International Journal of Electrical and Computer Engineering (IJECE)
Vol. 5, No. 6, December 2015, pp. 1458~1467
ISSN: 2088-8708

Ontologies and Bigram-based Approach for Isolated Non-word Errors Correction in OCR System

Aicha Eutamene*, Mohamed KhireddineKholladi, Hacene Belhadef***
* IFA Department, NTIC Faculty, University of Constantine 2 – ABEDLHAMID MEHRI, Algeria
** Computer Science Department, El Oued University, Algeria

Article Info

Article history:

Received Jul 15, 2015
Revised Sep 13, 2015
Accepted Sep 30, 2015

Keyword:

Bigram
OCR
Ontology
Spelling Correction
Word Recognition
WordNet

ABSTRACT

In the present paper, we describe a new and original approach for post-processing step in optical character recognition systems (OCR). This approach is based on a new method of spelling correction to correct automatically misspelled words resulting from character recognition step of scanned documents by combining both ontologies and bigram code in order to create a robust system able to solve automatically the anomalies of classical approaches. The proposed approach is based on a hybrid method which is spread over two stages, first one is character recognition by using the ontological model and the second one is word recognition based on spelling correction approach based on bigram codification for detection and correction of errors. The spelling error is broadly classified in two categories namely non-word error and real-word error. In this paper, we interested only on detection and correction of non-word errors because this is the only type of errors treated by an OCR. In addition, the use of an online external resource such as WordNet proves necessary to improve its performances.

Corresponding Author:

Hacene Belhadef,
IFA Department, NTIC Faculty
Universityof Constantine 2 – ABEDLHAMID MEHRI,
Constantine 25000, Algeria
Email: hacene.belhadef@univ-constantine2.dz

1. INTRODUCTION

With the advent of digital optical scanners, a lot of paper-based books, textbooks, magazines, articles, and documents are being transformed into an electronic version that can be manipulated by a computer. For this purpose, OCR, short for Optical Character Recognition was developed for handwriting recognition (HWR). It consists to translate scanned graphical text into editable computer text [1], [2].

OCR (Optical Character Recognition) means that a computer analyses character images automatically to achieve the text information, and OCR engine is an ICR (Independent Character Recognition) engine usually. It recognizes characters in every position of an image and gives all the possible candidate results. In ICR recognition process, only image information is used, and the results contain many incorrect words. So the spelling correction approaches, which play the most important role in OCR post-processing, are required [3].

The task of word recognition is to find all the valid entries of the dictionary that match the recognized word. To do that, one or several candidates could be generated for each word. Each candidate is also provided with a confidence score. The post-processing task is to pick the best candidate for each word by exploiting different types of information. The post-processing step helps to significantly reduce errors. However, it is not a completely separate of preceding steps. In general, this step shows the unrecognized characters and words that have not been found in the dictionary or in the lexicon. This step provides the user to make lexical and syntactic corrections using dictionaries and grammar rules to improve the recognition

rate [4], several approaches have been resorted to playing a consolidation role in this step. In our case, we focus on correction of isolated words. The use of external resources such as lexical ontologies proves necessary to improve the quality of the results. The ontologies play an important role in the field of knowledge representation and sharing of information, they can be modified, adapted and reused in different applications and domains. In recent years, ontologies have been shown to be an important instrument and useful for the representation, sharing and reuse of knowledge, also thanks to ontology's languages which express a rich semantic and provide best reasoning capabilities. This is why we opted for the use of ontology as a tool and a way to design a new recognition approach of characters and words.

Indeed, we found that whatever of the wealth of documents content, this wealth remains insufficient to help the process in order to accomplish its task. The approaches that exist currently do not take into account the semantic aspect which isn't explicitly visible for the machine, thing that oriented us to the use of ontology to complete this gap (called semantic gap).

The ontology has been applied in the character recognition step which we will describe later. On the other hand in the post-processing step, we adopted a spelling checking approach by using the bigram codification for detection and identification of errors in misspelled words.

The task of spell-checking is split into two parts, error-detection, and error-correction. There are error detection complex systems which may be used to detect words that are correctly spelled, but are unsuitable in the syntactic or semantic context; this is referred to as real-word error detection in context [5].

The word error is a major hindrance to the real world applications. In textual documents, word-error can be of two types. One is non-word error which has no meaning and other is real word error which is meaningful but not the intended word in the context of the sentence. Of these, non-word has been widely studied and algorithms to detect and suggest correction word for the error have been proposed. These algorithms are generally termed as spell-checker, which are integrated in various word processing software like Microsoft Word, Libre Office Writer ,Ispell, Aspelletc [6].

The research into algorithmic techniques for detecting and correcting spelling errors in the text has a long history in computer science. As an amalgamation of the traditional fields of artificial intelligence, pattern recognition, string matching, computational linguistics, and others, this fundamental problem in information science has been studied from the early 1960's to the present [7].

In this paper, we are interested in detecting and identification of errors in the misspelled word independently of its context, so-called isolated non-real word. To do that, we proposed a new method based on a general lookup dictionary and the specific lexicon of a domain. If the word does not exist, it will be codified in bigram code in order to facilitate the detection and identification the error in the word. A set of candidates will be generated by applying a set of rules that match the type of error detected. For more details, the reader can refer to the section that describes our method.

In section 2 of this paper, we present short definition of the spelling correction and its types. In Section 3, we will review some related works. We present our method in section 4, and we finish by a general conclusion.

2. SPELLING CORRECTION

Spell checking is the process of finding misspelled words in a written text and possibly to correct them. This problem has been widely studied, and spell-checkers are probably among the first successful NLP applications widely used by the general public [8].

Spelling correction is the task of correcting words in texts. Most of the available spelling correction tools only work on isolated words and compute a list of spelling suggestions ranked by edit-distance, letter-n-gram similarity or comparable measures. Although the probability of the best-ranked suggestion being correct in the current context is high, user intervention is usually necessary to choose the most appropriate suggestion. According to Kukich [6], the problem of spell checking can be classified in three categories of increasing difficulty: non-word error detection, isolated-word error correction, and context-dependent word correction.

We can classify spelling errors in two main groups: non-word errors and real-word errors. The non-word errors might be corrected without considering the context in which the error occurs, but a real-word error can be corrected only by taking context into account [9].

1) ***Non-word*** error occurs when a word in the OCR text is interpreted as a string that does not correspond to any valid word in a given word list (lexicon) or dictionary. For example, The*bok* is on the table.

The word *bok* does not exist in English, and it probably derives from a typo of the noun *book*. The correct phrase is: The *book* is on the table.

2) ***Real-word*** error occurs when a source-text word is interpreted as a string that actually does occur in the dictionary, but is different from the source-text word. For example, I saw *tree* trees in the park.

The noun *tree* exists in English, but in this context it is most likely a typo of the numeral *three*: I saw *three* trees in the park.

We recall that in this study presented in this paper, we are only interested in the first type of error.

3. RELATED WORK

The task of word error detection in OCR processing is often considered trivial or solved in many research papers dealing with spelling correction:

Xiang Tong and David A. Evans, describe an automatic, context-sensitive, word-error correction system based on statistical language modelling (SLM) as applied to optical character recognition (OCR) post-processing. Given a sentence to be corrected, the system decomposes each string in the sentence into letter n-grams and retrieves word candidates from the lexicon by comparing string n-grams with lexicon-entry n-grams. The retrieved candidates are ranked by the conditional probability of matches with the string, given character confusion probabilities. The word-bigram model and Viterbi algorithm are used to determine the best scoring word sequence for the sentence. In addition, the system can learn the character confusion probabilities for a specific OCR environment and use them in self-calibration to achieve better performance [10].

Mohammad Ali Elmi and Martha Evens, describe a spelling correction system that functions as part of an intelligent tutor that carries on a natural language dialogue with its users. The basic idea of their approach is the interaction between the parser and the spelling corrector. Alternative correction targets are fed back to the parser, which does a series of syntactic and semantic checks, based on the dialogue context, the sentence context, and the phrase context [11].

DavideFossati and Barbara Di Eugenio, address the problem of real-word spell checking, i.e., the detection and correction of typos that result in real words of the target language. Authors propose a methodology based on a mixed trigrams language model. Their experiments show promising results with respect to the hit rates of both detection and correction, even though the false positive rate is still high [8].

Sebastian Deorowocz, MARCIN G. CIURA, Account for a new technique of correcting isolated words in typed texts. A language-dependent set of string substitutions reflects the surface form of errors that result from vocabulary incompetence, misspellings, or mistypings. Candidate corrections are formed by applying the substitutions to text words absent from the computer lexicon. A minimal acyclic deterministic finite automaton storing the lexicon allows quick rejection of nonsense corrections, while costs associated with the substitutions serve to rank the remaining ones [12].

Martin Schierle, Sascha Schulz and Markus Ackermann, have developed an efficient context sensitive spelling correction system called dcClean by combining two approaches: the edit distance based ranking of an open source spelling corrector and neighbour co-occurrence statistics computed from a domain specific corpus. In combination with domain specific replacement and abbreviation lists, they are able to significantly improve the correction precision compared to edit distance or context based spelling correctors applied on their own [13].

Youssef Bassil and Mohammad Alwani propose a post-processing context-based error correction algorithm for detecting and correcting OCR non-word and real-word errors. The proposed algorithm is based on Google’s online spelling suggestion which harnesses an internal database containing a huge collection of terms and word sequences gathered from all over the web, convenient to suggest possible replacements for words that have been misspelled during the OCR process. According to the authors, the experiments carried out revealed a significant improvement in OCR error correction rate and they can improve upon the proposed algorithm so much so that it can be parallelized and executed over multiprocessing platforms [1].

Bakkali Hamza et al., have proposed a new approach for spell-checking errors committed in the Arabic language. The authors introduced the concept of morphological analysis in the process of spell-checking. Their system uses a stems dictionary of reduced size rather than exploiting a large dictionary not covering the all Arabic words. According to the authors, the obtained results are highly positive and satisfactory [14], [15].

Golding and Schabes [16] have introduced a hybrid approach called "Tribayes" combining Trigram and Bayes' method. Trigram method uses part-of-speech trigrams to encode the context whereas Bayes is a feature-based method. They use two types of features: context word and collocations. Later Golding with Roth [16] have proposed a Winnow-based method for real word detection and correction. They modified the previous method by applying a winnow multiplicative algorithm combining variants of winnow and weighted majority voting and achieved better accuracy.

Hirst and Budanitsky [18] made a study of the problem of spelling correction on same corpus of Wall Street Journal. Their method identifies tokens that are semantically unrelated to their context and was not restricted to checking words from predefined confusion set. They achieved Recall of 23%-50% and Precision of 18%-25%.

Emillia and al. [19] proposed a hybrid method for isolated word recognition based on the Ergodic Hidden Markov Model and they used the genetic algorithm to optimize the Baum-welch method in the training process in order to improve the accuracy of the recognition result which is produced by the HMM parameters that generate the low accuracy when the HMM are tested.

In another WordNet-based approach, Peddler [20] showed that the semantic association can be useful in detecting a real-word error using some confusion sets especially in case of Dyslexic text. She achieved recall and precision of correction 40% and 81%, respectively for Dyslexic text.

4. OUR CONTRIBUTION

Figure 1 shows the different component of our system which contains two main modules: Character Recognition (first processing unit) and Word Recognition (second post-processing unit). In the first unit, the system receives a sequence of graphemes resulting from a pre-processing step which consists in segmentation of text document's image in order to generate a sequential series of graphemes according their position in the document.

In this unit called OMCR (Ontological Model for Character Recognition), the system recognizes document's characters by basing on an ontological model [21]-[23]. The concepts of ontology represent the graphemes, by contrast the relationships represent the spatiality between the graphemes by respecting their order of appearance (Figure 2 describes all these steps).

The two units of our system have a complementary role. The first one generates the characters one by one by forming the lexical tokens (set of words separated by blank). These tokens are considered as the input of the second unit which consists in the checking and correction in order to generate an accepted word by the user.

Figure 1 shows the use of two types of ontologies, the domain ontology that we have created to model the textual structure of a document and which consists to supervise the character recognition process. The second is a lexical ontology called WordNet which is used as dictionary to verify the existence of token in the language vocabulary.

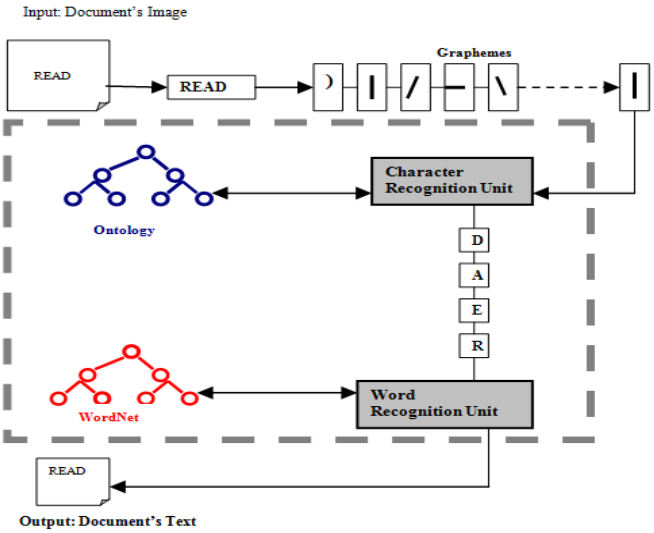

Figure 1. System Overview

4.1. Processing Step of Character Recognition

The step of character recognition is based on creation of ontology. It describes the vocabulary of processed documents and the relationships that exist between the elements of this vocabulary. This step is well described in Figure 2.

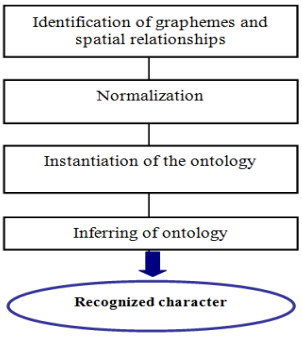

Figure 2. Processing unit overview

It is composed offour stages, in the beginning we must start by the extraction of graphemes and spatial relationships between them. See Figure 3 to have an idea on different types of relationshipimplemented in "Protégé" environment for ontologies development.

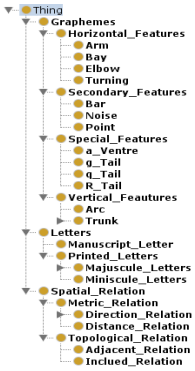

Figure 3. Taxonomy of graphemes and relationships

In the case of a manuscript script, a step of normalization proves necessary for clustering and identification of grapheme's class. Once we identify these elements, we proceed to the instantiation of the domain ontology in order to identify the graphemes which are connected by spatial relationships. These last will develop a character that can be recognised by applying a rule of SWRL type (see rule below).

Tronc(?X1) and Bar(?X2)→CharL (?X) and posLetter(i)

Example of SWRL rule to identify the letter '**L**'

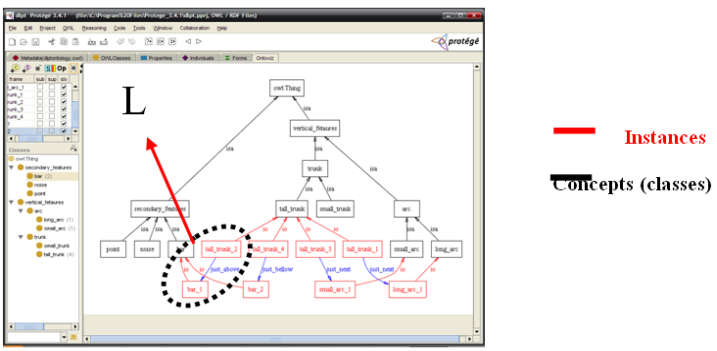

Figure 4. Recognition of the letter ***L*** by our ontology

Figure 4 shows the instantiation of domain ontology by different graphemes found in the document; it shows also the two graphemes and the relationship between them which forms the letter ***L***.

4.2. Post-processing Step: Word's Recognition and Reconstruction

Post-processing is the last stage of an OCR system whose purpose is to detect and to correct spelling errors resulting from the inability of the system to properly recognize the characters. This stage plays a role of recognition and reconstruction of words; it helps to reduce significantly the errors. However, this isn't completely separate stage of the preceding one, but it plays a complementary role. It comprises several steps which can be performed on the recognizer output, very often dictionaries or even word's lexicons are used to improve the recognition result by using appropriate similarity measures. In this paper, we propose a new approach for detecting and correction of misspelled words. The algorithm that represents this approach comprises several steps to be executed in order to correct misspelled word. The flowchart for the algorithm is shown in Figure 5 that summarizes the different steps of the proposed algorithm. Each of these components will be discussed in the following sections.

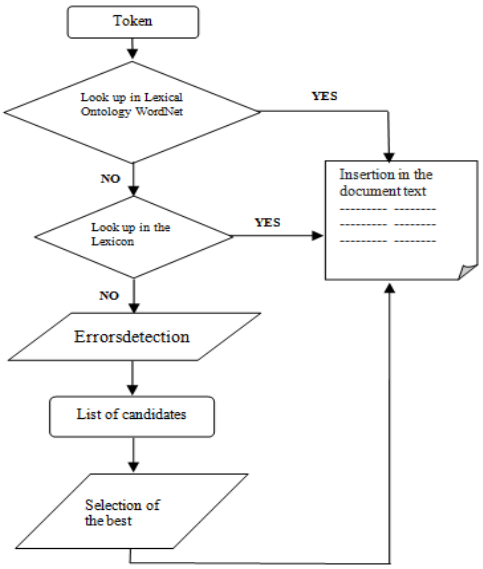

Figure 5. Flowchart of the post-processing process

4.2.1. Checking and Looking Up

The input of this step is a Token, this last is a string previously generated by the character recognition process. Firstly, we must check the existence of this Token in a standard dictionary, in our case we preferred to use the online lexical ontology (WordNet) as external resource because it is open access and regularly updated. Figure 6 shows a java programto check the existence of the word "*write*" in the WordNet dictionary.

```
publicclassWordNet {
publicstaticvoidmain(String[] args) {
NounSynsetnounSynset;
Synset[] synsets = database.getSynsets("Write", SynsetType.NOUN);
for(inti = 0; i<synsets.length; i++) {
        nounSynset = (NounSynset) synsets[i];
        System.out.println("Définition "+ i + " : "+ nounSynset.getDefinition());
        System.out.println("Synonyms of words : ");
        for(String syn : nounSynset.getWordForms())
                System.out.println("    "+ syn);
}}}
```

Figure 6. Example of Java program for access and lookup in WordNet

If the token exists in the dictionary, so it belongs to the vocabulary of the writing language of the document and it will be considered as a real word well recognized by our system and an insert operation will be triggered in order to insert it in the output file. Otherwise, if the token does not exist in the standard dictionary, we pass to another checking step prior to its correction.This second level of checking is optional and it consists to verify the existence of this Token into a specific lexicon (or specific domain ontology) related to subject dealt in the document (eg. medical, chemical, astronomical, historical, etc.).If the token does not exist yet, so it is considered as a non-real word and we pass to correct it in order to generate its corresponding real-word.

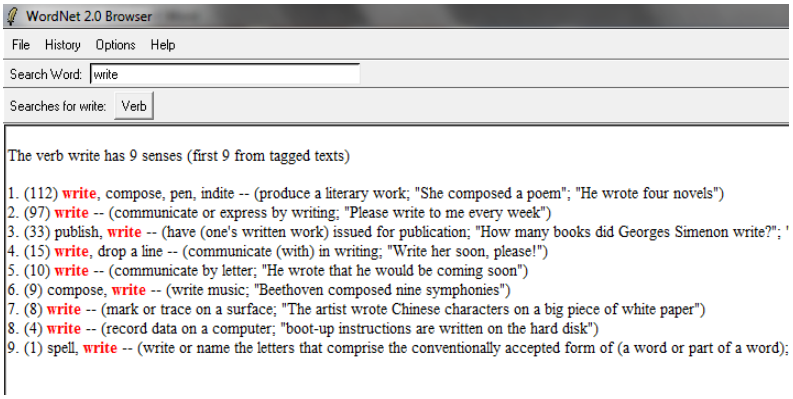

Figure 7. Graphical interface for lookup in WordNet

4.2.2. Candidates Generation

In this step, we generate a set of candidates for the misspelled word which has been detected during the checking step. In the output, we will have a set of words generated by applying some transformation rules (see table 3).

In order to facilitate the detection and identification of errors, we propose a new codification of the misspelled word based on bigram code that consists to transform a word to a pair of consecutive letters with a space character in the first and the last position. Table 1 shows an example of a misspelled word and its bigram code, and table2 shows the structure of such a codification.

Table 1. Bigram code

Correct word	Misspelled word	Bigram code
learn	learii	(-l, le, ea, ar, ri, ii, i-)

Table 1 shows that the word "learn" hasn't been recognized correctly. In this case the error lies in the false recognition of character "n"; it has been recognized by both letters "ii". Table 3 shows some classes of errors that we have treated.

Table 2. Data structure of Bigram's codification of misspelled word '**learii**'

Bigram's number	Bigram code	Absolute value of difference between one bigram	Rank of the second letter in the one bigram
1	-l	15	12
2	le	7	5
3	ea	4	1
4	ar	16	18
5	ri	9	8
6	ii	0	8
7	i-	18	27

Table 3. Mapping rules for errors correction

Error		Correction
ii	→	n
ii	→	u
ii	→	o
ii	→	v
rn	→	m
vv	→	w
cl	→	d
iii	→	m
b	→	h

4.2.2.1. Errors Detection

The codification shown in Table 2 allows detecting errors in the misspelled word. For example, if our system detect a zero in the third column of bigram codification (see Table 2), it automatically realizes that this is an error of one of the types: aa, bb, ... ii ... zz. We can also detect some other types of errors.

4.2.2.2. Error Identification

Only, by consulting the corresponding value in the fourth column that we can detect the exact error. The example of Table 2 shows an error of this type. Here, is about the letter 'i' which has the rank 8 in the alphabet (see Table 4).

Table 4. Rank of letters in the alphabet

1	2	3	…	…	8	…	…	26	27
a	B	c	….	….	i	…..	….	z	- (Space letter)

In this case, our system generates a set of candidate of the misspelled word by replacing the error by its corresponding rule(s).

For the misspelled word 'learii' shown in Table 2, we can generate four candidates by applying the first four rules: (learn, learu, learo, learv).

4.2.3. Candidates Selection

After generating the candidates, the system must select only one of them,the one that has been generated by applying the rule of high priority. For example, in Table 3, the rule (ii → n) has a high priority compared to the other rules of the same type of error.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a new approach that consolidates the work of an OCR system, this approach is based on two main steps; the first one consists to character recognition by using a domain ontology which models the document's language. The recognition is done by instantiation and inference of the ontology by using type SWRL's rules. The second step completes the first one by correcting the possible errors obtained from the previous step. It takes as input the recognized characters by forming them in groups (token or lexical item). It detects the existence of the possible errors on the word by checking its existence in the dictionary or in the domain lexicon. If this is not the case, a correction operation will be launched to correct these errors by using a bigram code that supports the detection and correction of words.

In the future work, we envisage to generalize our approach so that it can deal the real-word by taking into account the semantic and contextual information defined in the ontology.

ACKNOWLEDGEMENTS

The authors would like to thank the guest editors andanonymous reviewers for their valuable and constructive comments.

REFERENCES

[1] Bassil Youssef and Alwani Mohammad, "OCR post-processing error correction algorithm using google online spelling suggestion", 2012.

[2] Olakanmi Olufemi Oladayo, "Optical Character Recognition of Off-Line Typed and Handwritten English Text Using Morphological and Template Matching Techniques", *IAES International Journal of Artificial Intelligence*, Vol. 3, No. 3, 2014.

[3] Zhuang Li Bao, Ta Zhu Xiaoyan, *et al.*, "A Chinese OCR spelling check approach based on statistical language models", *In: Systems, Man and Cybernetics, 2004 IEEE International Conference on. IEEE*, pp. 4727-4732, 2004.

[4] MaedaJunji, Iizawa Takuya, IshizakaTohru, *et al.*, "Segmentation of natural images using anisotropic diffusion and linking of boundary edges", *Pattern Recognition*, Vol. 31, No. 12, pp. 1993-1999, 1998.

[5] Pirinen Tommi A., and Krister Lindén. "State-of-the-art in weighted finite-state spell-checking", *Computational Linguistics and Intelligent Text Processing. Springer Berlin Heidelberg*, pp. 519-532, 2014.

[6] Karen Kukich, "Techniques for Automatically Correcting Words in Text", *ACM Computing Surveys*, Vol. 24, No. 4, pp. 377-439, 1992.

[7] P. Samanta, *et al.*, "A simple real-word error detection and correction using local word bigram and trigram", *In Proceedings of ROCLING*, 2013.

[8] Fossati D., *et al.*, "A mixed trigrams approach for context sensitive spell checking", *In Computational Linguistics and Intelligent Text Processing"*, Springer Berlin Heidelberg, pp. 623-633, 2007.

[9] Agarwal, *et al.*, "Utilizing Big Data in Identification and Correction of OCR Errors", *UNLV Theses/Dissertations/Professional Papers/Capstones*, pp. 1914, 2013.

[10] Tong X., *et al.*, "A statistical approach to automatic OCR error correction in context". *The fourth workshop on very large corpora*, pp. 88-100, 1996.

[11] Elmi M. A., *et al.*, "Spelling correction using context", *The 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pp. 360-364, 1998.

[12] Deorowicz Sebastian, and Marcin G. Ciura, "Correcting spelling errors by modelling their causes". *International Journal of Applied Mathematics and Computer Science*, Vol. 15, No. 2, pp. 275-285, 2005.

[13] Schierle, *et al.*, "From spelling correction to text cleaning–using context information", *In Data Analysis, Machine Learning and Applications*, Springer Berlin Heidelberg, pp. 397-404, 2008.

[14] Bakkali Hamza, *et al.*, "For an Independent Spell-Checking System from the Arabic Language Vocabulary." *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 5, No. 1, 2014.

[15] Hicham Gueddah, "Introduction of the weight edition errors in the Levenshtein distance", 2012

[16] A. R. Golding, *et al.*, "Combining Trigram-based and Feature-based Methods for Context sensitive Spelling Correction", *The 34th Annual Meeting of the Association for Computational Linguistics*, pp. 71-78, 1996.

[17] Golding Andrew R., and Roth Dan, "A winnow-based approach to context-sensitive spelling correction". *Machine learning*, Vol. 34, No 1-3, pp. 107-130, 1999.

[18] Hirst Graeme, and Budanitsky Alexander, "Correcting real-word spelling errors by restoring lexical cohesion", *Natural Language Engineering*, Vol. 11, No 01, pp. 87-111, 2005.

[19] Rizkha Emillia Nyoman, Suyanto N. F. N, Maharani Warih, "Isolated word recognition using ergodic hidden markov models and genetic algorithm", *Telkomnika*, Vol. 10, No. 1, pp. 129-136, 2012.

[20] Pedler Jennifer, "Using semantic associations for the detection of real-word spelling errors", *In: Proceedings from the Corpus Linguistics Conference Series*, 2005.

[21] Eutamene Aicha, Kholladi Mohamed Khireddine, and Belhadef Hacene, "Ontological Model For Character Recognition Based On Spatial Relations", *Signal & Image Processing: An International Journal (SIPIJ)*, Vol. 04, No. 03, pp. 113- 124, 2013.

[22] Belhadef Hacene, Kholladi Mohamed Khireddine, and Eutamene Aicha, "Ontology of Graphemes for Latin Character Recognition", *Procedia Engineering*, Vol. 24, pp. 579-584, 2011.

[23] Eutamene Aicha, Belhadef Hacene, and Kholladi Mohamed Khireddine, "New process ontology-based character recognition", *In: Metadata and Semantic Research*. Springer Berlin Heidelberg, pp. 137-144, 2011.

BIOGRAPHIES OF AUTHORS

Eutamene Aicha was born in Constantine, Algeria. In 2009, she received the Master degree in computer engineering from Mentouri university of Constantine. Actually she is Ph.D. student in NTIC faculty in the Constantine 2 University in Algeria. Her research interests include Pattern recognition, digital image processing and Ontologies.

Mohamed Khireddine Kholladi was born in Constantine, Algeria. In 1991, he received the Ph.D degree in computer engineering from Claud Bernard University of Lyon-France. Actually, he is a Professor at the computer science department at the University El Ouad in Algeria. His research interests include Geographical Information System, Complex system, image processing.

Belahdef Hacene was born in Constantine, Algeria. In 2010 he received the Ph.D degree in computer science from the Mentouri University of Constantine and in 2013 he received HDR degree (Lecturer) from the NTIC faculty in the University of Constantine 2. Actually, he is an associate professor at the same university. His research interests include Ontologies field, Image Processing, Artificial intelligence and Data Mining.