the average number of neighboring forms for Arabic is 26.5 and it can reach a maximum of 185, which is a significant value compared to that calculated for English and French.

[1] The term "malapropism" is used here to designate a type of semantic error that replaces one content word by another existing word similar in sound or letters.

[2] Real-word errors are spelling errors that result in real words. They are not limited to semantic errors; they also include syntactic ones.

Table 2
Neighboring words

Per word	Arabic		English		French	
	Average	*Maximum*	*Average*	*Maximum*	*Average*	*Maximum*
All generated forms	458	1187	505	1483	892	1881
Correct forms	26.5	185	3	54	3.5	45
Average ratio	5.79%		0.59%		0.39%	

These counts also inform us about the probability of obtaining a correct word when an error is made on a word. Thus, we see that this probability for an Arabic word (5.79%) is 10 times greater than for an English word (0.59%) and 14 times greater than for a French word (0.39%). Let us note, however, that the results mentioned above are made on dictionaries[3] and not for textual data.

4. What are the difficulties for Arabic?

Due to the specificities of Arabic, detecting semantic errors is not an easy task compared to other languages (especially Indo-European ones). In fact, in Arabic there are numerous constraints of writing and various ambiguities. We focus here on those having a direct impact on our problem, namely: agglutination, vowelization, and sentence segmentation.

4.1. Agglutination

Agglutination is the addition of prefixes (e.g. articles, prepositions, conjunctions) and suffixes (e.g. pronouns), commonly called proclitics and enclitics to simple forms in order to get agglutinative forms or hyper forms. In Arabic, a textual form (or more generally a word) can represent a sentence thanks to its composite structure consisting of several elements concatenated to each other. Let us consider, for example, the following word: أتقدرونا/OtqdrwnnA/[4] it expresses the following sentence: “Do you respect us?” and it is segmented as follows: proclitic أ/O/(do you) + simple form تقدرون/tqdrwn/(respect) + enclitic نا/nA/(us).

Segmenting an Arabic agglutinative form into proclitic, simple form and enclitic is often ambiguous. Thus, a textual form (without considering any context) can be segmented in different ways.

With regard to semantic errors, when using enclitics one can easily get confused and thus, the meaning of the sentence would be disturbed or changed.

Example:

كتب التلميذ **لقلمه** (بقلمه)
/ktb AltlmY* **lqlmh** (bqlmh)/
The student wrote **for his pencil** (with his pencil)

This sentence is lexically and syntactically correct, but semantically incorrect since the proclitic ب/b/(with) was replaced by ل/l/(for).

4.2. Vowelization

Arabic texts can be fully vowelized, partially vowelized or un-vowelized. The absence of vowels makes the lexical items more ambiguous than in other languages, thereby aggravating the homography problem. The average number of ambiguities of a token in many languages is 2.3, whereas in Arabic, it can reach 19.2 [11]. For instance, the word العلم/AlElm can be read both as العِلْم/AloEilom/(science) or as العَلَم/AloEalam/(flag). As it is illustrated by the following sentence, a vowel error can cause a semantic error, since the modification of a vowel may change its meaning.

Example:

طَلَبُ **العَلَمِ** (العِلْمِ) وَ المَعْرِفَةُ فَرِيضَةٌ
/Talabu **AloEalami** (AloEilomi) wAlomaEorifati fariYDapN/
Seeking **flag** (knowledge) is required

[3] The experiment was carried out for an English dictionary of about 90,000 entries, a French dictionary of about 300,000 entries and an Arabic dictionary containing about 600,000 non-vowelized entries.

[4] We use the Buckwalter transliteration scheme to show romanized Arabic, [8].

4.3. Sentence segmentation

Segmentation of Arabic texts into sentences does not follow well-defined rules since punctuation marks do not occur always at sentence boundaries. Indeed, one can find whole paragraphs without punctuation except the full stop. Function words can sometimes substitute punctuation to mark the boundary of a sentence. *Example:* The coordinating conjunction و/w/(and) followed by a verb marks the beginning of a sentence.

For the detection of semantic errors, the context is a helping element to let us decide about the correctness of a word within a sentence. Thus, it would be important to know the type of context to consider.

5. What types of semantic errors to detect?

The disturbances caused by semantic errors can be divided into two categories: semantic inconsistencies and semantic incompleteness.

5.1. Semantic inconsistencies

When an error causes a meaningless sentence we say, there is a total inconsistency.
Example:

ترك له والده **ثورة** (ثروة)
/trk lh wAldh **vwrp** (vrwp)/
His father left him a **revolution** (fortune)

However, if the error causes an apparent meaningless sentence, such as a metaphorical expression, it is said to be partial inconsistency.
Example:

هذا الرجل **أسدا** (أسودا)
/hvA Alrjl **OsdA** (OswdA)/
This man is a **lion** (black)

In this example, deleting a letter in the word أسودا/OswdA/(black) results in the word أسدا/OsdA/(lion). Literally speaking, the sentence is meaningless. Yet, metaphorically, it means comparing a man with a lion.

5.2. Semantic incompleteness

Missing a coordinating conjunction or any other particle in a textual form within a sentence can make the sense of this sentence incomplete.
Example:

ضربت الولد **بكى** (فبكى)
/Drbt Alwld **bkY** (fbkY)/
She hit the boy **he cried** (then he cried)

Partial semantic inconsistency and semantic incompleteness are subtle and very difficult to detect. They require a higher level of treatment, such as pragmatic. Therefore, in this work, we focused on errors causing total semantic inconsistencies, while keeping in mind that partial semantic inconsistency can cause noise (over-detection) and the semantic incompleteness can engender silence (under-detection) for the detection system.

Moreover, to restrict the scope of our investigations, we assumed the existence of one error at most in a sentence. Also, we considered unvowelized texts since, except for a few didactic, poetic and literary manuscripts, Arabic writings such as newspapers or magazines generally do not write vowels.

6. Distributed problem solving approach: combining contextual methods

According to cognitive psychology, the process of human understanding is based on prior knowledge stored in semantic memory. Especially, the understanding of the meaning of a word is inferred from its mental representation obtained from the learning process [28]. Thus, to solve a problem or understand the meaning of a word in a text, the human uses resources stored in his memory to find a semantic link between the new information and the previous one. In the case of a polysemous word or a new word, he compares the current context of the word and the previous contexts stored in his memory in order to obtain the adequate meaning and thus to acquire new knowledge from prior knowledge.

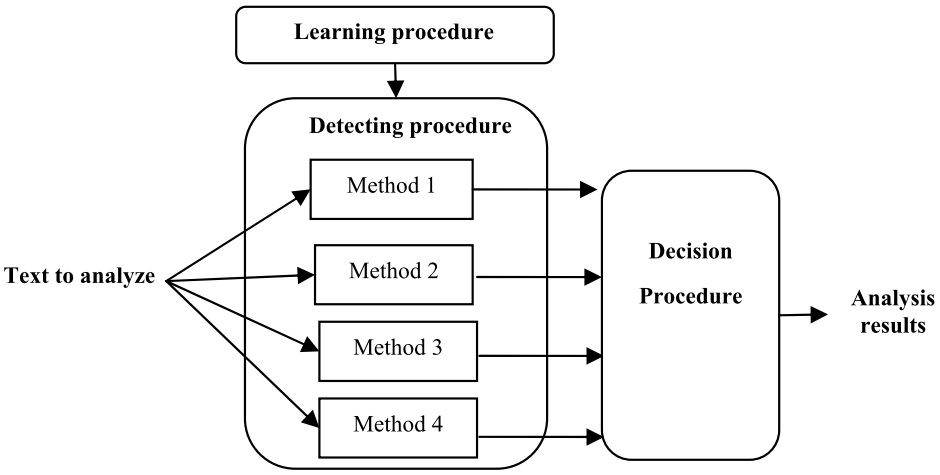


Fig. 1. Combining contextual methods to detect semantic errors.

Therefore, by analogy to humans, the computer should acquire prior knowledge about words and their different contexts to handle the meaning of words. This kind of knowledge can be obtained from several resources such as: semantic dictionaries, thesauri, semantic networks, anthologies or textual corpora.

With regard to our problem, these resources are required to obtain the meaning of a word and to check the coherence of its context. Because semantic resources for Arabic are not widely available, we chose a solution based on learning the meaning of words from textual corpora. This choice is based on the distributional linguistic theory that states: "The meaning of a word can be defined statistically, from all contexts (i.e. paragraphs, sentences, texts) in which the word appears" [25]. For example, the word "airplane" appears frequently with words such as "take off", "wing", "airport", and rarely with words such as "lion" or "forest".

To detect semantic errors, we propose an approach using the context (Fig. 1). We combine four contextual methods. Each method gives its own representation of words within sentences in terms of their contexts and compares this representation to the representations previously acquired during the learning stage. To select a single semantic error in a sentence, we use a decision procedure that confronts the different results given by the methods and identifies the most plausible error. We will show that the combination of methods is very helpful. In fact, the contribution of various methods can create a synergy that overcomes the limitations of each method and gives globally more satisfactory results.

7. Contextual methods

The methods we propose are based on the immediate context (within the sentence or the paragraph) and on the distant one (within the text) to verify the semantic validity of sentences within the text to analyze. A sentence is considered semantically valid, if it does not contain words causing a semantic disturbance. To detect semantic errors within a sentence, the methods calculate for each word to check a "semantic validity coefficient" based on the data collected during the learning stage. This ratio is compared to a threshold of acceptability (based on empirical experiments and previously set by each method) to determine the plausible errors.

Before beginning the presentation of these methods, we must point out that by "word" we mean, the lexical form rather than the surface form. In other words, for inflected forms and agglutinated ones we consider their canonical form (the lemma). Arabic is highly inflectional and agglutinative, making it possible to gather forms (plurals, duals, masculine, feminine, etc.) carrying the same meaning in the same lexical form. In addition, only content words are considered. Function words (like particles, proper names, and numbers) are ignored since they have low discriminating semantic power.

We distinguish in what follows, between long sentence, called "macro-sentence", and the short sentence, called "micro-sentence." The long sentence is delimited by punctuation. On the other hand, the short sentence is delimited by either punctuation or function word, and it is considered as a collection of words grouped to express a complete meaning.

We use the following notation:

$S = \{w_1, \ldots, w_i, \ldots, w_n\}$: The sentence to analyze (long or short, it depends on the applied method).

w_i: The word to analyze.

$\Omega(w_i)$: The semantic validity coefficient of the word to analyze.

$C = \{c_1, \ldots, c_j, \ldots, c_k\}$: The context of the word to analyze.

7.1. Method 1: Co-occurrence

We make here the following assumption: "*A correct word has a certain "affinity" with the words within its immediate context (neighboring words, sentence or paragraph)*". In other words, it is not uncommon to find it in co-occurrence with these same words in other contexts.

By referring to semantic knowledge already learned, one can then assume that a word can be considered erroneous if it is not previously seen (or rarely) in the presence of words in its near context. Let us consider the following example:

تعد البطالة أحد أهم الظواهر التي تلازم **المستمع** (المجتمع) الذي يتبنى نظام اقتصاد السوق

/tEd AlbTAlp Ohm AlZwAhr AltY tlAzm **AlmstmE**
(AlmjtmE) Al*y ytbnY nZAm AqtSAd Alswq/
Unemployment is considered as one of the most important aspects of **the listener** (the society) that adopts the market economy system

One can note here that the word مستمع/mstmE/(listener) is incorrect and must be substituted by the correct word مجتمع/mjtmE/(society). The near context here can help to identify this error since words such as: بطالة/bTAlt/(unemployment), اقتصاد/AqtSAd/(economy) and سوق /swq/ (market) rarely appear with the word مستمع/mstmE/(listener). The key question at this stage is which context to consider: Neighboring words, words within the sentence or words within the paragraph? For Arabic, the context that seems most suitable is the "macro-sentence". This is justified by the fact that such sentences are generally cohesive, expressing the main idea and containing words related to a given theme. We therefore believe that the use of a greater number of neighboring words ensures better validation of a word within its context.

Consequently, to calculate the semantic validity coefficient for each word w_i to check, we propose to first calculate its frequencies with all the words within its context (in our case the macro-sentence). Let $X_i = \{x_1, \ldots, x_k\}$ be the set of these frequencies. Then, we weight the average of X_i by the inverse of the standard deviation, as indicated below:

$$\Omega(w_i) = \frac{\overline{X}_i}{\sigma_i} \tag{1}$$

where:

$\overline{X}_i = \frac{1}{k}\sum_{j=1}^{k} x_j$: The average of co-occurrence frequencies of the word w_i with the words of its context.

$\sigma_i = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(x_j - \overline{X}_i)^2}$: The standard deviation of co-occurrence frequencies of the word w_i with the words of its context.

The standard deviation shows how much dispersion of co-occurrence frequencies there is from the average. Generally, high standard deviation indicates that the frequencies are spread out over a large range of values. Consequently, the division of the average frequency by the standard deviation would highlight words having frequencies of co-occurrence relatively well distributed over the entire sentence. For example, for a sentence containing six words, a word with the following co-occurrence frequencies $X_i = \{0, 0, 0, 0, 0, 20\}$ will have a lower semantic validity coefficient than another word with $X_{i'} = \{1, 5, 2, 3, 5, 4\}$ even if their co-occurrence frequencies averages are equal.

7.2. Method 2: Co-occurrence_Collocation

The assumption on which we rely in proposing this method states: "*Words within a sentence would have privileged relationship*". This method is quite similar to the preceding one, since it recommends that neighboring words can help to determine the semantic validity of a word. However, it differs from it in two ways. First, it takes into account the collocational relations between words. This would lead us to consider only "micro-sentences". Second, it considers that a sentence is a coherent set, consisting of a collection of linked words expressing a complete meaning.

The inclusion of collocations would be beneficial. In fact, words occurring in collocation can be regarded as mutually confirming. For example, the expression شوارع المدينة/$wArE Almdynp/(the streets of the city) is a collocation. The identification of this collocation should validate the two words شوارع/$wArE/(streets) and المدينة/Almdynp/(the city). Moreover, by using the collocations, we consider the syntactic relation between words and we are not limited to a "bag-of-words", which is generally insufficient to decide the semantic validity of a word.

Therefore, this method measures the semantic validity coefficient of a word w_i by calculating the weighted average of the probability of observing the word knowing the sentence in which it is located and its collocational coefficient, as follows:

$$\Omega(w_i) = \frac{\alpha p(w_i|S) + \beta\Delta(w_i)}{\alpha + \beta} \tag{2}$$

where $p(w_i|S)$ is the probability of co-occurrence of the word within the sentence; $\Delta(w_i)$ is its collocational coefficient (ranging between 1 and 0); α and β are weights assigned to these results to underline their respective contributions. It should be noted that these latter values are not constant. They can change during the experiments according to the efficiency of the used method (collocation or contextual words).

7.2.1. Co-occurrence probability

The probability of co-occurrence of a word w_i within the sentence S is expressed as follows:

$$p(w_i|S) = p(w_i|c_1, \ldots, c_k).$$

Calculating this probability is not easy because it requires a lot of learning data. We apply the reversing Bayes rule to obtain:

$$p(w_i|c_1,\ldots,c_k) = \frac{p(c_1,\ldots,c_k|w_i) \times p(w_i)}{p(c_1,\ldots,c_k)}.$$

By assuming that the presence of a word in a context does not depend on the presence of other words within this context, we perform the following approximation as previously proved [12]:

$$p(c_1,\ldots,c_k|w_i) = \prod_{j=1}^{k} p(c_j|w_i).$$

As we want to select the words having the highest co-occurrence probability, the probability $p(c_1,\ldots,c_k)$ can be ignored because it is the same for all words within the sentence and it has no effect on the result.

Thus, the probability of co-occurrence is calculated according to the following formula:

$$p(w_i|c_1,\ldots,c_k) = \prod_{j=1}^{k} p(c_j|w_i) \times p(w_i)$$

where:

$$p(c_j|w_i) = \frac{\text{Number of cooccurrences of } c_j \text{ with } w_i}{\text{Total number of occurrences of } w_i},$$

$$p(w_i) = \frac{\text{Number of occurrences of } w_i}{\text{Total number of words}}.$$

7.2.2. Collocation coefficient

Based on the assumption that collocations are syntactically well formed, we verify if the word to analyze matches with one or more morpho-syntactic collocation patterns. For this purpose, we have built 11 finite-state automata (3 represent verbal collocations, and the remaining are for nominal collocations) to identify these morph syntactic patterns represented as regular expressions.

If the word to check matches with more than one collocation pattern, the calculation of its collocational coefficient is done for each expression and, the word is given the highest value. The coefficient we used is the *Kulczinsky* measure [24] which is an association criterion identifying the degree of correlation of two lemmas L_i and L_j within an expression. It is calculated using the following formula:

$$\Omega(w_i) = KUC = \frac{a}{2}\left(\frac{1}{a+b} + \frac{1}{a+c}\right) \tag{3}$$

where:

a: the number of occurrences of the pair (L_i, L_j),

b: the number of occurrences of couples where L_i is not followed by L_j,

c: the number of occurrences of couples where L_j is not preceded by L_i.

The value of this coefficient varies between 0 and 1, when L_i (resp. L_j) is only observed with L_j (resp. L_i), it is greater than 0.5.

7.3. Method 3: Vocabulary_Vector

The method proposed here is based on the well-known model of vector representation used in the field of natural language processing and information retrieval. This model represents elements according to their descriptive features with vectors, in order to compare or rank them. The vector model has proved efficient for many applications. We cite as an example: semantic disambiguation and document retrieval. If we consider for example the case of conceptual vectors, where the features are the concepts of a thesaurus, these vectors represent ideas associated with any textual segment (words, sentences, texts) referring to concepts [22]. That is to say, a vector represents the meaning of a textual segment in terms of concepts and can be semantically compared with other similar vectors and thus a semantic comparison can be made between words in terms of synonyms, antonyms, etc.

The stage of "vectorization" (representation of a text by vectors) needs to be sensitive to the fundamental choice of elements and features to be made. These latter must be representative and discriminating but easy to extract and not very numerous. It depends, of course, on the type of task to achieve. For our problem, we consider that the vocabulary of a given domain is a good indicator of the text cohesion. Accordingly, and adopting the principle of vector representation cited above, we suggest studying the semantic validity of a sentence by representing each word with a vector based on vocabulary terms obtained from the training corpus. The method proposed by Mokrane [27] is used to choose the most representative vocabulary words. The set of the representative terms results from the union of two sets. The first one contains the most

frequent terms weighted by their distributions in the different texts of the corpus. The second set contains the most frequent co-occurrences in the corpus.

We make then the following assumption: "*If two words co-occur with the same vocabulary terms and their representative vectors according to this vocabulary are relatively close, they tend to appear frequently together in the same context*". Therefore, we can study the semantic correlation between words by comparing their relationship with the domain vocabulary.

Each word in the sentence (macro-sentence) is represented with a vector according to its co-occurrence frequencies with each vocabulary word. To assess the proximity between two word vectors $V w_i = \{v w_{i1}, \ldots, v w_{im}\}$ and $V w_j = \{v w_{j1}, \ldots, v w_{jm}\}$ ($m =$ number of words in the vocabulary), we use the metric of angular distance expressed as follows:

$$D(V w_i, V w_j) = \arccos \frac{V w_i \bullet V w_j}{\|V w_i\| \times \|V w_j\|} = \frac{\sum_{t=1}^{m} v w_{it} \times v w_{jt}}{\sqrt{\sum_{t=1}^{m} v w_{it}^2 \times \sum_{j=1}^{m} v w_{jt}^2}}. \tag{4}$$

Usually, this distance is interpreted as follows: "two words x and y are semantically close if $D\,(Vx, Vy) \leqslant 45°$. For $D(Vx, Vy) > 45°$, the semantic proximity is low and for about 90°, x and y have no relationship" [31]. The calculation of the semantic validity of a word w_i, is made by summing the angular distances between the word vector $V w_i$ and all the other word vectors $V w_j$ within the same sentence (macro-sentence).

$$\Omega(w_i) = \sum_{j=1}^{k} D(V w_i, V w_j). \tag{5}$$

7.4. Method 4: Latent Semantic Analysis (LSA)

This method is based on the Latent Semantic Analysis (*LSA*) model, which is a process for the knowledge acquisition from the fully automatic analysis of large textual corpora. More specifically, it identifies the semantic similarity between two words, two textual segments or their combination even if the words or the textual segments are not co-occurrents. *LSA* is based on the following definition: "Two words are similar if they appear in similar contexts; two contexts are similar if they contain similar words" [25]. This cross-recursion requires a much more complex mechanism than a simple count of occurrences.

In fact, the notion of co-occurrence can identify superficial relationships between words because it assumes that two words are semantically correlated if they appear in the same context. This statistical information about the context of a word is not sufficient, since it says nothing about the semantic links with all other words that never appear in conjunction with this word. For example, the statistical context of the word "lampshade" ("lighted", "lighting", "light", etc.) gives insufficient information on its meaning. If "lantern" never appears with "lampshade", we have no information about the semantic link between these two words, although "lantern" should be semantically considered close to "lampshade" because it co-occurs with words such as "lighted", "lighting" and "light".

7.4.1. Application of the LSA approach

In the LSA method a textual corpus is represented by a matrix. The rows represent the lexical units (i.e. terms) and the columns represent the textual units (i.e. sentences, paragraphs, documents). The value of each cell expresses the frequency of occurrence of a lexical unit in the corresponding textual unit. Thus, each lexical unit is represented by a semantic vector indicating its occurrence frequencies in the textual units. Two lexical units are considered semantically close if they are represented by close vectors obtained from the constructed matrix.

In our case, this matrix is obtained from the training corpus, where words (more specifically the lemmas) correspond to lexical units and sentences (macro-sentences) correspond to textual units. The construction of the matrix involves three steps:

The first step is to build an original matrix X, with m rows representing lemmas with n columns representing sentences. In order to highlight the importance of a word within its context, the matrix is transformed by weighting each cell. Thus, the word frequency in each cell is converted to its *log*. Then, the entropy of each word is computed over all entries in its row, and each cell entry is then divided by the row entropy value. The effect of this transformation is to weight each word occurrence directly by an estimation of its importance in the passage and inversely with the degree of to which knowing that a word occurrence provides information about which passage it appeared in.

In the second step, the matrix X is submitted to a factor analysis procedure that decomposes it into a product of three matrices: $T(m, t)$, $S(t, t)$ and $D(t, n)$ where T is an orthogonal matrix of $m \times t$ dimensions, S is a diagonal matrix of $t \times t$ dimensions, also known as the singular value matrix and D is an orthogonal matrix of $t \times n$ dimensions, as shown in the following figure:

The smallest values of the diagonal matrix S are removed in order to reduce the vector space dimensions, from t to k (as $k \ll t$) and to keep only the most important semantic information. This transformation of the initial matrix X is the third step of the *LSA* method. It gives the "compressed" or "smoothed" matrix X' which is the product of the three matrices $T(m, k)$, $S(k, k)$ and $D(k, n)$.

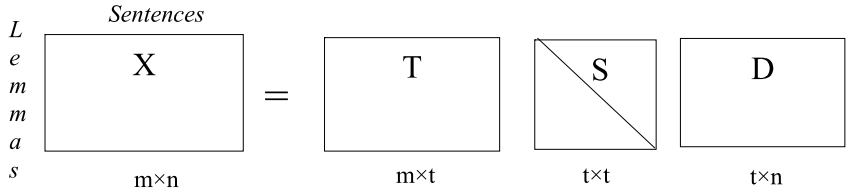


Fig. 2. Singular value decomposition of the matrix of co-occurrences.

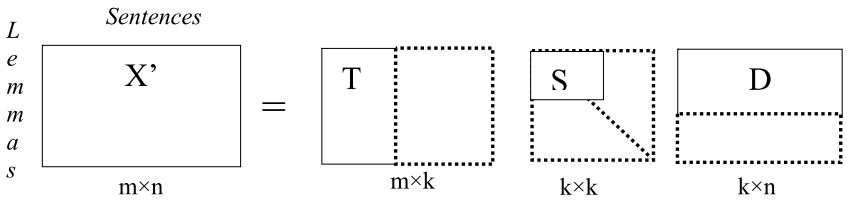


Fig. 3. Matrix size reduction.

Once the transformed matrix X' is constructed, it is used to calculate the validity semantic coefficient $\Omega(w_i)$ of the word w_i to check, and this (as with the previous method), by performing the sum of angular distances calculated for its word vector $V w_i$ compared to all the other word vectors $V w_j$ within the same sentence.

8. Implementation: a Multi-Agent System (MAS)

Since 1980, several Natural Language Processing (NLP) systems have been based on the multi-agent approach. We can cite as examples for the European languages, chronologically ordered: the system HERSAY II [10] for speech understanding, the system HELENE [38] for understanding hospitalization reports, the system CARAMEL [29] dedicated to automatic understanding of stories, the system CELINE [13] for the detection and correction of lexical and syntactic errors, and finally, the TALISMAN II system [34] dedicated to the morpho-syntactic analysis. For Arabic, we can mention: the system MASPAR [1] for parsing and the system MOUHALIL [17] for morphological tagging.

8.1. Motivation

NLP systems using a sequential architecture tend to show many of weaknesses such as the lack of interaction between the different phases of treatment, the difficulty of distributing knowledge into modular components, the rigidity of the architecture, the difficulty of updating the system and finally the processing time. Distributed systems and especially MASs represent a potential alternative to overcome these drawbacks. They reduce the complexity through the distribution of knowledge and tasks. Also, system development is made more flexible with this approach. In fact, one can add new agents without changing the system. Finally, they provide considerable gain in terms of time processing by the parallel or pseudo-parallel execution of processes attached to agents.

These advantages are particularly obvious when the system tackles a complex task that requires the intervention of several skills involving various knowledge and methods. This is especially true for the detection of semantic errors which is a relatively difficult task and needs more than one paradigm to be solved. It also requires the interaction with other levels of treatment such as syntactic or pragmatic ones for more efficiency.

8.2. General architecture

There is no consensus about the best design for a distributed system for NLP. It depends on specific needs and objectives. All the proposed architectures have both advantages and disadvantages. Hence, it is not possible to say that one architecture is better than another.

The architecture we propose for our semantic agents group is both hierarchical and pyramidal. It includes a supervisor agent, expert agents, and worker agents. The supervisor receives as input a preprocessed[5] text and it is responsible for activating and coordinating the actions of the expert agents. Each expert agent applies a method to detect semantic errors. These are the *Co-occurrence* agent, the *Co-occurrence_Collocation* agent, the *Vocabulary_Vector* agent and the *LSA* agent. The *Co-occurrence_Collocation* agent supervises two worker agents (that calculate the collocation coefficient ant the co-occurrence probability). It is responsible for their remuneration (by incrementing their weights) on success and for the combination of their results. The figure below illustrates this architecture.

[5] Preprocessing steps are as following: morphological analyzer, morpho-syntactic disambiguation (including lemmatization) and segmentation into micro-sentences. We used for this our fully automated tool and to avoid errors, the final results were manually controlled.

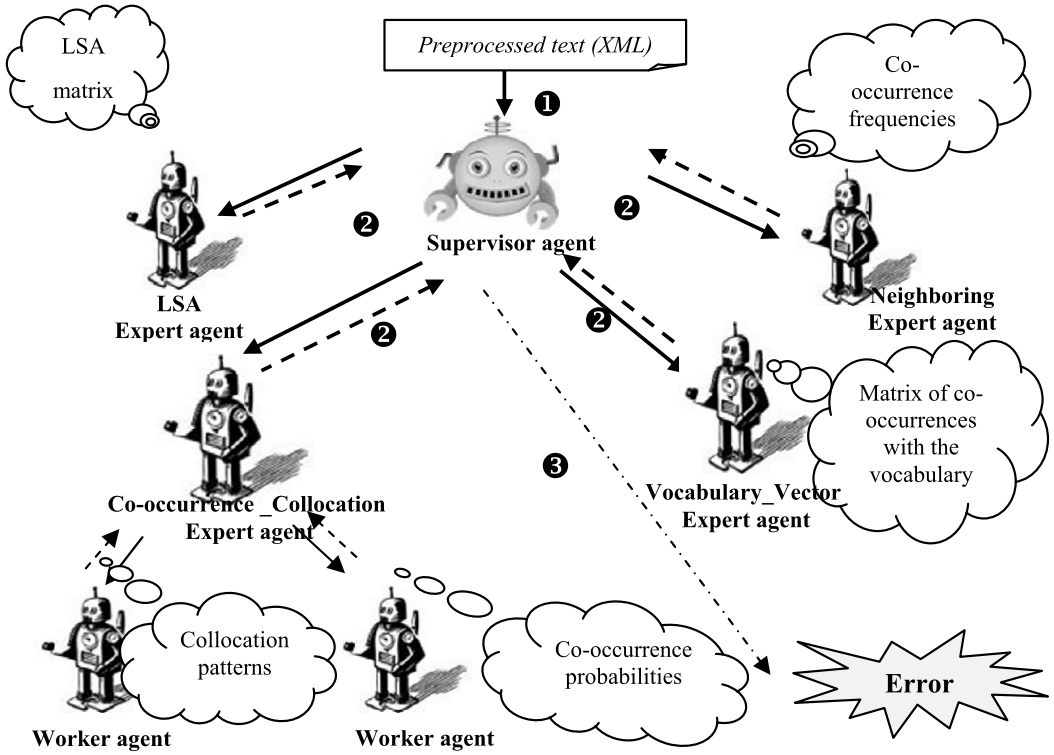


Fig. 4. MAS architecture for detecting semantic errors.

8.3. Resolution of conflicts

All the semantic expert agents aim to find suspicious words by consulting their context. They can therefore be contradictory and in conflict. For this reason we chose to set up a procedure for managing these conflicts based on the vote in order to maximize the robustness of the system. This procedure is applied by the *Supervisor* agent. Its role is to decide about the word to incriminate. Because our system of detecting semantic errors supposes one error at most per sentence and the proposed errors are sorted according to the semantic validity coefficient, we have opted for a *ranked choice voting* (only one winner from a set of ranked candidates). We present here briefly the principle of the method that we have adopted for the voting process.

(1) To count the number of occurrences of the different errors proposed by all the semantic agents that are present in each list and located at the top.

(2) To select the errors having the greatest number of occurrences. If only one error obtains the absolute majority, it is selected as the most likely error in the sentence. Otherwise, we calculate a new value of occurrences for errors in the next position.

(3) To keep repeating this process until to find a single error with an absolute majority of occurrences.

However, the proposed voting method may sometimes lead to a deadlock when the number of occurrences of errors selected in the first place remains invariant. In this case, the *Supervisor* agent considers the error given by the agent having the highest degree of confidence. Each expert agent has a counter that is incremented each time its proposal is retained by the voting system.

9. Experiments and discussion

The experiments we are going to describe in this section were performed in two stages: a learning stage and a test stage. In the learning stage we collected all statistical and linguistic information used by the different expert agents in order to apply their respective methods. The test stage was devoted to the evaluation of the performance of the system in detecting semantic errors.

9.1. Learning stage

The training data collected during this stage are as follows: the frequency of co-occurrences of lemmas (method 1), the probability of co-occurrences of lemmas and collocation patterns (method 2), the matrix of co-occurrences of lemmas with the vocabulary words (method 3) and the *LSA* matrix (method 4) (with $k = 300$, selected empirically). These data were

Table 3
Counts related to the training corpus.

Number of macro-sentences	26,823
Number of micro-sentences	82,219
Number of words	1,134,632
Number of different words	145,465
Number of different lemmas	30,305
Number of vocabulary terms	856

extracted from a pretreated textual corpus that we had previously built. This corpus is composed of economic articles of the Egyptian newspaper Al-Ahram[6] (2009–2010). Some information related to this corpus is presented in the table below:

9.2. Test stage

Texts containing genuine semantic errors are not easily available. Therefore, we simulated them by inserting deliberate errors. The test corpus was extracted from the same newspaper *Al-Ahram* (of the years 2009 and 2010) and represents 20% of the entire corpus used. It includes economic texts (like the training corpus), and contains 283,658 words. We applied the same procedure of artificial error insertion that Islam and Inkpen [21] used (see Section 2) by randomly introducing errors into the test corpus at a frequency of approximately one error in every 200 words. We obtained 1,398 semantic errors, varying in spelling from the original words with an edit distance of 1 (by the insertion, deletion, or replacement of a single character, or the transposition of two adjacent characters). The generation of these errors was done semi-automatically. We used our spelling checker [2] conceived initially for non-word errors. When it has a correct word as input, it provides automatically its spelling variations. However, the choice of the error was made manually to verify three conditions: the error must not yield a syntactically incorrect sentence; it must not be a function word; and it must cause semantic incoherence. Here is a sample of a malapropism inserted in our test corpus:

...ويتطلب المشروع قرضا من **الحنك** (البنك) الدولي...

/...wytTlb Alm$rwE qrDA mn **AlHnk** (Albnk) Aldwly.../

... and the project will require a credit from the international **palate** (bank)...

Besides, for the needs of comparison with Arabic baselines, we applied three baseline algorithms. All of them select one error at most one error per sentence. The first baseline was used also by Hirst and Budanitsky [18]. It is based on a random choice ("chance"), by flagging errors in the same proportion as they are expected to occur. Inspired from Golding [14], the second baseline selects the less frequent word within a macro-sentence, having a frequency inferior to a threshold of 10. The last baseline is based on trigram words (considering lemmas and with regard to macro-sentences). An error is flagged when it is not found with at least a similar trigram in the learning corpus. When there is more than one error, the less frequent word is selected. This baseline is quite similar to the method used by Verberne [32]. However, neither canonical forms nor sentence boundaries were considered in this latter. Moreover each word in an unattested trigram was considered as erroneous.

The evaluation of our system uses the two common metrics of Precision and Recall. The formulas are as follows:

$$\text{Precision} = \frac{\text{Number of errors correctly detected}}{\text{Total number of detections}}, \tag{6}$$

$$\text{Recall} = \frac{\text{Number of errors correctly detected}}{\text{Total number of introduced errors}}. \tag{7}$$

Table 4 summarizes the results for: the three baselines, each agent separately and the entire system (voting). This table shows that the detection system gave a precision rate of about 90% and a recall rate of about 83%. The achieved results are largely superior to the three baseline procedures. The first two baselines are very simple. They represent nonetheless an indicator of a minimal competency. The third one is less basic and its results are better but remain largely inferior to ours. However, let us note that, in general, the methods based on the N-gram model are used to detect real-word errors (not only semantic ones). Moreover, these methods are not very appropriate for morphologically rich and semi-free word languages. This is the case of Arabic, where, for three basic components Subject (S), Verb (V) and Object (O), one can have different combinations (VSO, VOS, and SVO) which are all correct.

We can notice also that the *Co-occurrence_Collocation* agent achieves the best precision (86.53%). The introduction of linguistic information has been beneficial. This seems to confirm the complementarity of the two phenomena: co-occurrence and collocation. Indeed, if we compare this agent with the *Co-occurrence* agent which is based only on the counting of

[6] The issues are extracted from the journal archives, available on: http://www.ahram.org.eg/.

Table 4
Evaluation of the semantic agents.

Agent	Precision (%)	Recall (%)	F-score (%)
Baseline 1	2.36	2.36	2.36
Baseline 2	27.49	38.84	32.19
Baseline 3	31.98	53.79	40.11
Co-occurrence	69.30	75.57	72.30
Co-occurrence_Collocation	86.53	79.66	82.95
Vocabulary_Vector	61.04	63.23	62.12
LSA	79.62	84.44	81.96
Voting	**90.55**	**82.73**	**86.46**

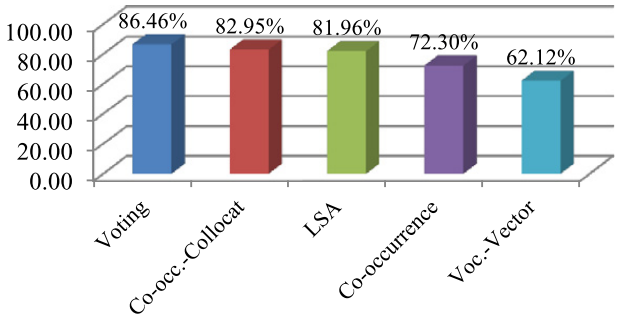


Fig. 5. Evaluation of the semantic agents sorted by F-measure.

co-occurrences in macro-sentences, we observe that this latter is less efficient even if it examines a larger scope of neighboring words. Besides, that *LSA* agent (79.62%) ranks second in the system, it is better than the *Vocabulary_Vector* agent which achieves a precision of about 61%. A better selection of the vocabulary words for the training corpus words would certainly improve these latter results. We also notice that the precision rate of the entire system (about 90%) is higher than the precision of each agent taken separately. Combining methods in order to increase the system performance, principally the precision rate, is thus confirmed according to this experiment. This can be intuitively explained merely because when several experts collaborate to make a decision, it is likely that the final result will be more accurate. However, we note that the global recall rate is lower (83%) than that one of the *LSA* agent. Obviously, keeping only one solution can cause inevitably more silence.

If we consider now the F-measure metric ($2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall})$), which is a good synthesis indicator, we find that combining the results of agents remains the highest performing approach (86.46%) (Fig. 5). Thus, as we hoped, the combination of several methods achieves globally more satisfactory results. We also notice that the probability of *Co-occurrence_Collocation* agent (82.95%) is globally better than the *LSA* agent (81.96%). The semantic similarity on which the *LSA* agent is based on is essentially lexical and the vector representation of a document is crude: It is an unstructured list of terms appearing in the text and the local relationships (including syntactic relations) are lost. Thus, one may want to use formal methods developed in the field of computational linguistics to analyze sentences and texts. These methods provide more structured semantic representations, but they are not appropriate for processing large corpora since they are very expensive and difficult to implement. A good compromise would be the use of both lexical features and simple syntactic features, as it the case for the *Co-occurrence_ Collocation* agent.

9.3. Example results

In this section, we give examples of situations in which the system succeeded and those in which it failed:

– *Example of success for all the agents*:

...تشهدت حركة تعاملات **الخرافة** (الصرافة) خلال الأسبوع الماضي ارتفاعا...

/...$hdt Hrkp tEAmlAt **AlxrAfp** (AlSrAfp) xlAl AlOsbwE AlmADy ArtfAEA.../

... Movement of trading on the **fable** (stock exchange) has observed during the last week a rise ...

This error was detected by all the agents since no relationship was found between the word "خرافة"/xrAfp/(fable) and its context.

– *Example of success for some agents*:

...أرجع البعض السبب إلى ارتفاع **أسفار** (أسعار) النفط مما خفف من القلق الذي أصاب المستثمر...

/... OrjE AlbED Alsbb IlY ArtfAE **OsfAr** (OsEAr) AlnfT mmA xff mn Alqlq Al*y ASAb Almstvmr .../

... Some ones have seen that the reason is the rise of petrol **travels** (prices), this has eased the concern of the investor ...

The error "أسفار"/OsfAr/(travels) was correctly detected by the whole system, but the agents did not all agree. Unlike the three other agents, the *Vocabulary_Vector* agent proposed instead the word "قلق"/qlq/(concern) as being the most likely error since no relationship was found with the vocabulary terms.

– *Example of failure, case of false positive error:*

... لضمان لحاق الاقتصاد **بركب** الانتعاش وبناء القوة الاقتصادية الحقيقية ...

/... lDmAn lHAq AlAqtSAd **brkb** AlAntEA$ wbnA' Alqwp AlAqtSAdyp AlHqyqyp .../

... to ensure the joining of the economy to the **convoy** of the revival and the building of the real economic strength ...

In this example, the expression "لحاق ... بركب"/lHAq ...brkb /(to join the convoy means to catch up with) is a non-contiguous collocation. It contains words unrelated to the context. Except the *Co-occurrence_Collication* agent, all the other agents suspected the word "ركب"/ rkb /(convoy) as being the most likely semantic error since it had no relationship with the context.

– *Example of failure, case of false negative error*:

...قيمة **العروض** (القروض) التي تم صرفها لتمويل المشاريع الصغيرة...

/... qymp **AlErwD** (AlqrwD) Alty tm SrfhA ltmwyl Alm$AryE AlSgyrt .../

... value of the **propositions** (loans) that has been disbursed to finance small projects ...

The error عروض/ErwD/(propositions) was not detected by all the agents. It was not connected to the near word صرف/Srf/(to disburse), but it was found to be related to تمويل/tmwyl/(to finance) and مشاريع/m$AryE/(projects).

9.4. Comparison with related work

It would be interesting to compare our results with others' ones. Unfortunately, there are no previous works (that we know of) about the detection of semantic errors in Arabic texts. Therefore, we try here to make somehow meaningful comparisons with related work for other languages although this would be difficult since the experimental conditions (e.g. the treated language, the type of texts and the errors) are not the same. First, we consider approaches using confusion sets. The precision rate reported in the literature is superior to ours and it can reach 99% (see Section 2). The main disadvantage of these methods is that they are limited to only common errors which are described by the confusion sets. Therefore, uncommon errors and typing errors are not considered. Their advantage over our approach is that they can handle function word errors simply by considering confusion sets such as {than, then}.

With regard to the second type of methods (not using confusion sets), we can compare our results especially to those of [18,36] and [21] since our evaluation was carried out on a test data containing semantic errors with spelling variation,[7] as they did, with roughly the same size and the same density of errors (1 error in approximately 200 words). When considering the F-measure rate, we note that the result achieved by our MAS system is superior (86%) than the best of them [21] with an F-measure rate of 59% for detection. At first sight, these good results can be explained by the assumption that we have made about one error at most per sentence, which causes inevitably less false positives and subsequently better precision. In return, however, this advantage is balanced by the elimination of errors candidates which may engender more false negatives. It has been the case, for our detection system since it achieved a better precision but a worse recall than the best of these works (89%). However, let us stress that a straightforward comparison is not really possible because of the different dataset used and the different language.

10. Conclusion

In order to detect semantic errors in Arabic texts, we proposed an approach combining different methods based on statistical and linguistic information within a MAS. The evaluation system showed good performance compared to related

[7] Our detection system is not restricted to semantic errors with spelling variation. It can also detect semantic errors that are phonetically or orthographically different to the intended word.

work, considering both precision and recall. The combination of methods achieved globally better results and helped to overcome the limitations of each method working separately. Additional tests are in prospect to analyze other types of texts and to treat other languages such as English.

However, we think that these results can still be enhanced, by increasing the size of the learning corpus and by tuning the methods we used. The developed system can also be improved, for example, by detecting vowelization errors. As another extension to this work, we aim to investigate correction. The complexity of this task would certainly depend on the assumptions made about the causes of the errors. If one assumes that a semantic error is caused by keyboard slips or the writer's ignorance of the correct spelling, a simple spelling checker would be sufficient to find candidate corrections. By giving it a correct word instead of a non-word error, a spelling checker can also play the role of a spelling variation generator. In return, when the error is supposed to be due to the writer's ignorance of the exact meaning (as in the case of "member" and "peace"), the correction task would be more complicated. It would find words that bear a semantic similarity to the error.

References

[1] C. Aloulou, Un modèle multi-agent pour l'analyse syntaxique de la langue arabe, PHD Thesis, La Manouba University, Tunisia, 2005.
[2] C. Ben Othmane, A. Zribi, Algorithmes pour la correction des erreurs orthographiques en Arabe, in: 6ème Conférence sur le Traitement des Langues Naturelles, Corse, France, 1999.
[3] C. Ben Othmane Zribi, M. Ben Ahmed, Efficient automatic correction of misspelled Arabic words based on contextual information, Lecture Notes in Computer Science, vol. 2773, Springer, 2003, pp. 770–777.
[4] C. Ben Othmane Zribi, F. Ben Fraj, M. Ben Ahmed, Un système multi-agent pour la détection et la correction des erreurs cachées en langue Arabe, in: Actes de la 5ème conférence sur le Traitement Automatique des Langues Naturelles, Dourdan, France, 2005, pp. 143–153.
[5] C. Ben Othmane Zribi, H. Mejri, M. Ben Ahmed, Combining methods for detecting semantic hidden errors in Arabic texts, Lectures Notes in Computer Science, vol. 4394, Springer, 2007, pp. 634–645.
[6] S. Bergsma, D. Lin, R. Goebel, Web-scale N-gram models for lexical disambiguation, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, California, USA, 2009, pp. 1507–1512.
[7] S. Bergsma, E. Pitler, D. Lin, Creating robust supervised classifiers via web-scale N-gram data, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Sweden, 2010, pp. 865–874.
[8] T. Buckwalter, Buckwalter Arabic morphological analyzer, in: Linguistic Data Consortium (LDC2002L49), 2002.
[9] A.J. Carlson, J. Rosen, D. Roth, Scaling up context-sensitive text correction, in: Proceedings of 13th Conference on Innovative Applications of Artificial Intelligence IAAI'01, Washington, USA, 2001, pp. 45–50.
[10] L. Erman, F. Roth, V. Lesser, R. Reddy, The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty, ACM Computing Surveys 12 (1980).
[11] A. Farghaly, Computer processing of Arabic script-based languages: current state and future directions, in: Workshop on Computational Approaches to Arabic Script-based Languages, Switzerland, 2004.
[12] W. Gale, K.W. Church, D. Yarowsky, Discrimination decisions for 100,000 dimensional spaces, in: Current Issues in Computational Linguistics, In Honour of Don Walker, Kluwer Academic Publishers, 1994, pp. 429–450.
[13] D. Genethial, J. Courtin, J. Ménézo, Distributing and porting general linguistic tools, International Conference on Computational Linguistics, Denmark, 1996.
[14] A.R. Golding, A Bayesian hybrid method for context-sensitive spelling correction, in: Proceedings of the 3rd Workshop on Very Large Corpora, Massachusetts, USA, 1995, pp. 39–53.
[15] A.R. Golding, Y. Schabes, 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, pp. 71–78.
[16] A.R. Golding, D. Roth, A winnow-based approach to context-sensitive spelling correction, Machine Learning Journal 34 (1–3) (1999) 107–130.
[17] H. Haddad, H. Ben Ghezala, M. Ghnima, Conception d'un catégoriseur morphologique fondé sur le principe d'Eric Brill dans un contexte multi-agents, in: 26th Conference on Lexis and Grammar, Bonifacio, 2007.
[18] G. Hirst, A. Budanitsky, Correcting real-word spelling errors by restoring lexical cohesion, Natural Language Engineering 11 (2005) 87–111.
[19] K. Ingason, S.B. Jóhannsson, S. Helgadóttir, H. Loftsson, E. Rögnvaldsson, Context-sensitive spelling correction and rich morphology, in: Proceedings of the 17th Nordic Conference of Computational Linguistics, Denmark, 2009, pp. 231–234.
[20] D. Inkpen, A. Désilets, Semantic similarity for detecting recognition errors in automatic speech transcripts, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, 2005, pp. 49–56.
[21] A. Islam, D. Inkpen, Real-word spelling correction using Google Web 1T 3-grams, in: Proceedings of the 9th Conference on Empirical Methods in Natural Language, Singapore, 2009, pp. 1241–1249.
[22] F. Jalabert, M. La Fourcade, Nommage de sens à l'aide des vecteurs conceptuels, 14ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle, Toulouse, France, 2004.
[23] M.P. Jones, J.H. Martin, Contextual spelling correction using latent semantic analysis, in: Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, 1997, pp. 166–173.
[24] S. Kulczynski, Die Pflanzenassoziationen der Pieninen, Bulletin International de l'Académie Polonaise des Sciences et des Lettres, Classe des Sciences Mathématiques et Naturelles, Série B 2 (1927) 57–203.
[25] T.K. Landauer, P.W. Foltz, D. Laham, An introduction to latent semantic analysis, Discourse Processes 25 (1998) 259–284.
[26] L. Mangu, E. Brill, Automatic rule acquisition for spelling correction, in: Proceedings of the 14th International Conference on Machine Learning, Nashville, 1997, pp. 734–747.
[27] M.A. Mokrane, Représentation de collections de documents textuels: application à la caractérisation thématique, PHD Thesis, Montpellier II University, 2006.
[28] S. Nogry, Comment faire apprendre des connaissances abstraites à partir d'exemples: application au projet Ambre, Colloque EIAH, Strabourg, 2003, pp. 67–70.
[29] G. Sabah, CARAMEL: A computational model of natural language understanding using a parallel implementation, in: Proceedings of the 9th European Conference on Artificial Intelligence, Stockholm, Sweden, 1990, pp. 563–565.
[30] A. Sarma, D.D. Palmer, Context-based speech recognition error detection and correction, in: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Boston, 2004, pp. 85–88.
[31] H. Schutze, Automatic word sense discrimination, Journal of Computational Linguistics 24 (1998) 97–123.

[32] S. Verberne, Context sensitive spell checking based on word trigram probabilities, Master thesis, Taal, Spraak & Informatica, Nijmegen University, 2002.
[33] K. Voll, S. Atkins, B. Forster, Improving the utility of speech recognition through error detection, Journal of Digital Imaging (2007) 1–7.
[34] K. Warren, Gestion de conflits dans une architecture multi-agents d'analyse automatique de textes, PHD Thesis, Stendhal University, Grenoble, 1998.
[35] C. Whitelaw, B. Hutchinson, G.Y. Chung, G. Ellis, Using the web for language independent spellchecking and autocorrection, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, 2009, pp. 890–899.
[36] A. Wilcox-O'Hearn, G. Hirst G, A. Budanitsky, Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model, in: Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 605–616.
[37] W. Xu, J. Tetreault, M. Chodorow, R. Grishman, L. Zhao, Exploiting syntactic and distributional information for spelling correction with web-scale N-gram models, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Scotland, 2011, pp. 1291–1300.
[38] P. Zweigenbaum, B. Bachimont, J. Bouaud, M. Cavazza M, L. Dore HELENE, Compréhension de comptes-rendus d'hospitalisation. Deuxième école d'été sur le traitement des langues naturelles, Lanion, 1989.