

# Correcting optical character recognition result via a novel approach

Otman Maarouf<sup>1</sup>, Rachid El Ayachi<sup>1</sup>, Mohamed Biniz<sup>2</sup>

<sup>1</sup>Department of Computer Science, Faculty of Science and Technology, Sultan Moulay Slimane University, Beni-Mellal, Morocco

<sup>2</sup>Department of Computer Science, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco

## Article Info

### Article history:

Received May 14, 2021

Revised Dec 25, 2021

Accepted Jan 13, 2022

### Keywords:

Correction center

Natural language processing

Optical character recognition

Recursive technique

Sentences correction

Tifinagh

## ABSTRACT

Optical character recognition (OCR) is a recognition system used to recognize the substance of a checked picture. This system gives erroneous results, which necessitates a post-treatment, for the sentence correction. In this paper, we proposed a new method for syntactic and semantic correction of sentences it is based on the frequency of two correct words in the sentence and a recursive technique. This approach starts with the frequency calculation of each two words successive in the corpora, the words that have the greatest frequency build a correction center. We found 98% using our approach when we used the noisy channel. Further, we obtained 96% using the same corpus in the same conditions.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Otman Maarouf

Department of Computer Science, Sultan Moulay Slimane University

Av Med V, BP 591, Beni-Mellal 23000, Morocco

Email: maarouf.otman94@gmail.com

## 1. INTRODUCTION

With the advancement in innovation and handling speed, an ever-increasing number of complicated calculations for optical character recognition (OCR) frameworks including AI and neural organizations are proposed. [1] OCR is a course of changing over a picture portrayal into editable and text design. It is the technique for digitizing printed and written by hand text [2]. Numerous applications including number plate acknowledgment, book checking, and continuous transformation of transcribed text advantage from OCR. [3] Unfortunately, the results given by OCR are not always satisfying; it contains errors influencing the meaning of sentences. These errors are divided into several types: i) Missing characters: the result of the OCR contains several characters less than the number of characters in the image to be recognized. ii) Addition of characters: the result of the OCR contains several characters greater than the number of characters appearing in the image to be recognized. iii) Character modification: the result of the OCR contains several characters equal to the number of characters in the image to be recognized, but there are some that are modified by other characters different from the origin.

To solve this problem, post-processing must be added. At this level, we propose to use a new approach that applies in two stages; it begins with the detection of errors, then it attacks the syntactic and semantic correction of the OCR output, this approach is based on the frequency of two correct words in the sentence and a recursive technique. This approach starts with the frequency calculation of each two words successive in the corpora, the words that have the greatest frequency build a correction center, then it begins to correct all words using the recursive technique that will describe in the next section. This approach belongs to the natural language processing (NLP) domain, Which is a branch of artificial intelligence that analyzes,

understands, and generates natural languages used by humans to interact with computers in written contexts and spoken using natural human languages rather than computer languages [4].

In the literature, we find that the NLP domain is used in several languages, namely: English [5] French [6] Arabic [7]. On the other hand, the Amazigh language, which uses the Tifinagh characters, has not benefited the advantages offered by this domain. This lack motivated us to approach this area to improve OCR results for Tifinagh characters.

Tifinagh [8] is the set of alphabets used by the Amazigh population. The Royal Institute of Amazigh Culture (IRCAM) has normalized the Tifinagh alphabet of thirty-three characters as shown in Figure 1.



Figure 1. Tifinagh characters (IRCAM)

The remainder of the paper is coordinated as follows: section 2 portrays the engineering of the OCR framework utilized for the acknowledgment of composed archives in Tifinagh, section 3 examines the proposed approach of NLP took on to further develop the outcomes given by an OCR, section 4 shows the test results acquired to pass judgment on the exhibition of the proposed approach, at last, an end is given, to sum up, the reason for the work and to report the extricated ends.

## 2. OPTICAL CHARACTER RECOGNITION SYSTEM

Explaining research chronological, including research design, research procedure (in the form of algorithms, Pseudocode, or other), how to test, and data acquisition [3]–[5]. The description of the course of research should be supported references, so the explanation can be accepted scientifically [4]–[6].

The advancement in pattern recognition has accelerated recently due to the many emerging applications which are not only challenging, but also computationally more demanding, such as evident in OCR, [9] document classification, Computer vision [10], data mining, shape recognition, and biometric authentication, for example. The space of OCR is turning into an indispensable piece of archive scanners and is utilized in numerous applications like postal handling, script acknowledgment, banking, security (for example visa confirmation), and language distinguishing proof. The exploration in this space has been progressing for over 50 years and the results have been surprising with fruitful acknowledgment rates for printed characters surpassing close to 100%, with huge enhancements in execution for written by hand cursive person acknowledgment where acknowledgment rates have surpassed the 90% imprint. These days, numerous associations are relying upon OCR frameworks to dispense with human connections for better execution and proficiency [11].

Under the current work, a character recognition system is presented for recognizing Tifinagh characters extracted from pictures/designs implanted text records, for example, business cards pictures [12]. Figure 2 describes our OCR steps, it takes as the input image, it starts by the text extraction, then the binarization phase, Segmentation phase, and the last one is recognition finally it gives as output a text file.

### 2.1. Image acquisition

The proposed OCR begins with the image acquisition [3] process, it is the first step in OCR, which consists of getting a digital image and converting it into a proper form that can be easily handled by a computer. This may include quantization as well as compression of the image [13]. A special case of quantification is binarization, which involves only two image levels [14]. In most cases, the binary image is sufficient to characterize the image. The compression itself can be lossy or lossless. An overview of the different image compression techniques in [15]. This OCR takes any typewritten image containing Tifinagh characters in either “.png” or “.jpg” format.

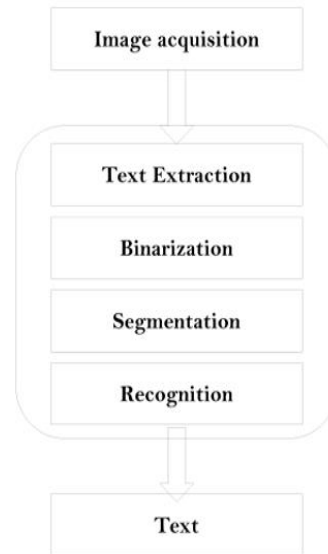


Figure 2. Block diagram of the present system

## 2.2. Text extraction

Through the filtering system, an advanced picture of the first record is caught. In OCR optical scanners are utilized, which for the most part comprise a vehicle instrument in addition to a detecting gadget that converts light power into dark levels. [16] Printed records generally comprise of dark print on a white foundation. Henceforth, when performing OCR, it isn't unexpected practice to change over the staggered picture into a bi-level picture of high contrast. Regularly this cycle, known as thresholding, is performed on the scanner to save memory space and computational exertion. Issues in thresholding can be seen in Figure 3.

The thresholding system is significant as the aftereffects of the accompanying acknowledgment are absolutely subject to the nature of the bi-level picture. In any case, the thresholding performed on the scanner is normally exceptionally straightforward. [17] A fixed limit is utilized, where dim levels underneath this edge are supposed to be dark and levels above are supposed to be white. For a high-balance archive with a uniform foundation, a prechosen fixed limit can be sufficient. In any case, a ton of archives experienced practically speaking have a somewhat huge reach interestingly. In these cases, more modern strategies for thresholding are needed to get a decent outcome [18].

The best techniques for thresholding are generally those, which can differ the limit over the record adjusting to the neighborhood properties as differentiation and splendor. In any case, such techniques ordinarily rely on staggered filtering of the record, which requires more memory and computational limit. In this manner, such procedures are rarely utilized regarding OCR frameworks, despite the fact that they bring about better pictures [11].

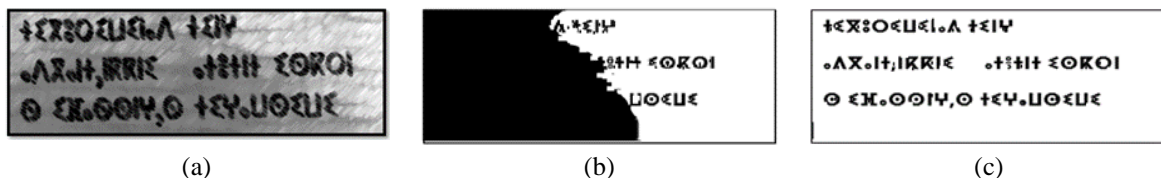


Figure 3. Issues in thresholding: (a) original dark level pictures, (b) image thresholded with worldwide strategy, and (c) image thresholded with a versatile technique

## 2.3. Binarization

A skew revised text locale is binarized utilizing a straightforward yet proficient binarization technique created by us before dividing it [19]. The calculation has been given below. Essentially, this is a further developed form of Bernsen's binarization strategy. In his technique, the number juggling method for the most extreme ( $G_{max}$ ) and the base ( $G_{min}$ ) dim levels around a pixel is taken as the limit for binarizing

the pixel. In the current calculation, the eight quick neighbors around the pixel subject to binarization are likewise taken as a central consideration for binarization. This sort of approach is particularly valuable to interface the detached forefront pixels of a person [12].

#### 2.4. Segmentation

Segmentation is the separation of characters or words. Most optical person acknowledgment calculations fragment the words into segregated characters, which are perceived separately [11]. Typically, this division is performed by segregating each associated part that is each associated dark region. This procedure is not difficult to carry out, however, issues happen if characters contact or then again in case characters are divided and comprise of a few sections. The primary issues in the division might be partitioned into four gatherings:

- Extraction of touching and fragmented characters.
- Distinguishing noise from the text.
- Mistaking graphics or geometry for text.
- Mistaking text for graphics.

For our case, the operation of the segmentation process is based on histogram-based thresholding. We will signify the histogram of pixel esteems by  $h_0, h_1, \dots, h_N$ , where  $h_K$  specifies the quantity of pixels in a picture with greyscale esteem  $k$  and  $N$  is the greatest pixel esteem (regularly 255). Ridler and Calvard (1978) and Trussell (1979) proposed a straightforward algorithm for picking a solitary edge. We will allude to it as the entomb implies algorithm.

At first, a supposition should be made at a potential incentive for the edge. From this, the mean upsides of pixels in the two classes created utilizing this limit are determined. The limit is repositioned to lie precisely somewhere between the two methods. Mean qualities are determined once more, and another limit is gotten until the edge quits evolving esteem [20].

#### 2.5. Recognition

Recognition is the last phase of the OCR system which is used to identify the segmented content. In this step, the correlation coefficients are used in the classification. The correlation coefficient is processed from the example information estimates the strength and bearing of a connection between two factors. A relationship coefficient is a number somewhere in the range of 0 and 1. In case there is no relationship, between the anticipated qualities and the real qualities, the connection coefficient is 0 or exceptionally low (the anticipated qualities are no more excellent than irregular numbers). As the strength of the connection between the anticipated qualities and real worth increments so does the relationship coefficient. An ideal fit gives a coefficient of 1.0. In this way, a superior outcome is compared to the higher correlation coefficient [21]. Corr2 computes the correlation coefficient using:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (1)$$

where

- A and B are two matrices
- m: number of rows
- n: number of columns

### 3. THE NOISY CHANNEL MODEL

Around here, we present the boisterous channel model and advise the most ideal way of applying it to the endeavor of recognizing and changing spelling botches. The boisterous channel model was applied to the spelling remedy task at about a similar time by research artificial AT and T Bell [22] and IBM Watson Research [23].

The nature of the loud channel model in Figure 4 is to treat the inaccurately spelled uproarious channel word like an adequately spelled word had been "ravaged" by being used as a clamorous correspondence procedure. This channel presents "uproar" as replacements or various changes to the letters, making it hard to see the "certified" word. The goal, by then, is to develop a model of the boisterous channel. Given this model, we then find the authentic word bypassing every declaration of the language through the model of the loud channel and see which one comes the closest to the mistakenly spelled word [24].

In the noisy channel model, we envision that the surfaces' structure we see is really a "misshaped" type of a unique word that went through a loud channel. The decoder goes every theory through a model of this channel and picks the word that best matches the surface boisterous word [25]

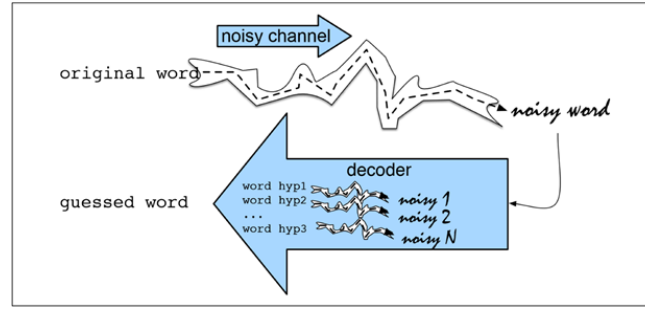


Figure 4. Loud channel model

### 3.1. Extraction of candidates

This noisy channel model is a sort of Bayesian inference. We see a perception  $x$  (an incorrectly spelled word) and the work is to find the word  $w$  that created this incorrectly spelled word. Out of all potential words in the jargon,  $V$  we need to find the word  $w$  to such an extent that  $P(w|x)$  is most elevated. We utilize the cap documentation  $\hat{\cdot}$  to signify "the gauge of the right word".

$$\hat{w} = \operatorname{argmax}_{w \in V} P(w|x) \quad (2)$$

The function  $\operatorname{argmax}_x f(x)$  means "the  $x$  such that  $f(x)$  is maximized". The (2) in this way implies that out of all words in the jargon, we need the specific word that augments the right-hand side  $P(w|x)$ .

### 3.2. Bayesian classification

The instinct of Bayesian classification is to utilize Bayes rule to change (2) into a bunch of different probabilities. Bayes rule is introduced in (3) it offers us a way of reprieving down any contingent likelihood  $P(a|b)$  into three different probabilities:

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)} \quad (3)$$

it can then substitute (3) into (2) to get (4):

$$w = \operatorname{argmax}_{w \in V} \frac{P(x|w)P(w)}{P(x)} \quad (4)$$

It can helpfully streamline (4) by dropping the denominator  $P(x)$ . For what reason is that? Since we are picking a potential adjustment word out everything being equal, we will process  $\frac{P(x|w)P(w)}{P(x)}$  for each word. In any case,  $P(x)$  does not change for each word; we are continually approaching the no doubt word for the equivalent watched mistake  $x$ , which must have the same probability  $P(x)$ . Along these lines, we can pick the word that augments this less difficult [24].

$$w = \operatorname{argmax}_{w \in V} P(x|w)P(w) \quad (5)$$

To summarize, the noisy channel model says that we have some evident basic word  $w$ , and we have an uproarious channel that modifies the word into some conceivable incorrectly spelled noticed surface structure. The probability or channel model of the uproarious channel model delivering a specific perception grouping  $x$  is demonstrated by  $P(x|w)$ . The earlier likelihood of a secret word is demonstrated by  $P(w)$ . We can process the most earlier likelihood plausible word  $\hat{w}$  considering that we have seen some noticed incorrect spelling  $x$  by increasing the prior  $P(w)$  and the likelihood  $P(x|w)$  and picking the word for which this item is most prominent.

The noisy channel approach way to deal with rectifying non-word spelling blunders by taking any word not in the spell word reference, creating a rundown of up-and-comer words, positioning them as per (5), and picking the most noteworthy positioned one. We can adjust (5) to allude to this rundown of competitor words rather than the full jargon  $V$  as by [24].

$$\hat{w} = \underset{w \in C}{\operatorname{argmax}} \overline{P(x|w)} \overline{P(w)} \quad (6)$$

#### 4. THE PROPOSED APPROACH

NLP is a way for computers to analyze, comprehend, and get importance from human language in a shrewd and helpful manner. By using NLP, designers can coordinate and construction information to perform assignments like programmed rundown, interpretation, named substance acknowledgment, relationship extraction, opinion investigation, discourse acknowledgment, and subject division.

##### 4.1. Correction of words

Correcting the wrong word requires a list of candidates to select among them. In this part, we will treat the word correction based on a dictionary of Tifinagh words, calculating the distance between words in order to have the list of candidates taking the words that we have a maximum distance. The distance between two words is the number of common letters between the two letters in the same position, which is being calculated by (7).

$$\begin{aligned} dist(w_1, w_2) = & |\{c_i: (c_i \in w_1^k \wedge c_i \in w_2^k) \vee (c_i \in w_1^k \wedge c_i \in w_2^{k \pm 1}) \\ & \vee (c_i \in w_1^{k \pm 1} \wedge c_i \in w_2^k) \vee (c_i \in w_1^{k \pm 1} \wedge c_i \in w_2^{k \pm 1}) \vee \}| \end{aligned} \quad (7)$$

with

- $c_i$ : The character of index  $i$  in the word 1.
- $w_1^k$ : The position  $k$  in the word 1.

in OCR We can find certain types of errors, erroneous words by deletion, insertion, transposition or substitution.

Table 1 shows us that the two types of erroneous words (by transposition and substitution), have the same number of the correct word characters, on the other hand, the words erased by suppression have a number of characters of less than the correct word character, and the words erased by the insertion have a number of characters greater than the number of correct word characters.

Erroneous words	Correct words	types of error
◦O◦O	◦ΛO◦O	deletion
◦ΛΛO◦O	◦ΛO◦O	insertion
◦OΛ◦O	◦ΛO◦O	transposition
◦Λ⊙◦O	◦ΛO◦O	substitution

To correct an erroneous word, we will calculate its distance with all the dictionary words that have the same size or a size  $\pm 1$ , after which we will return the max of the distance by (8).

$$\max\_dist(w) = \max \{ dist_i(w, w_i) \} \quad (8)$$

with:

- $w$  the corrected word
- $w_i$ : are dictionary words that have a size equal to the size or size  $\pm 1$  of  $w$

we can retrieve all the words that have a maximum distance with the wrong word, using the maximum of the distance indicated (7).

$$words\_corrects(w) = \{ w_i: dist(w, w_i) = \max\_dist(w) \} \quad (9)$$

##### 4.2. Words frequency

To correct a sentence, we need to start the correction with correct words. Therefore, we will determine the word position belongs to the dictionary, such that the word that follows also belongs to the dictionary, by (10).

$$posw_i = \{ i : w_i \in \text{dictionnaire} \wedge w_{i+1} \in \text{dictionnaire} \} \quad (10)$$

with:

- $w_i$  the word of the sentence has the position  $i$ .

For  $posw_i = \emptyset$ , we will recalculate  $posw_i$  by (11).

$$posw_i = \{i : w_i \in \text{dictionnaire}\} \quad (11)$$

We calculate the frequency, using (11) of the words  $w_i$  in the corpus knowing that  $w_{i+1}$  consecutive with  $w_i$  by (12).

$$freqw_i = \{n_{i/i+1}/N : i \in posw\} \quad (12)$$

where:

- $n_{i/i+1}$  is a number of occurrences of  $w_i$  in the corpus followed by  $w_{i+1}$
- $N$  is a total number of corpora words

if we used (6), we will calculate the frequency of the words by (13).

$$freqw_i = \{n_i/N : i \in posw\} \quad (13)$$

with

- $n_i$ : the number of word occurrences in the corpus
- $N$ : total number of corpora words

after the calculation of the frequencies of the words, we will seek the maximum frequency by (14).

$$\text{maxfreq} = \{\max(freqw)\} \quad (14)$$

Then we will return the word position that has a maximum frequency by (15). Therefore, the correction process focuses on this position (called: the correction center).

$$pos = \{i : \text{maxfreq} = freqw_i\} \quad (15)$$

#### 4.3. Correction of the sentence

The frequency of the words that we calculated in the previous section, allowed us to correct each word of a sentence, based on the words that exist in the dictionary. Now we will compute the frequency of words that are close to the erroneous word, using the result of (14) and (15), by a calculation that takes into consideration  $n$  correct words after the erroneous word or before the erroneous word by (16).

i) Left formula:

$$freqw_{i/pos}^L = \{\frac{n_{i/pos}}{N} : w_i \in \text{words\_corrects}(\bar{w})\} \quad (16)$$

where

- $n_{i/pos}$  the number of occurrences of the word  $w_i$  in the corpus knowing that it is by (17)  $w_{i+1}w_{pos}$ .
- $N$ : The total number of corpora words
- $\bar{w}$ : The word to correct

ii) Right formula:

$$freqw_{i/pos}^R = \{\frac{n_{i/pos}}{N} : w_i \in \text{words\_corrects}(\bar{w})\} \quad (17)$$

with:

- $n_{i/pos}$  The number of occurrences of the word  $w_i$  in the corpora knowing that preceded by  $w_0.w_{pos+1}$
- $N$ : The total number of corpora words
- $\bar{w}$ : The word to correct

For each (11) and (12), we calculate the maximum of frequencies on the left or the right as by (18) and (19).

$$\text{maxfreq}_{i/pos}^L = \{\max(freqw_{i/pos}^L)\} \quad (18)$$

and

$$\text{maxfreq}_{i/pos}^R = \{\max(freqw_{i/pos}^R)\} \quad (19)$$

If we have  $maxfreq_{i/pos}^R = \text{zero}$ , we will apply the recursive technique: The incrementing of pos (pos=pos+1) until  $maxfreq_{i/pos}^R > 0$ . Similarly, if we have  $maxfreq_{i/pos}^L = \text{zero}$  we will apply the recursive technique: The decrement of pos (pos=pos-1) until  $maxfreq_{i/pos}^L > 0$ . To correct an erroneous word, we will use two formulas depending on its position in relation to the correction center. If we have the position of the wrong word above the correction center:

$$correct^R(\bar{w}) = \{w_i: maxfreq_{i/pos}^R = freqw_{i/pos}^R \text{ and } w_i \in words\_corrects(\bar{w})\} \quad (20)$$

if we have the position of the wrong word below the correction center:

$$correct^L(\bar{w}) = \{w_i: maxfreq_{i/pos}^L = freqw_{i/pos}^L \text{ and } w_i \in words\_corrects(\bar{w})\} \quad (21)$$

with  $\bar{w}$  is the wrong word.

Finally, we can correct a sentence, using the word correction on the left or the right and the recursive technique, by (22).

$$correct(sentence) = \{correct^L(w_{i < pos}); correct^R(w_{i > pos})\} \quad (22)$$

where

- $w_{i < pos}$ : The sentence words locate before the word exists in the correction center
- $w_{i > pos}$ : The sentence words locate after the word exists in the correction center

## 5. TEST AND MEASURE

The evaluation of our system requires experiments, for that, we divided the dataset, containing 5000 images, into two parts: 80% for training, and 20% to test the performance of the system.

### 5.1. Experimental results of OCR

The realized OCR system starts by reading an image containing a text written in Tifinagh, then performs the recognition of the content, finally generates a file containing the result of the recognition. Figure 5 is an example of the image, containing a sentence written in Tifinagh, used to test our OCR. The OCR output obtained is illustrated in Figure 6. The observation of this result shows that there are two errors, these errors are reported in Table 2. In the first word, the error exists in the third character. On the other hand, in the second word, the error appears in the first character.



Figure 5. Example of image presented at the OCR input

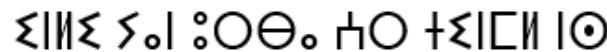


Figure 6. Example of image retrieved at the OCR output

Table 2. Errors extracted

Input	Output
ⵍⵉⵎⵉⵔ	ⵍⵉⵎⵉⵔ
ⵉⵎⵉⵔ	ⵉⵎⵉⵔ

### 5.2. Experimental results of noisy channel

Find words that have a comparative spelling to the info word. Investigation of spelling mistake information has shown that most spelling blunders comprise of a solitary letter change thus we regularly make the working on the supposition that these competitors have an alter distance of 1 from the blunder word. To find this rundown of applicants we'll utilize the base alter distance calculation, yet stretched out so that notwithstanding inclusions, erasures, and replacements, we'll add a fourth sort of alter, interpretations, in



which two letters are traded. The form of alter distance with interpretation is called Damerau-Levenshtein update distance. Applying all such single changes to “ $\xi l \mu \xi$ ” yields the rundown of up-and-comer words as shown in Table 3.

Once have a bunch of up-and-comers, to score every one utilizing (6) requires that process the earlier and the channel model. The earlier likelihood of every amendment  $P(w)$  is the language model likelihood of the word  $w$  in setting, which can be processed utilizing any language model, from unigram to trigram or 4-gram. For this model, let us start in the accompanying Table 4 by accepting a unigram language model.

Table 3. Candidate corrections for the misspelling “ $\xi l \mu \xi$ ” and the transformations that would have produced the error (after [22] “—” represents a null letter)

Error	Correction	Types
$\xi l \mu \xi$	$\xi l \mu \xi l$	deletion
$\xi l \mu \xi$	$\xi l \mu \xi$	substitution
$\xi l \mu \xi$	$\xi l \mu$	insertion

Table 4. The probability to correct the misspelling “ $\xi l \mu \xi$ ”

w	Count(w)	P(w)
$\xi l \mu \xi l$	2 798	0.000256
$\xi l \mu \xi$	19 300	0.001565
$l \xi \mu \xi$	230	0.0000695

How might appraise the probability  $P(x|w)$ , likewise called the channel model or blunder model? An ideal model of the likelihood that a word will be mistyped would blunder model condition on a wide range of components: who the typist was, regardless of whether the typist was left given or right-gave. Fortunately, we can get a sensible gauge of  $P(x|w)$  just by checking out nearby setting: the character of the right letter itself, the incorrect spelling, and the encompassing letters.

Once have the disarray lattices, we can appraise  $P(x|w)$  as follows (where  $w_i$  is the  $i$ th character of the correct word  $w$ ) and  $x_i$  is the  $i$ th character of the typo  $x$ . Using the counts from [17] results in the error model probabilities for acres shown in Table 5.

$$P(x|w) = \begin{cases} \frac{del[xi-1,wi]}{count[xi-1wi]}, & \text{if deletion} \\ \frac{ins[xi-1,wi]}{count[wi-1]}, & \text{if insertion} \\ \frac{sub[xi,wi]}{count[wi]}, & \text{if substitution} \end{cases} \quad (23)$$

Table 5. Channel model for  $\xi l \mu \xi$

Candidate correction	Correct letter	Error letter	x/w	P(x/w)
$\xi l \mu \xi l$	-	l	l/#	0.000045
$\xi l \mu \xi$	$\mu$	$\mu$	$\mu/\mu$	0.00064
$l \xi \mu \xi$	l	$\xi$	$\xi/l$	0.0000012

Table 6 shows the final results for each of the corrections; the unigram prior is calculated with (23) and the confusion matrices. The computations in Table 6. show that the noisy channel model chooses  $\xi l \mu \xi$  as the better, and  $\xi l \mu \xi l$  as the second most likely word. For this reason, it is important to use larger language models than unigrams. For example, if we use a corpus o compute bigram probability for the words  $\xi l \mu \xi$  and  $\xi l \mu \xi l$  in their context using add-one smoothing, we get the following probabilities:

$$P(\mathcal{S}_o | \xi l \mu \xi) = .000051$$

$$P(\mathcal{S}_o | \xi l \mu \xi l) = .000002$$

Combining the language model with the error model in Table 6, the bigram noisy channel model now chooses the correct word  $\xi l \mu \xi$ .

Table 6. Calculated the ranking for each word correction

Candidate correction	Correct letter	Error letter	x/w	P(x/w)	P(w)
$\xi l \mu \xi l$	-	l	l/#	0.000045	0.000256
$\xi l \mu \xi$	$\mu$	$\mu$	$\mu/\mu$	0.00064	0.001565
$l \xi \mu \xi$	$l \xi$	$\xi l$	$\xi l/l \xi$	0.0000012	0.0000695

### 5.3. Experimental results of proposed approach

Errors correction will be realized using the proposed approach (NLP). Using (14), we found the following results for the correction of “ $\xi l \mu \xi$ ”: The candidate words are  $\xi l \odot \xi$ ,  $\xi l \mu \xi$ ,  $\xi l \mu \xi$  and  $\xi l \times \xi$ . For the correction of “ $\mu \odot$ ”, we have three candidate words:  $\times \odot$ ,  $\mu \odot$  and  $\square \odot$ . Now we will look for the position words most frequently used in Figure 7.

Table 7. Frequency of two successive correct words

Words	Frequency
$\mu \odot l \quad \odot \odot \odot$	6.895301738404073E-4
$\mu \xi l \mu \odot \quad \odot$	9.85043105486296E-5

According to the results in Table 7, it can be seen that “ $\mu \odot l \quad \odot \odot \odot$ ” is the most common combination, so the position returned by (15) is “1”. This is the correction center and the position of the word “ $\mu \odot l$ ”. Using the correction on the left, we will correct the words that are before the correction center (20). The word “ $\xi l \mu \xi$ ” does not exist in the dictionary, we will calculate the frequency of all the words that are close to it using (20) with pos=1 (i.e., frequency of words closes to “ $\xi l \mu \xi$ ” followed by “ $\mu \odot l \quad \odot \odot \odot$ ”) as shown in Table 8.

Table 8. Frequency of each word among the candidate words of “ $\xi l \mu \xi$ ”

The candidate word	The frequency
$\xi l \odot \xi$	0.0
$\xi l \mu \xi$	6.855846505486296E-9
$\xi l \mu \xi$	8.855.254505486296E-7

According to the results indicate in Table 8, we conclude that the correct word is “ $\xi l \mu \xi$ ”, instead of putting “ $\xi l \mu \xi$ ”, we will replace it with “ $\xi l \mu \xi$ ”. Now we will go to the right correction, we have the word «  $\mu \odot$  » does not exist in the dictionary, we will compute the frequency of all the words that are close to it using (21) with pos=1 (i.e., the frequency of words closes to “ $\mu \odot$ ” preceded by “ $\xi l \mu \xi \quad \mu \odot l \quad \odot \odot \odot$ ”) can be seen in Table 9.

Table 9. Frequency of each word among the candidate words of “ $\mu \odot$ ”

Word	Frequency
$\mu \odot$	8.31651623321656E-8
$\times \odot$	0.0
$\square \odot$	0.0




From the results, indicate in Table 9, we can conclude that the correct word is “ $\mu \odot$ ”, instead of putting “ $\mu \odot$ ”, we will replace it with “ $\mu \odot$ ”. We have the word “ $\mu \xi l \mu$ ” exists in the dictionary its frequency knowing that it is preceded by “ $\xi l \mu \xi \quad \mu \odot l \quad \odot \odot \odot \quad \mu \odot$ ” is 1.3558880795743596E-6 not null, so the word “ $\mu \xi l \mu$ ” is correct. Similarly, we have the word “ $\odot$ ” exists in the dictionary its frequency knowing that it is preceded by “ $\xi l \mu \xi \quad \mu \odot l \quad \odot \odot \odot \quad \mu \odot \quad \mu \xi l \mu$ ” is 1.3558880795743596E-6 not null, so the word “ $\odot$ ” is correct. Similarly, we have the word “ $\odot$ ” exists in the dictionary its frequency knowing that it is preceded by “ $\xi l \mu \xi$






- [11] H. Modi and M. C., 'A review on optical character recognition techniques', *International Journal of Computer Applications*, vol. 160, no. 6, pp. 20–24, Feb. 2017, doi: 10.5120/ijca2017913061.
- [12] A. F. Mollah, N. Majumder, S. Basu, and M. Nasipuri, 'Design of an optical character recognition system for camera- based handheld devices', *IJCSI International Journal of Computer Science*, vol. 8, no. 4, pp. 283–289, 2011.
- [13] J. Lázaro, J. L. Martín, J. Arias, A. Astarloa, and C. Cuadrado, 'Neuro semantic thresholding using OCR software for high precision OCR applications', *Image and Vision Computing*, vol. 28, no. 4, pp. 571–578, 2010.
- [14] N. Islam, Z. Islam, and N. Noor, 'A Survey on Optical Character Recognition System', *arXiv:1710.05703 [cs]*, Oct. 2017, Accessed: Feb. 07, 2022. [Online]. Available: <http://arxiv.org/abs/1710.05703>
- [15] W. B. Lund, D. J. Kennard, and E. K. Ringger, 'Combining multiple thresholding binarization values to improve OCR output', in *Document Recognition and Retrieval XX*, 2013, vol. 8658, p. 86580R.
- [16] K. M. Yindumathi, S. S. Chaudhari, and R. Aparna, 'Analysis of image classification for text extraction from bills and invoices', in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2020, pp. 1–6. doi: 10.1109/ICCCNT49239.2020.9225564.
- [17] R. Deepa and K. N. Lalwani, 'Image classification and text extraction using machine learning', in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Jun. 2019, pp. 680–684. doi: 10.1109/ICECA.2019.8821936.
- [18] D. Saravanan and D. Joseph, 'Image data extraction using image similarities', 2019, pp. 409–420. doi: 10.1007/978-981-13-1906-8\_43.
- [19] A. F. Mollah, S. Basu, N. Das, R. Sarkar, M. N. Nasipuri, and M. Kundu, 'Binarizing business card images for mobile devices', *Int. Conf. On advances in computer vision and it*, pp. 968–975, 2009.
- [20] M.-H. Yang and N. Ahuja, 'Recognizing hand gesture using motion trajectories', 2001, pp. 53–81. doi: 10.1007/978-1-4615-1423-7\_3.
- [21] S. Sahoo, N. Dash, and P. Sahoo, 'Word extraction from speech recognition using correlation coefficients', *International Journal of Computer Applications*, vol. 51, no. 13, pp. 21–25, Aug. 2012, doi: 10.5120/8102-1694.
- [22] M. D. Kemighan, K. Church, and W. A. Gale, 'A spelling correction program based on a noisy channel model', in *in the 13th International Conference on Computational Linguistics*, 1990, pp. 205–210.
- [23] E. Mays, F. J. Damerau, and R. L. Mercer, 'Context based spelling correction', *Information Processing & Management*, vol. 27, no. 5, pp. 517–522, Jan. 1991, doi: 10.1016/0306-4573(91)90066-U.
- [24] D. Jurafsky and J. H. Martin, *Speech and language processing (3rd ed. draft)*. 2009.
- [25] H. Lane, C. Howard, and H. M. Hapke, *Natural language processing in action: understanding, analyzing, and generating text with python. shelter island*. Simon and Schuster, 2019.

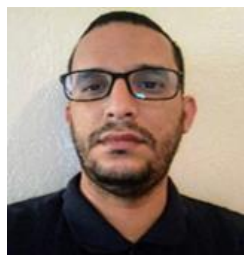
## BIOGRAPHIES OF AUTHORS






**Otman Maarouf**    received his master's degree in business intelligence in 2018 from the Faculty of Science and Technology, University Sultan Moulay Slimane Beni-Mellal. He is currently a PhD degree student. His research activities are located in the area of natural language processing specifically; it deals with Part of speech, named entity recognition, and limatization-steaming of the Amazigh language. He can be contacted at email: [maarouf.otman94@gmail.com](mailto:maarouf.otman94@gmail.com).



**Rachid El Ayachi**    obtained a degree in Master of Informatic Telecom and Multimedia (ITM) in 2006 from the Faculty of Sciences, Mohammed V University (Morocco) and a Ph.D. degree in computer science from the Faculty of Science and Technology, Sultan Moulay Slimane University (Morocco). He is currently a member of laboratory TIAD and a professor at the Faculty of Science and Technology, Sultan Moulay Slimane University, Morocco. His research focuses on image processing, pattern recognition, machine learning, semantic web. He can be contacted at email: [rachid.elayachi@usms.ma](mailto:rachid.elayachi@usms.ma).



**Mohamed Biniz**    received his master's degree in business intelligence in 2014 and Ph.D degree in computer science in 2018 from the Faculty of Science and Technology, University Sultan Moulay Slimane Beni-Mellal. He is a professor at polydisciplinary faculty University Sultan My Slimane Beni Mellal morocco. His research activities are located in the area of the semantic web engineering and deep learning specifically, it deals with the research question of the evolution of ontology, big data, natural language processing, dynamic programming. He can be contacted at email: [mohamedbiniz@gmail.com](mailto:mohamedbiniz@gmail.com).