# Realization of an intelligent evaluation system

**Otman Maarouf[1], Rachid El Ayachi[1], Mohamed Biniz[2]**
[1]Department of Computer Science, Faculty of Science and Technology, Sultan Moulay Slimane University, Beni Mellal, Morocco
[2]Department of Computer Science, Faculty of Polydisciplinary, Sultan Moulay Slimane University, Beni Mellal, Morocco

**Article Info**

***Article history:***

Received May 17, 2021
Revised Dec 24, 2021
Accepted Mar 9, 2023

***Keywords:***

Intelligent evaluation system
Open questions
Programming questions
Semantic similarity
Syntactical similarity

**ABSTRACT**

A number of benefits have been reported for computer-based assessments over traditional paper-based exams, both in terms of IT support for question development, reduced distribution and test administration costs, and automated support. Possible for the ranking. However, existing computerized assessment systems do not provide all kinds of questions, namely open questions that require writing solutions. To overcome the challenges of the existing, the objective of this work is to achieve an intelligent evaluation system (IES) responding to the problems identified, and which adapts to the different types of questions, especially open-ended questions of which the answer requires sentence writing or programming.

***Corresponding Author:***

Otman Maarouf
Department of Computer Science, Faculty of Science and Technology, Sultan Moulay Slimane University
Beni Mellal, Morocco
Email: maarouf.otman94@gmail.com

## 1. INTRODUCTION

More and more scientific findings are becoming available as a result of the ongoing advancements in natural science, humanities, and science and technology. Modern educational ideas, concepts, and the utilization of contemporary educational technology are necessary in order to address the modernization, cosmopolitanization, and future-oriented nature of all subject matter instruction today [1]. The delivery of customized education would be ineffective without the constant observation and evaluation of the student, both for the purpose of an efficient educational assessment and to enable the system to adjust to the demands of the learners. It can be difficult to monitor and evaluate students' performance in a classroom setting, especially when it's required to be done in real time.

Classical assessments have encountered several problems, namely: printing, monitoring, space management and correction. This requires the automation of the evaluation process using computerized assessment systems. According to the research carried out, we are only the evaluation systems are limited in terms of questions asked. They are based on questions of the QCM type, true or false, filling of empty fields, and multiple answers. But we note the absence of open-type questions, writing answers, which are of great importance in the learner’s theoretical and practical assessments. Attempts have been made to overcome the problems encountered in the classical evaluation system software engineering institute (SEI). This system offers two other types of evaluation: i) use the system compiler to answer the questions and ii) the second is interested in the questions that require writing answers in the form of sentences; the correction of this kind of answer has adopted the notion of semantic similarity.

An intelligent evaluation system (IES) in the form of a website has been developed to enhance the automated correction of assessments and allow for the assessment of learners' learning levels. It provides a number of options, including the ability for students to take assessments online, prepare examinations with a variety of topics, including open-ended and programming-related questions, and automatically correct answers based on teachers' responses, then it gives a score of the student. Regarding paper's structure, there are two sections; the first one contains the different methods for calculating the semantic and syntaxique similarity between two sentences. The second section give the general architecture of the realized system, all tools used in this project are mentioned in this section, finally the conclusion will be given.

## 2. CONCEPT AND TYPES OF SIMILARITIES

### 2.1. Introduction

Measuring the similarity between two sentences (or short texts) consists in evaluating to what extent the meaning of these sentences is close [2]. This task semantic textual similarity (STS) is often used in several important areas of the automatic language processing (TAL), among which we can mention the search for information, the categorization of texts [3], the summary of text [4], and the machine translation [5]. When we talk about similarity we talk about classification, clustering (or clustering) to describe data partitioning and a cluster is then a set of data or elements with similarities. The description language of the objects of a database must make it possible to define the distance of this object from the others.

In the field of artificial intelligence, similarity is one of the criteria for computer analysis of clusters and for data partitioning. This automatic classification step is necessary for the implementation of the machine learning methods. Expert software also seeks to take into account the context, according to which the similarity may vary [6]. The software will do a lot more relevant work as the attributes of the data will be useful and relevant in the context.

### 2.2. Syntactic similarity

Measuring syntactic similarity, "the syntactic word denotes in the linguistic sense a method of classification of languages according to the order of appearance of words in the sentence. Between words, the text runs in the field of data mining plays an important role [7].

### 2.2.1. Term frequency-inverse document frequency method

Term frequency-inverse document frequency (TF-IDF) *est une approche de pondération reconnue par un poids, et qui est souvent utilisée dans la recherche d'information et l'exploration de texte* [8]. *Ce poids est une mesure statistique numérique destinée à refléter l'importance d'un mot pour un document dans une collection ou un corpus* [9]. Typically, the weight of TF-IDF is composed of two terms: the first calculates the normalized TF, which is the number of times a word appears in a document, divided by the total number of words in that document [10]. The second term is the IDF, calculated as the logarithm of the number of documents in the corpus divided by the number of documents containing the specific term. TF is defined as [11].

– Term frequency

We notice: $Tf\ (w,d) = \{w' \in d: w' = w\}$ where w is a word, and $d = \{w_1, \dots, w_m\}$ is a document such as:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

where;

$n_{i,j}$ : the number of appearances of the word we want to calculate.

$n_{k,j}$ : the sum of all the words existing in the document by eliminating the punctuation, the spaces, the apostrophes.

– Inverse document frequency

$$idf_i = log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{2}$$

with,

$|D|$ : total number of documents in the corpus

$|\{d_j : t_i \in d_j\}|$ : number of documents where the term appears (i.e. $n_{i,j} \neq 0$)

– Calculation of TF-IDF

Finally, to put it all together, the total weight of TF-IDF for a token in a document is the product of its TF and IDF weights:

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{3}$$

#### 2.2.2. Similarity cosine

Cosine similarity is frequently used as a measure of similarity between two documents [12]. It may be a question of comparing the texts resulting from a corpus in an optic of classification or search of information (in this case, a vectorized document is constituted by the words of the request and is compared by measure of the cosine of the angle with vectors corresponding to all the documents present in the corpus, so we evaluate which ones are closest) [13]. As the angle measurement between two vectors can only be done with numerical values, we must imagine a way to convert the words of a document into numbers. This is why we rely on the results of the previous TD-IDF method which is the weight of each word in a document and we can consider it as a vector. The cosine similarity between two documents d1 and d2 is a measure of similarity. It is a question of calculating the cosine of the angle between the vector representations of the documents to compare [14].

Similarity is $sim_{cosinus}(d_1, d_2) \in [0,1]$

$$\mathrm{sim_{cosinus}}(\mathrm{d_1}, \mathrm{d_2}) = \frac{\overrightarrow{\mathrm{d1}}.\overrightarrow{\mathrm{d2}}}{\|\overrightarrow{\mathrm{d1}}\|\|\overrightarrow{\mathrm{d2}}\|} \tag{4}$$

### 2.3. Semantic similarity

Semantic similarity is a concept in which a set of documents or terms are given a metric based on the similarity of their meaning/semantic content [15]. Concretely, this can be done by defining a topological similarity, for example, by using ontologies to define a distance between words, or by defining a statistical similarity, for example by using a vector space model to correlate terms and conditions [16]. Contexts from an appropriate body of text (co-occurrence) [17]. Text similarity is a field of research whereby two terms or expressions are assigned a score based on the likeness of their meaning. Kocoń and Maziarz [18] short text similarity measures have an important role in many applications such as word sense disambiguation, synonymy detection, spell checking, thesauri generation, machine translation, information retrieval, and question answering [19].

#### 2.3.1. Measure of Wu & Palmer [20]

In a domain of concepts, similarity is defined with respect to the distance between two concepts in the hierarchy and their position relative to the root. This similarity also takes into account the length of the original path $ci$ and the extremity $c_j$ but also the depth of their most specific common subsuming, i.e. the length of the original path and $c_0$ and the extremity [21]. The similarity between C1 and C2 is:

$$\mathrm{ConSim(C1, C2)} = \frac{2*\mathrm{N3}}{\mathrm{N1+N2+2*N3}} \tag{5}$$

with, N1 is the distance between the concept C1 and the concept C3; N2 is the distance between the C2 concept and the C3 concept; N3 is the distance between the concept C3 and the root.

This measure has the advantage of being simple to implement and of having good performance than the other similarity measures [22]. The Figure 1 describe the relationships between the conceptual C1, C2, C3, and root.

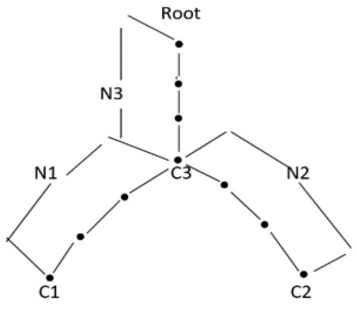


Figure 1. Conceptual relationships [20]

#### 2.3.2. Similarity of Mihalcia [23]

Simple lexical correspondence is described in [23]. The word-for-word similarity measures and a word specificity measure are used to estimate the semantic similarity of the sentence pairs [24]. The following notation function was used [25]:

$$\mathrm{Sim}(\mathrm{T}_1,\mathrm{T}_2)=\frac{1}{2}\left(\frac{\sum_{\mathrm{w}\in\{\mathrm{T}_1\}}(\mathrm{maxSim}(\mathrm{w},\mathrm{T}_2)*\mathrm{idf}(\mathrm{w}))}{\sum_{\mathrm{w}\in\{\mathrm{T}_1\}}\mathrm{idf}(\mathrm{w})}+\frac{\sum_{\mathrm{w}\in\{\mathrm{T}_2\}}(\mathrm{maxSim}(\mathrm{w},\mathrm{T}_1)*\mathrm{idf}(\mathrm{w}))}{\sum_{\mathrm{w}\in\{\mathrm{T}_2\}}\mathrm{idf}(\mathrm{w})}\right) \qquad (6)$$

where $maxSim\ (w,T)$ is the maximum score between the word w and the words in T according to a word-for-word similarity measure, and $idf\ (w)$ is the inverse document frequency of the word. A threshold of 0.5 was used for classification: a score above the threshold was classified as paraphrasing other than paraphrase.

According to Mihalcea *et al.* [23], takes into account the syntactic nature of terms and restricts comparisons of similarities to terms of the same syntactic nature: verbs, nouns, and adjectives between them. He tested several types of restrictions more or less binding related to the syntactic nature of the terms: from nouns/ nouns, adjectives/adjectives, and verbs/verbs to only proper nouns/proper nouns. In spite of what could be predicted by the similarity tests between terms according to their syntactic natures, these different tests all led to a very marked deterioration of the results. One might think, for example, that restricting a named entity to being comparable only to another named entity cannot damage the results, but experience has shown that this discrimination leads to a bad similarity between expressions such as "the Japanese president… 'And' in Japan, the president…" [26]. The system whose results are given in the evaluations thus operates without any restriction as to the syntactic nature of the terms compared [27]. Below is an example of calculating the similarity between two sentences. Table 1 shows the similarity matrix of Wu and Palmer [20]. $P_1$ et $P_2$ two sentences such as: i) $P_1$: eventually, a huge cyclone hit the entrance of my house and ii) $P_2$: finally, a massive hurricane attacked my home.

Table 1. Similarity matrix of Wu and Palmer [20] between two sentences

| | Cyclone/NN | Hit/VBD | Entrance/NN | House/NN |
|---|---|---|---|---|
| Hurricane/NN | 0.9565 | - | 0.2857 | 0.3158 |
| Attacked/VBD | - | 0.8571 | - | - |
| Home/NN | 0.3529 | - | 0.6667 | 1.0000 |

$$Sim(P_1,P_2)=\frac{1}{2}\Bigg(\frac{\left[0.9565*\log\left(\frac{2}{1}\right)+0.857*\log\left(\frac{2}{1}\right)+1*\log\left(\frac{2}{1}\right)\right]}{3*\log\left(\frac{2}{1}\right)}+$$

$$\frac{\left[0.9565*\log\left(\frac{2}{1}\right)+0.857*\log\left(\frac{2}{1}\right)+0.6667*\log\left(\frac{2}{1}\right)+1*\log\left(\frac{2}{1}\right)\right]}{4*\log\left(\frac{2}{1}\right)}=\frac{(0.93+0.865)}{2}=0.89$$

From these results, we find that the two sentences $P_1$ and $P_2$ are similar.

## 3. DESCRIPTION OF THE REALIZED EVALUATION SYSTEM

### 3.1. Architecture of IES

The IES developed is a website for assessing learner's level of learning. It offers several opportunities, namely: the online passage of assessment tests by students, the preparation of exams that contain all kinds of questions including programming questions and open questions and automatic correction based on the answers provided by teachers. IES is an evaluation system developed to automate the process of assessing learners' competencies, with several parties communicating with each other. Figure 2 shows the architecture of IES. The IES system has a set of components that are:

- Teacher space: this space allows teachers to register and authenticate in the platform to build an exam (questions, answers, and notation) and to enter the codes of students who are allowed to take the exam.
- Database: the database contains the information of the professors and students, as well as the exams (questions and answers proposed by the professors).
- Student area: this space allows students to register and to authenticate in the platform to pass exams.

### 3.2. Mechanism for correcting the open questions

To correct the open questions (the questions that have writing responses), the IES system makes use of the notion of semantic similarity; this similarity is calculated between the answer provided by the learner and the answer given by the teacher. The operation of the correction is done in several steps: i) the syntactic correctness of the answers, ii) sentence segmentation, labeling of speech parts, extraction of named entities using OPENNLP, iii) the elimination of stop words and the punctuation, and iv) the calculation of semantic similarity using the Mihalcea *et al.* [21] approach.

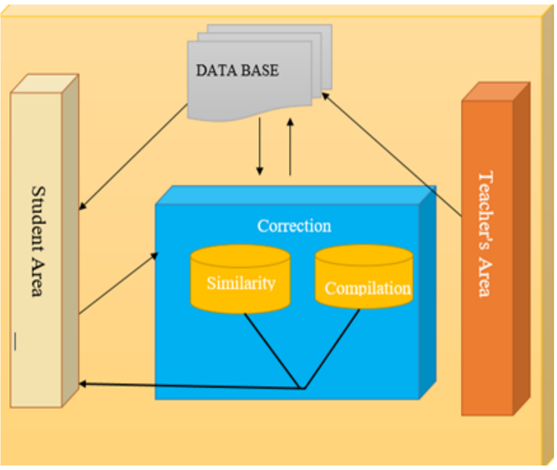


Figure 2. IES architecture developed

### 3.2.1. Syntax correction

The syntax correction of the sentence is a technique that allows to correct the syntax errors based on a corpus and a dictionary to correct a sentence syntactically, we need to determine the correction center of the sentence (the position of the word in the sentence that has a maximum frequency in the corpus) using (7) [28]:

$$pos = \{i : maxfreqw = \ freqw_i\} \tag{7}$$

with $maxfreqw$ is the maximum frequency of the phrase words.

The second step is the detection of the erroneous words of the sentence by doing a search and a comparison of the words of the sentence with the words of the dictionary, then the calculation of the distance between the erroneous words and the words of the dictionary, in order to recover all the words close to the erroneous word, by (8) and (9) [28]:

$$dist(w_1, w_2) = |\{\ c_i: (c_i \epsilon w_1^k \wedge c_i \epsilon w_2^k) \vee (c_i \epsilon w_1^k \wedge c_i \epsilon w_2^{k\pm1}) \\ \vee (c_i \epsilon w_1^{k\pm1} \wedge c_i \epsilon w_2^k) \vee (c_i \epsilon w_1^{k\pm1} \wedge c_i \epsilon w_2^{k\pm1}) \vee\}| \tag{8}$$

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

In (8) is adopted to calculate the distance between the erroneous word and all the words in the dictionary. In (9) is used to retrieve all words close to the erroneous word. With $max\_dist$ is the maximum distance between the wrong word and the dictionary words.

Afterwards, the correction of the wrong word is based on the correct words of the sentence, the correction center and the list of words close to the wrong word. This processing applies a recursive technique [12] and the n-gram correction to the left and right to choose the correct word among the words of the list by calculating the frequency of each word in the list followed or preceded by a n correct word of the phrase [29]. Then the correct word is that has a non-zero frequency with a maximum n.

$$correct^R(\overline{w}) = \{w_i: maxfreq^R_{i/pos} = \ freqw^R_{i/pos}\ et\ w_i \epsilon\ words\_corrects(\overline{w})\} \tag{10}$$

$$correct^L(\overline{w}) = \{w_i: maxfreq^L_{i/pos} = \ freqw^L_{i/pos}\ et\ w_i \epsilon\ words\_corrects(\overline{w})\} \tag{11}$$

If the position of the erroneous word exists after the correction center, then (10) is used. Otherwise, (11) is adopted. With $\overline{w}$ represents the wrong word. Finally, we can correct the sentence, using the correct word left or right and the recursive technique, by (12):

$$correct(phrase) = \{correct^L(w_{i<pos});\ correct^R(w_{i>pos})\} \tag{12}$$

with $w_{i<pos}$ represent the words of sentence located before the word existing in the correction center. $w_{i>pos}$ represent sentence words located after the word existing in the correction center.

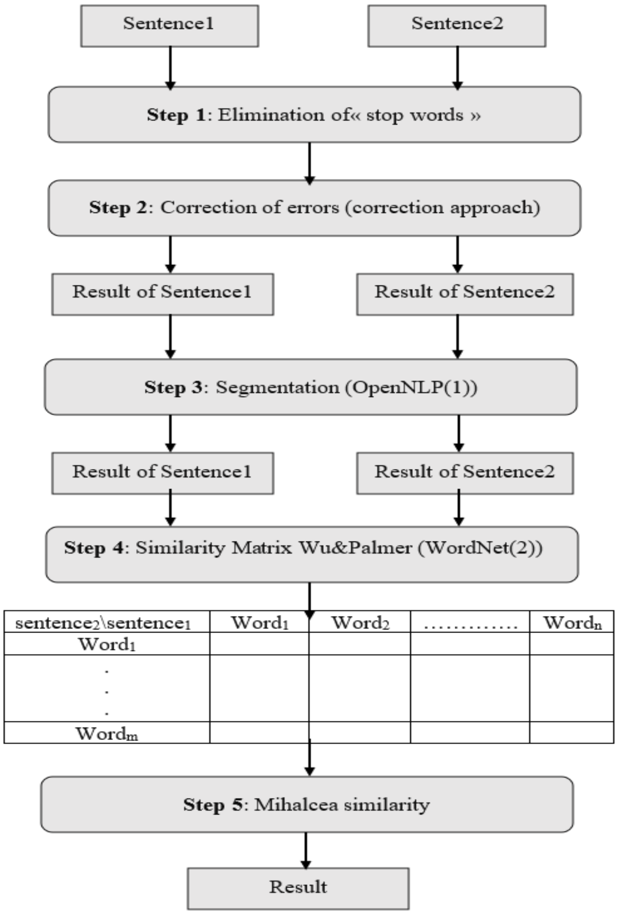
### 3.2.2. Segmentation of sentences and labeling of parts of speech

In this step, we used natural language processing (openNLP) to segment the sentence (answer of the question) and to make a labeling of each word of the sentence; this is adopted to calculate the semantic similarity between the answer proposed by the teacher and the other given by the student, using the approach of Mihalcea *et al.* [23]. We generate the similarity matrix of Wu & Palmer words of sentences that have the same grammatical field to use (6). If we find a similarity greater than or equal to 0.75, we consider that the similar sentences, that is to say the answer given by the student is correct, so he takes the complete note of the question. If no, we calculate the question score (13) by multiplying the similarity of the two answers and the scale of the question.

$$note(q) = Sim(R, R_p) * bar \tag{13}$$

Where R is the answer of the student; $R_p$ is the answer proposed by the professor; $bar$ is the full note of the question.

Figure 3 shows the different steps of the similarity calculation, starting with sentence cleaning, segmentation, generation of the similarity matrix and ending with the Mihalcea similarity calculation.


Sentence1
Sentence2
Step 1: Elimination of« stop words »
Step 2: Correction of errors (correction approach)
Result of Sentence1
Result of Sentence2
Step 3: Segmentation (OpenNLP(1))
Result of Sentence1
Result of Sentence2
Step 4: Similarity Matrix Wu&Palmer (WordNet(2))
Step 5: Mihalcea similarity
Result

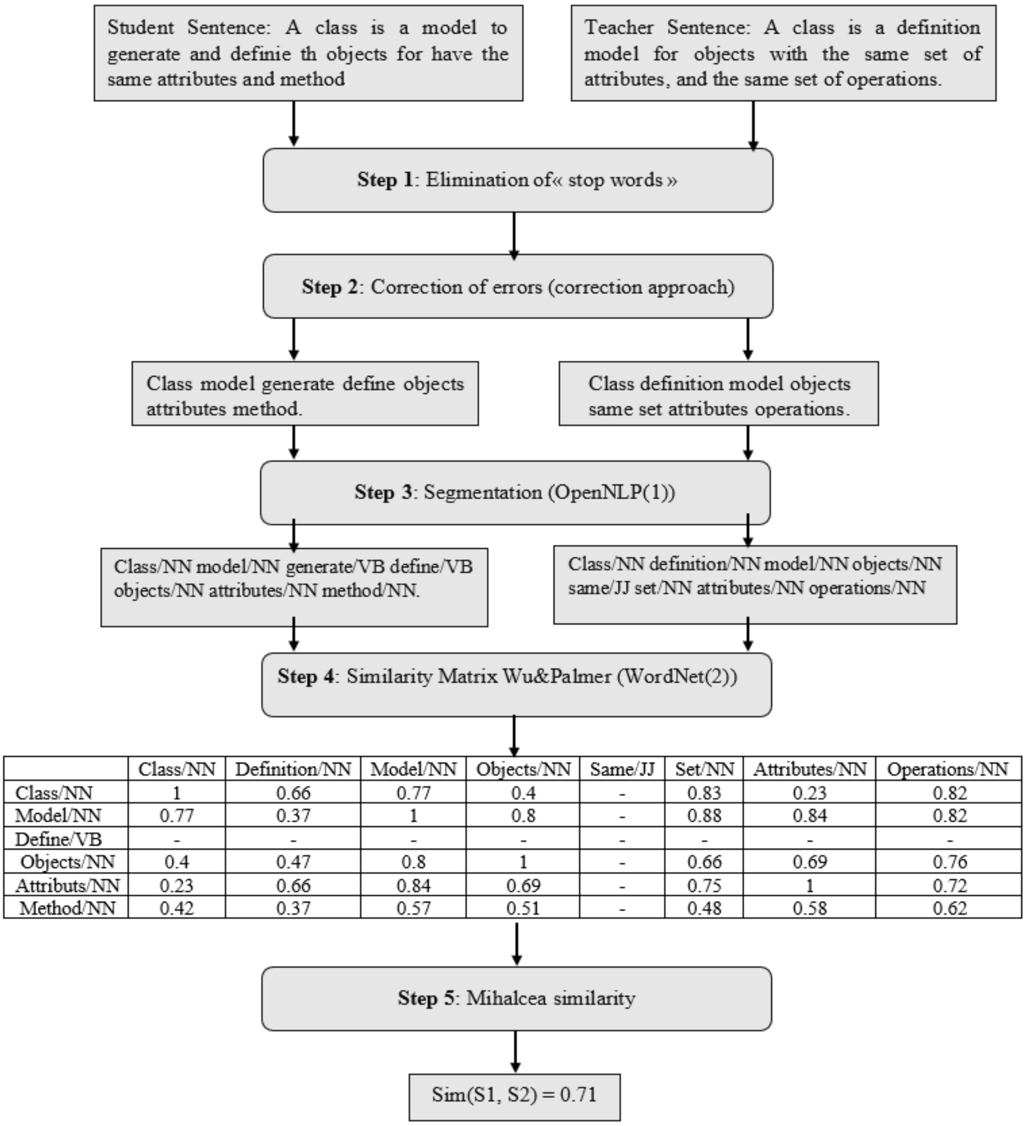
| $sentence_2 \backslash sentence_1$ | $Word_1$ | $Word_2$ | ………… | $Word_n$ |
|---|---|---|---|---|
| $Word_1$ | | | | |
| .<br>.<br>. | | | | |
| $Word_m$ | | | | |

Figure 3. Flowchart that presents the semantic similarity calculation process

- Apache openNLP: the apache OpenNLP library is a toolbox. Support of the openNLP library: i) tokenization, ii) segmentation of the sentence, iii) marking of part of speech, iv) named entity extraction, v) data chunking, vi) data analysis.
- WordNet: is a wide-ranging lexical database, developed for over 20 years by linguists from the cognitive science laboratory at Princeton University for the English language. It is freely usable, even been created manually or automatically from, in extension to, or in addition to WordNet. Programs from the world of artificial intelligence have also established bridges with WordNet. It is a semantic network of the English language, which is based on a psychological theory of language. The first version released dates back to June 1991 [13].

### 3.2.3. Application example

The statement of the question is as follows: “give the definition of java class”. The answer proposed by the professor is as follows: “a class is a definition model for objects with the same set of attributes, and the same set of operations”. The answer written by the student is: “a class is a model to generate and define the objects to have the same attributes and method”. The note reserved for this question is 3 points. The correction of this type of question starts with the verification of the spelling errors, the spelling errors, in this example we found two errors “defined” and “th”, after the syntax correction of this answer [28] we consider the new answer is “a class is a model to generate and define the objects for the same attributes and method”. Figure 4 demonstrate an example to calculate the similarity between response of teacher and student.


Student Sentence: A class is a model to generate and definie th objects for have the same attributes and method
Teacher Sentence: A class is a definition model for objects with the same set of attributes, and the same set of operations.
Step 1: Elimination of« stop words »
Step 2: Correction of errors (correction approach)
Class model generate define objects attributes method.
Class definition model objects same set attributes operations.
Step 3: Segmentation (OpenNLP(1))
Class/NN model/NN generate/VB define/VB objects/NN attributes/NN method/NN.
Class/NN definition/NN model/NN objects/NN same/JJ set/NN attributes/NN operations/NN
Step 4: Similarity Matrix Wu&Palmer (WordNet(2))
Step 5: Mihalcea similarity
Sim(S1, S2) = 0.71

| | Class/NN | Definition/NN | Model/NN | Objects/NN | Same/JJ | Set/NN | Attributes/NN | Operations/NN |
|---|---|---|---|---|---|---|---|---|
| Class/NN | 1 | 0.66 | 0.77 | 0.4 | - | 0.83 | 0.23 | 0.82 |
| Model/NN | 0.77 | 0.37 | 1 | 0.8 | - | 0.88 | 0.84 | 0.82 |
| Define/VB | - | - | - | - | - | - | - | - |
| Objects/NN | 0.4 | 0.47 | 0.8 | 1 | - | 0.66 | 0.69 | 0.76 |
| Attributs/NN | 0.23 | 0.66 | 0.84 | 0.69 | - | 0.75 | 1 | 0.72 |
| Method/NN | 0.42 | 0.37 | 0.57 | 0.51 | - | 0.48 | 0.58 | 0.62 |

Figure 4. Example of calculating the semantic similarity between two sentences sentence (student) and sentence (teacher)

If we have a similarity between the proposed answer and the student’s response less than 0.25, we consider that the answer is false, i.e., the grade that will be given to the student for this question is “0”. Otherwise, if the similarity exists in [0.25, 0.75], we consider that the answer is partially correct by a percentage that is to say the student’s note for this question equal to “sim * note by cons if the similarity is greater than or equal to 0.75, we consider that the answer is correct and the student will have the whole note. The similarity between the proposed answer and the student’s answer in the example equals ‘0.71’, so in this case the student’s score is ‘0.71*3=2.13’. Figure 5 explain the process to calculate the score of the student response.

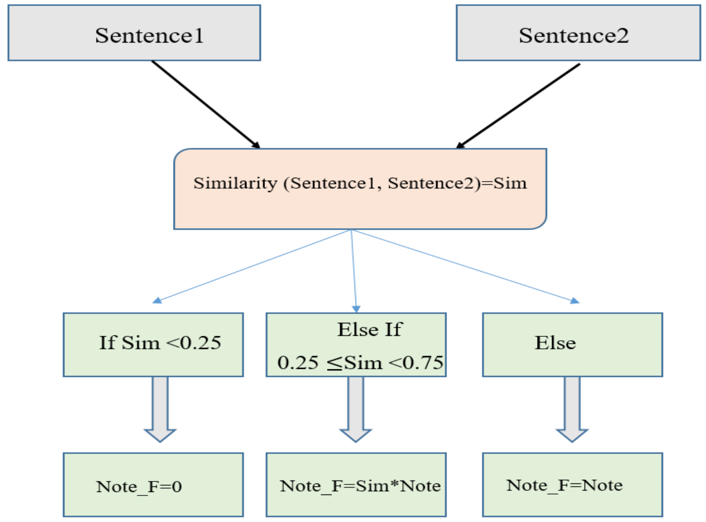


Figure 5. Flow chart that explains the calculation process

## 4. CONCLUSION

In this end-of-studies project, we have developed an intelligent evaluation system (SEI), which makes it possible to assess learners effectively; this effectiveness lies in the diversification of the types of questions to be proposed in an evaluation. The particularity of this system compared to the existing one is that it added other types of questions, namely: i) programming issues where the learner must answer the question through a program using a programming language and ii) open questions where the learner has to write the answer of the question as a text. The realization of this system required several steps; starting with the study of the existing, then looking for improvement as well as resolution approaches, and finally the choice of tools and the development of the system. Several perspectives are conceivable, namely: i) adapt this evaluation system to correct all types of questions of all subjects and ii) use multilingual dictionaries and corpora to correct answers to questions from different languages.

## REFERENCES

[1] B. Gao, “Research and Implementation of Intelligent Evaluation System of Teaching Quality in Universities Based on Artificial Intelligence Neural Network Model,” *Math. Probl. Eng.*, vol. 2022, p. e8224184, Mar. 2022, doi: 10.1155/2022/8224184.

[2] S. Abujar, M. Hasan, and S. A. Hossain, “Sentence Similarity Estimation for Text Summarization Using Deep Learning,” in *Proc. of the 2nd Inter. Conf. on Data Eng. and Comm. Tech.y*, Springer Singapore, 2018, pp. 155–164, doi: 10.1007/978-981-13-1610-4_16.

[3] X. Guo, H. Mirzaalian, E. Sabir, A. Jaiswal, and W. Abd-Almageed, “CORD19STS: COVID-19 Semantic Textual Similarity Dataset,” 2020, doi: 10.48550/ARXIV.2007.02461.

[4] Y. Wang *et al.*, “MedSTS: a resource for clinical semantic textual similarity,” *Lang. Resour. Eval.*, vol. 54, no. 1, pp. 57–72, Oct. 2018, doi: 10.1007/s10579-018-9431-1.

[5] C. Lo and M. Simard, “Fully Unsupervised Crosslingual Semantic Textual Similarity Metric Based on BERT for Identifying Parallel Data,” in *Proc. of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 2019, doi: 10.18653/v1/k19-1020.

[6] R. Yan, D. Qiu, and H. Jiang, “Sentence Similarity Calculation Based on Probabilistic Tolerance Rough Sets,” *Math. Probl. Eng.*, vol. 2021, pp. 1–9, Jan. 2021, doi: 10.1155/2021/1635708.

[7] N. Gali, R. Mariescu-Istodor, D. Hostettler, and P. Fränti, “Framework for syntactic string similarity measures,” *Expert Syst. Appl.*, vol. 129, pp. 169–185, Sep. 2019, doi: 10.1016/j.eswa.2019.03.048.

[8] A. Thakkar and K. Chaudhari, “Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks,” *Appl. Soft Comput.*, vol. 96, Nov. 2020, doi: 10.1016/j.asoc.2020.106684.

[9] A. Mahmoud and M. Zrigui, “Semantic Similarity Analysis for Paraphrase Identification in Arabic Texts,” in *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, Nov. 2017, pp. 274–281. Accessed: Aug. 18, 2022. [Online]. Available: https://aclanthology.org/Y17-1037

[10] L. Suryani and K. Edy, “Application development ‘Lost & Found’ android-based using term frequency-inverse document frequency (TF-IDF) and Cosine similarity method,” *Electro Luceat*, vol. 6, no. 2, pp. 190–204, Nov. 2020, doi: 10.32531/jelekn.v6i2.232.

[11] R. G. Pawar, S. U. Ghumbre, and R. R. Deshmukh, “Visual Similarity Using Convolution Neural Network over Textual Similarity in Content- Based Recommender System,” *Int. J. Adv. Sci. Technol.*, vol. 27, pp. 137–147, Sep. 2019.

[12] L. Zheng, K. Jia, W. Wu, Q. Liu, T. Bi, and Q. Yang, “Cosine Similarity Based Line Protection for Large Scale Wind Farms Part II—The Industrial Application,” *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2599–2609, Mar. 2022, doi: 10.1109/tie.2021.3069400.

[13] R. Zhang, Z. Xu, and X. Gou, “ELECTRE II method based on the cosine similarity to evaluate the performance of financial logistics enterprises under double hierarchy hesitant fuzzy linguistic environment,” *Fuzzy Optim. Decis. Mak.*, Feb. 2022, doi: 10.1007/s10700-022-09382-3.

[14] D. Liu, X. Chen, and D. Peng, "Some cosine similarity measures and distance measures between q-rung orthopair fuzzy sets," *Int. J. Intell. Syst.*, vol. 34, no. 7, pp. 1572–1587, Mar. 2019, doi: 10.1002/int.22108.

[15] M. AlMousa, R. Benlamri, and R. Khoury, "Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in WordNet," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, doi: 10.1016/j.knosys.2020.106565.

[16] M. Farouk, "Measuring Sentences Similarity: A Survey," *Indian J. Sci. Technol.*, vol. 12, no. 25, pp. 1–11, Jul. 2019, doi: 10.17485/ijst/2019/v12i25/143977.

[17] M. Farouk, "Sentence Semantic Similarity based on Word Embedding and WordNet," in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, Dec. 2018, doi: 10.1109/icces.2018.8639211.

[18] J. Kocoń and M. Maziarz, "Mapping WordNet onto human brain connectome in emotion processing and semantic similarity recognition," *Inf. Process. Manag.*, vol. 58, no. 3, p. 102530, May 2021, doi: 10.1016/j.ipm.2021.102530.

[19] W. H. Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, Apr. 2013, doi: 10.5120/11638-7118.

[20] Z. Wu and M. Palmer, "Verb semantics and lexical selection," *32nd Annu. Meet. Assoc. Comput. Linguist.*, pp. 133–138, 1994, doi: 10.3115/981732.981751.

[21] H. Zargayouna and S. Salotti, "Mesure de similarité dans une ontologie pour l'indexation sémantique de documents XML," in *15èmes Journées francophones d'Ingénierie des Connaissances*, Lyon, France, May 2004, pp. 249–260.

[22] I. Atoum and A. Otoom, "Efficient Hybrid Semantic Text Similarity using Wordnet and a Corpus," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 9, 2016, doi: 10.14569/IJACSA.2016.070917.

[23] R. Mihalcea, C. Courtney, and o S. Carlo, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity," *Am. Assoc. Artificial Intell.*, p. 6, 2006.

[24] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on-EACL '09*, 2009, doi: 10.3115/1609067.1609130.

[25] M. Biniz, R. E. Ayachi, and M. Fakir, "Ontology Matching using BabelNet Dictionary and Word Sense Disambiguation Algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 5, no. 1, p. 196, Jan. 2017, doi: 10.11591/ijeecs.v5.i1.pp196-205.

[26] H. Vu, "Random indexing and inter-sentences similarity applied to automatic summarization," 2016.

[27] C. Welch, C. Gu, J. Kummerfeld, V. Perez-Rosas, and R. Mihalcea, "Leveraging Similar Users for Personalized Language Modeling with Limited Data," *Proc. 60th Annu. Meet. Assoc. Comput. Linguist.*, vol. 1, pp. 1742–1752, 2022.

[28] O. Maarouf, R. E. Ayachi, and M. Biniz, "Correcting optical character recognition result via a novel approach," *Int. J. Inform. Commun. Technol. IJ-ICT*, vol. 11, no. 1, p. 8, Apr. 2022, doi: 10.11591/ijict.v11i1.pp8-19.

[29] O. Maarouf, M. Biniz, and R. El Ayachi, "A new Approach of Natural Language Processing to correct the Result of an OCR," *CBI Beni Mellal*, p. 6, Apr. 2018.

## 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 Sliman 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.

**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, and semantic web. He can be contacted at email: 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 Sliman Beni Mellal. He is a professor at polydisciplinary faculty University Sultan Moulay 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, and dynamic programming. He can be contacted at email: mohamedbiniz@gmail.com.