[1.]Doru Anastasiu POPESCU, [1.]Nicolae BOLD, [1.]Ion Alexandru POPESCU

STUDENT ASSESSMENT USING EDITING DISTANCE AND KEYWORDS MATCHING

[1.] National University of Science and Technology POLITEHNICA Bucharest, Faculty of Sciences, Physical Education and Computer Science, Department of Mathematics and Computer Science, Pitesti, ROMANIA

Abstract: This paper presents an automated assessment model that is based on assessment tests formed of Short-Answer Items (SAQ). This type consists in a textual answer with a given maximum length. In order to compare the correct answer with the one given by the student, an algorithm that computes the editing distance (also known as Levenshtein distance) between the two answers and the existence of given keywords (semantic tags) is used. Within the model, an item is considered to be answered correctly if the obtained editing distance is inferior to a threshold and the answer contains a set proportion or specific semantic tags. The model will also include the analysis of the optimal way of inputting the needed data by a user by introducing descriptive non-technical requirements. In this matter, the threshold for the editing distance and the usage of semantic tags will be established using natural language-based descriptors (e.g., scales, fuzzy, proportions), offering an intuitive non-technical modality of inputting data that is later mapped to the edit distance and required keywords. In order to implement the described model, a web application can be created. This web application extracts the items and the values needed in the assessment process from a database and computes the result automatically. The obtained data is then stored to be further analysed by an assessor or using an automated learning method.
Keywords: educational assessment, editing distance, semantic tag

1. INTRODUCTION & LITERATURE REVIEW

Automated short answer scoring is a growing area of interest in educational technology, aiming to improve the efficiency, objectivity, and scalability of the grading process. However, the complexity of natural language and the variability of student expressions require the development of flexible methods that overcome the limitations of strictly accurate scoring.

The model proposed in this paper integrates a fuzzy logic approach to determine the acceptability of a short answer, taking into account both structural similarity (through Levenshtein distance) and semantic validation (through the presence of predefined key terms). The tolerance level is expressed by the teacher in linguistic terms (e.g. strict, medium, permissive) and is translated into fuzzy membership functions that determine the maximum allowed error rate.

Automated short-answer grading [3] has been the subject of extensive research [1, 4] due to its potential to support scalable and objective educational assessment [7]. Traditional approaches, such as those based on string matching or bag-of-words representations, often fail to capture semantic similarity and tolerate acceptable linguistic variations. Some algorithms [5, 6] provide methodological inspiration for separating semantic layers or latent structures in more complex textual responses. Techniques such as edit distance metrics (e.g., Levenstein distance [14]) have been widely used for string similarity measurement, providing a quantitative estimation of the effort required to transform one string into another. While effective for detecting minor differences, these methods are sensitive to superficial changes and do not account for deeper semantic correctness. Although not directly related to fuzzy-based evaluation, work on maxentropic reconstruction of Markov chains [2] introduces concepts of information uncertainty and probabilistic structure which are conceptually adjacent to uncertainty modeling in fuzzy logic systems. While the model in this paper relies on fuzzy logic and semantic validation, earlier mathematical studies such as [12] explored multiobjective fractional optimization using bifunctions — a concept structurally similar to multi-criteria decision-making in automated evaluation systems.

To address the limitation of the usage of single Levenshtein distance, hybrid models combining syntactic similarity with semantic validation have emerged. One approach is the integration of keyword detection (e.g., semantic tags or concept labels) to ensure that core ideas are preserved in student responses. Recent research has explored the application of fuzzy logic [13] to model uncertainty and subjectivity in grading, allowing instructors to define tolerance levels using linguistic terms (e.g., “medium tolerance”) rather than technical thresholds. Fuzzy inference systems have proven effective in translating these linguistic preferences into numerical scoring models, bridging the gap between human judgment and automated decision-making [8, 9].

This work builds on the above by proposing a fuzzy-based evaluation model on previous models [10, 11] that incorporates both Levenshtein distance and keyword constraints. It introduces a defuzzification phase based on centroid computation via definite integrals, offering a mathematically transparent and adaptable framework for short-answer grading. The methodology aims to preserve flexibility, interpretability, and fairness in evaluation, particularly in educational contexts where nuanced language understanding is essential.

2. MODEL DESCRIPTION

The model presents the definition, the research methodology, the mathematical description and an example to illustrate the appliance to a real-world scenario. The next subsections present the enumerated elements of the model.

Problem definition

We can define the automatic assessment of short-answer items as a binary classification problem: a student response is either accepted as correct or rejected as incorrect, based on its textual similarity to a reference answer and the presence of required semantic concepts.

Research methodology

The followed methodology is based on quantitative approach (through distances, scores) combined with computational linguistics. The steps of the methodology are:

Step 1. Literature review: The analysis of existing methods: edit distance, fuzzy logic in evaluation was made.

Step 2. Formal modeling: The definition of variables, fuzzy functions, thresholds was made.

Step 3. Practice on examples: An application to real or simulated responses was made.

Step 4. Discussion and limitations: The interpretation of accepted / rejected cases was built and possible extensions of the problem were taken into account.

Mathematical description

The mathematical description of the model is part of the formal modeling of the research. The next subsection shows the problem statement, the presentation of input data, the variables, the relationships between them, the constraints and an exemplification of the model for a real-world scenario.

- Statement and input. Let R_s be the student's response and R_c the correct response. The goal is to evaluate whether R_s can be accepted as correct, based on its textual similarity to R_c and the presence of certain required semantic elements (semantic tags or keywords).

The input variables are expressed as follows:

≡ $R_C \in \Sigma^*$: the correct expected answer, established by the assessor, where Σ is the alphabet, meaning that the response is textual;

≡ $R_S \in \Sigma^*$: the student answer, given as a response to the SAQ;

≡ $T = \{t_1, t_2, \dots, t_k\} \in \Sigma^*$: the set of semantic tags / semantic elements / keywords desired to be present in the answer, where k is the number of the desired semantic tags;

≡ $\tau \in [1,5]$: the fuzzy threshold, which is a numeric value that represents intuitively the degree of permissiveness related to the comparison with the ideal response. From a scale from 1 to 5, a level of 1 represents a lower permissiveness (more strict assessment) and 5 represents a higher permissiveness (more permissive assessment).

- Variables. The variables taken into account serve as the variables used for the calculation of similarity degree and the presence of the semantic tags. In this matter, the variables are presented in Table 1.

Table 1. The variables of the model taken into account

	Symbol	Description	Possible values
1	R_C	The correct expected answer	Σ*
2	R_C'	The normalized form of R_C	Σ*
3	R_S	The student answer	Σ*
4	R_S'	The normalized form of R_S	Σ*
5	$L(R_C', R_S')$	The Levenshtein distance between the normalized answers	R_+
6	k	The number of semantic tags	N_+
7	$T = \{t_1, t_2, \dots, t_k\}$	The set of semantic tags	Σ*
8	τ	The level of permissiveness defined by the assessor	[1,5]
9	p	The maximum proportion of permitted error	[0.05,0.5]
10	θ	Final Levenshtein distance threshold	R_+
11	L_{max}	The maximum length of the response	N_+
12	$E(R_S)$	The status of response (incorrect / correct)	{0,1}

Observations related to some of the variables above must be made. In this matter, the normalization of R_C and R_S responses, represented by R_C' and R_S', is essential to reduce the influence of insignificant variations (e.g. spacing, capitalization, inflectional forms). Levenshtein distance, denoted L(R_C', R_S'), provides a quantitative measure of the difference between the expected response and the one provided by the student, playing a central role in automated assessment. The threshold θ is determined according to the permissiveness level τ and the maximum error proportion p accepted by the evaluator, which allows the assessment to be adapted to different educational contexts. The set of semantic labels and their number, k, can influence the degree of acceptance of alternative responses, especially in situations where the correct responses may vary lexically but are semantically equivalent. Thus, the final status of an answer, E(R_S), depends on both the calculated distance and the configuration parameters established by the evaluator.

— Relationships. The main relationships used in the mathematical model refer to functions of pre-processing, fuzzification, defuzzification and the calculation of the distance threshold. Thus, these functions are:

≡ $\phi: \Sigma^* \rightarrow \Sigma^*$: a pre-processing function which transforms the initial textual input data in processed data. The processing consists in the next operations: (1) lowercase transformation; (2) punctuation marks elimination; (3) stopwords elimination; (4) space normalization; (5) special character change (e.g., diacritics from Romanian are transformed into their correspondent in the English alphabet). This function is applied on R_C, R_S and T, thus obtaining the next relationships:

$$R_C{}' = \phi(R_C), R_S{}' = \phi(R_S), T' = \{\phi(t_i | t_i \in T)\} \quad (1)$$

≡ $L: \Sigma^* \times \Sigma^* \rightarrow \mathbb{N}$: a function that defines the computation of the Levenshtein distance between two strings. The function determines the minimum amount of operations made to reach from the first string to the second one. The operations are defined as: (1) INSERT (the addition of a character); (2) DELETE (the elimination of a character); (3) REPLACE (the change of a character with a different character). In our model, this function has the next form:

$$L(R_C{}', R_S{}') = D(|R_C{}'|, |R_S{}'|) \quad (2)$$

where function D is calculated recursively, *D(i,j)* being the distance between the first *i* characters from *R_C'* and the first *j* characters from *R_S'*:

$$D(i,j) = \begin{cases} i, j = 0 \\ j, i = 0 \\ d(i-1, j-1) + \delta(r_i, s_j), i > 0, j > 0 \end{cases} \quad (3)$$

and *δ(a,b)* is 0 if *a* = *b* and 1 if *a* ≠ *b*.

≡ $\mu: [1,5] \rightarrow [0,1]$: a fuzzy function that transforms the input of the user related to error permissiveness into a fuzzified value of it, which will be used in the dynamic determination of the threshold of error permitted by the assessor. The function is applied to the level of permissiveness τ and to the maximum proportion of error *p*. It has the next form:

$$\mu(x) = \max(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0) \quad (4)$$

where *x, a, b* and *c* are shown in Table 2. The effect for the τ value is that the comparison of the threshold is made based on a more granular determination rather than a crisp one and the assessor may input a value that is more natural to oneself (e.g., a numerical value from a range, a textual degree of permissiveness - strict, medium, permissive etc.) rather than an unknown measurement value. The effect for the *p* value is that the function is calculated based on the degree of error related to the response length (e.g., a Levenshtein distance of 2 is different for a text of length 10 characters than for a length 100).

≡ $\alpha_i: [1,5] \rightarrow [0,1]$: the activation of the fuzzy functions. Each fuzzy function μ_i evaluates the degree to which the chosen tolerance τ belongs to a linguistic label. This degree becomes the activation factor a_i for the corresponding fuzzy rule. In this matter:

$$\alpha_i = \mu_i(\tau), i \in \{strict, medium, permissive\} \quad (5)$$

≡ $\mu^{(\alpha_i)}: [0.05, 0.5] \rightarrow [0, \alpha_i]$: the output function clipping. The fuzzy output function $\mu_{Y_i}(p)$ is cut horizontally at the activation level a_i, where Y_i is the textual label of the rule, reflecting how much that rule contributes to the total output:

$$\mu^{(\alpha_i)}(p) = \min(\mu_{Y_i}(p), \alpha_i) \quad (6)$$

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

≡ $\mu_{agg}: [0.05{,}0.5] \rightarrow [0{,}1]$: the output function aggregation. All the clipped fuzzy functions are combined, taking the maximum value for each point *p*. This results in a single aggregate fuzzy function that synthesizes all the active rules:

$$\mu_{agg}(p) = \max_i(\mu^{(\alpha_i)}(p)) \quad (7)$$

≡ $d\mu: [0{,}1] \rightarrow [0.05{,}0.5]$: the defuzzification function. The center of gravity of the aggregate fuzzy function is calculated to obtain a single numerical value *p**, which represents the maximum allowable error proportion.

$$d\mu(\mathrm{p}) = p^* = \frac{\int_{0.05}^{0.5} p \times \mu_{agg}(p)dp}{\int_{0.05}^{0.5} \mu_{agg}(p)dp} \quad (8)$$

This is used to compute the absolute error threshold in characters, proportional to the length of the correct answer ($\theta = p^* \times |R_C{}'|$). It will be compared with the Levenshtein distance to decide whether an answer is acceptable.

≡ $E: \Sigma^* \rightarrow [0{,}1]$: the function of correct answer determination, which has value 1 if all the conditions from Equation (9) are true and 0 otherwise.

Table 2. The delimitation of the fuzzy functions parameters

	Fuzzy label	Variable	[a,b,c] parameters
1	strict	τ	[1,1,3]
2	medium	τ	[2,3,4]
3	permissive	τ	[3,5,5]
4	low	p	[0.05,0.05,0.15]
5	medium	p	[0.10,0.25,0.30]
6	high	p	[0.25,0.50,0.50]

— Fuzzy rules. A fuzzy rule R_i is a correspondence between a linguistic input value and a linguistic output value, mathematically transposed by a fuzzy implication function, shown in Equation (6).

A fuzzy rule has the next general form: IF *τ* is A_i THEN *p* is Y_i, where A_i is a linguistic term on input (A_i ∈ {strict, medium, permissive}) and Y_i is a linguistic term on output (Y_i ∈ {low, medium, high}). An example of a fuzzy rule is "IF *τ* is medium THEN *p* is medium". Thus, a fuzzy rule is a sentence that expresses the next fact: "if the teacher is moderately tolerant, then an average difference between answers is accepted"; the system does not apply the rule with the same force all the time, but only as the condition matches — and the result is used to decide how much error is allowed.

— Constraints. The constraints are related to the Levenshtein distance threshold, the presence of the semantic tags and the maximum length of the assessee response:

$$\begin{cases} (1): L(R_C{}', R_S{}') \le \theta \\ (2): \forall t_i \in T, t_i \in R_S{}' \\ (3): |R_S| \le L_{max} \end{cases} \quad (9)$$

— Model methodology. Related to the model, we will determine a formal methodology that must be followed for the model. In this matter, the next steps for the determination of the correctness of an answer are followed:

Step 1. The input data (R_C, R_S, *T* and *τ*) is read.

Step 2. The textual input data is pre-processed by applying the function ϕ.

Step 3. The dynamic Levenshtein distance threshold value θ is obtained, applying the next substeps:

a. The fuzzy function *μ* computation for the input data τ.
b. The determination of the activation values *a*.
c. The determination of the output function clipping.
d. The computation of the output function aggregation.
e. The computation of the Levenshtein distance threshold value θ based on the defuzzification function.

Step 4. The value of *E* function is obtained, thus the correctness of the given answer.

— Example. We will take an example related to the previous described steps. We have the next input values:

≡ R_S = "Artificial intelligence is transforming many industries";
≡ R_C = "Artificial intelligence transforms various industries";
≡ *T* = {"intelligence", "industries"}

 - ≡ τ = 3.4;
 - ≡ L_{max} = 80.

For these values, the next steps are applied:

≡ At Step 2, the input are pre-processed. Thus, R_S' = "artificial intelligence transforming industries" (|R_S'| = 44), R_C' = "artificial intelligence transforms various industries" (|R_C'| = 50) and T' = {"intelligence", "industries"}.

≡ At Step 3a, the fuzzy membership functions for given τ are computed. The next values are obtained: $\mu(\tau)_{strict}$ = 0, $\mu(\tau)_{medium}$ = 0.6 and $\mu(\tau)_{permissive}$ = 0.2.

≡ At step 3b, the activation values a are determined. The next values are obtained: a_{strict} = 0, a_{medium} = 0.6 and $a_{permissive}$ = 0.2.

≡ At Step 3c, the output function clipping values are determined. We obtain two truncated functions $\mu^{(a=0.6)}{}_{medium}$ = 0.6 and $\mu^{(a=0.2)}{}_{permissive}$ = 0.2, based on the calculated ones $\mu(p)_{medium}$ = 1.0 and $\mu(p)_{permissive}$ = 0.8.

≡ At Step 3d, the output function aggregation is computed. A function μ_{agg} for each interval [0.10, 0.20), [0.20, 0.25], (0.25, 0.30] and (0.30, 0.50] is computed.

≡ At Step 3e, the Levenshtein distance threshold value θ is computed. The p* value is obtained to be 0.2961. Then, the value of θ is 13.03.

≡ At Step 4, all the three constrains are verified:
 - ✓ The Levenshtein distance between the two strings $L(R_C, R_S)$ is 9. This value is compared to the value of θ. The constrain is true.
 - ✓ The two keywords are present in the answer. The constrain is true.
 - ✓ The length of the answer |R_C| = 53, lower than L_{max} = 80. The constrain is true.

The constrains are true, thus the answer is accepted as valid.

3. RESULTS AND DISCUSSIONS

In order to determine the effectiveness of the model, a set of answers related to a given answer was used and the presented methodology was run on this data set within a research empiric experiment. This dataset was built to test an automated short answer scoring model based on fuzzy logic, Levenshtein distance, and semantic validation. The input data was set up based on several hypotheses or assumptions related to pedagogical and technical aspects of answer check. Thus, the input data / input parameters used for the experiment were:

≡ R_S was extracted from a set of 1000 answers generated randomly which were labeled as True or False;

≡ R_C = "Artificial intelligence and machine learning improve data analysis";

≡ T = {"intelligence", "data", "learning"}

≡ τ = 3 / random in the interval [2.5, 4.5];

≡ L_{max} = 80.

The initial dataset of answers was generated randomly. The dataset structure was set up as containing the answer given by the student and its label as "True" or "False", depending on the correctness of the answer assessed initially. The description of the dataset is presented in Table 3.

Table 3. The description of the initial dataset

	Feature	Value
1	Number of answers	1000
2	Number of answers labelled "True"	700
3	Number of answers labelled "False"	300
4	Average length (after pre-processing)	44
5	Average of the presence of semantic tags in the answer	2.26
6	Average Levenshtein distance	15.99

Thus, the initial database contains answers that contain all 3 keywords and have a very small distance from the reference (mode = 1). Incorrect answers tend to completely lack semantic relevance or close linguistic structure. Based on these initial values, we have determined four experimental scenarios by varying two of the initial parameters:

≡ level of fuzzy tolerance, expressed by the value of τ, either fixed or random;

≡ semantic threshold, expressed by minimum number of semantic tags requested in the answer.

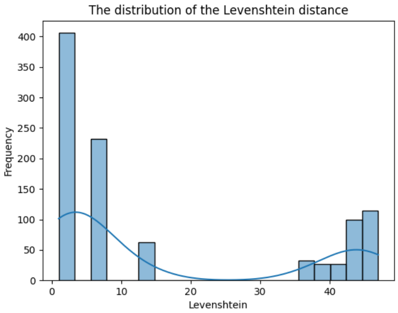
The four configurations are presented in Table 4.

Table 4. The description of the experimental scenarios run in the experiment

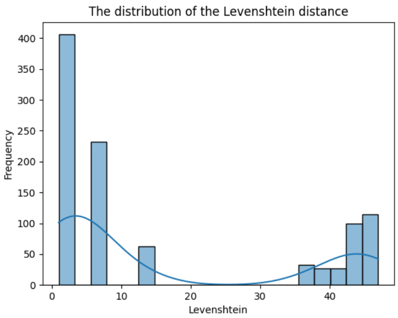
	Value of τ	Value of T	Tested aspect
	random	≥ 2 keywords	System behavior under flexible conditions
	3	≥ 2 keywords	How a constant fuzzy tolerance affects a permissive semantic system
	random	== 3 keywords	The impact of a strict semantic requirement in a fuzzy tolerant system
	3	== 3 keywords	Tests the limits of acceptance

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

The four experimental configurations combine different values of fuzzy tolerance (τ) and semantic strictness (T) to evaluate how the system behaves under flexible or constrained conditions, ranging from a fully adaptive setup to the strictest acceptance criteria. For the scenarios taken into account, several elements were studied from a statistical point of view.

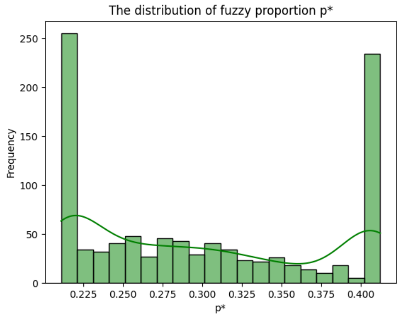


(a)

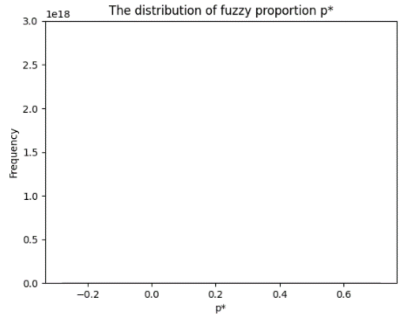


(b)

Figure 1. Levenshtein distance distribution for "≥ 2 keywords" condition when: (a) τ random and (b) τ = 3



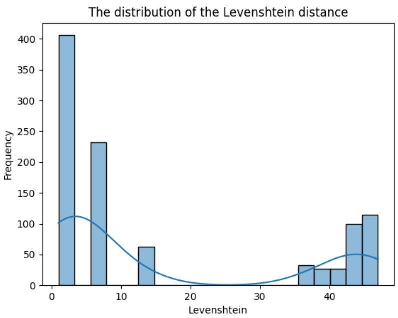
(a)



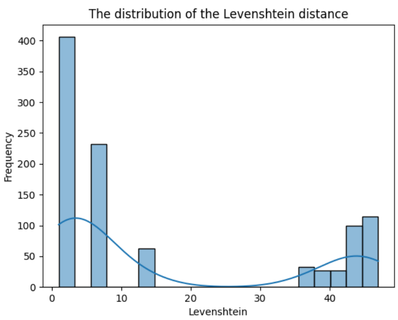
(b)

Figure 2. Fuzzy proportion distribution p^* for "≥ 2 keywords" condition when (a) τ random and (b) τ = 3

The comparison between the distributions when the condition "≥ 2 keywords" is applied shows that when τ is randomly varied, both the Levenshtein distances and fuzzy proportions p^* exhibit natural variability and spread, whereas with τ fixed at 3, the p^* values collapse into a constant, and the Levenshtein distances retain their bimodal pattern, highlighting the reduced adaptability of the system under fixed fuzzy tolerance.

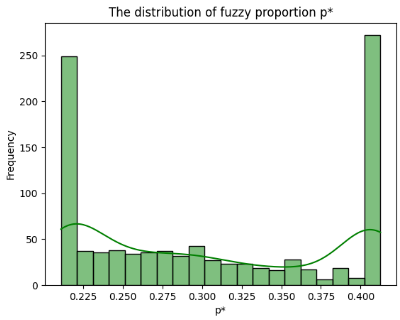


(a)

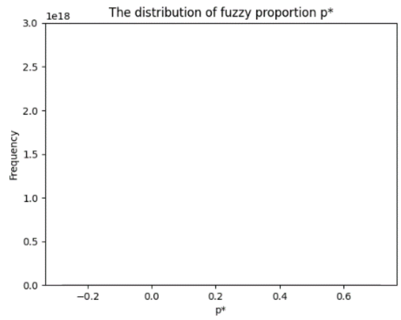


(b)

Figure 3. Levenshtein distance distribution for "== 3 keywords" condition when: (a) τ random and (b) τ = 3



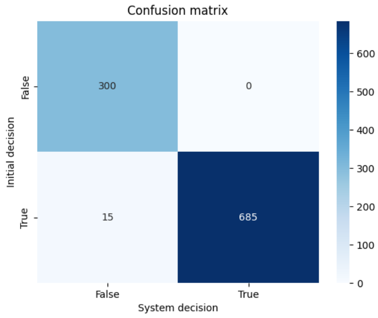
(a)



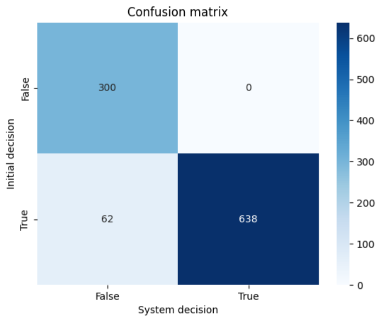
(b)

Figure 4. Fuzzy proportion distribution p^* for "== 3 keywords" condition when (a) τ random and (b) τ = 3

The four distributions illustrate that, under the strict semantic condition (== 3 keywords), fixing τ at 3 leads to a single-valued fuzzy threshold and a static p* distribution, while using random τ introduces a spread of p* values that increases flexibility; in both cases, the Levenshtein distance remains bimodal, reflecting the clear separation between correct and incorrect responses.

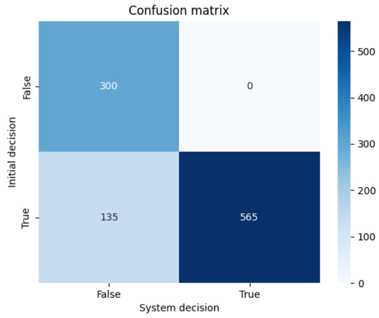


(a)

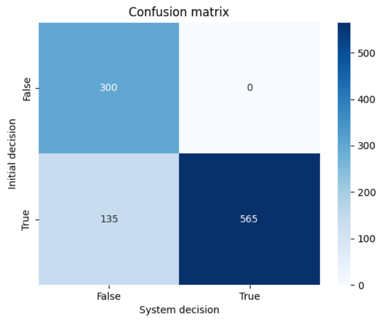


(b)

Figure 5. Confusion matrix for "≥ 2 keywords" when: (a) τ random and (b) τ = 3



(a)



(b)

Figure 6. Confusion matrix for "== 3 keywords" when: (a) τ random and (b) τ = 3

The confusion matrices reveal that the combination of random fuzzy tolerance (τ) with a permissive semantic rule (≥ 2 keywords) achieves the best balance between sensitivity and precision, while increasing semantic strictness (== 3 keywords) significantly raises the number of false negatives – especially when τ is fixed – indicating that both reduced flexibility and stricter concept matching limit the system's ability to recognize valid, though slightly varied, answers.

Table 5. Values of performance indicators for the four scenarios

	Performance indicator	≥ 2 keywords		== 3 keywords	
		τ = random	τ = 3	τ = random	τ = 3
1	Accuracy	0.985	0.938	0.865	0.865
2	Precision	1.0	1.0	1.0	1.0
3	Recall	0.9786	0.9114	0.8071	0.8071
4	F1 Score	0.9892	0.9537	0.8933	0.8933
5	True Positives (TP)	685	638	565	565
6	True Negatives (TN)	300	300	300	300
7	False Positives (FP)	0	0	0	0
8	False Negatives (FN)	15	62	135	135

The performance results show that a permissive semantic rule (≥ 2 keywords) combined with a variable fuzzy tolerance (τ random) yields the highest accuracy (98.5%) and recall (97.9%), while enforcing a strict semantic condition (== 3 keywords) substantially increases false negatives—especially under fixed tolerance (τ = 3)—leading to a significant drop in recall (80.7%) and F1 score (89.3%), despite perfect precision across all scenarios.

All experimental results show that the proposed model is accurate, adaptable and robust, managing to correctly classify the majority of responses, with optimal performance when operating in adaptive and semantically permissive regime, and offering full control by adjusting the parameters τ and T according to pedagogical requirements.

4. CONCLUSIONS

Experimental results demonstrate that the proposed automatic assessment system, based on Levenshtein distance and fuzzy logic, offers an effective balance between rigor and flexibility. The four tested scenarios, resulting from the combination of fuzzy tolerance (τ) with semantic threshold (minimum number of keywords), highlight that maximum performance is obtained when

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

the system operates in an adaptive regime (random τ) and with a permissive semantic rule (≥ 2 keywords). However, for contexts where semantic accuracy is essential, applying a strict threshold (== 3 keywords) ensures total precision, at the expense of recall. Statistical analysis and the distributed behavior of the fuzzy metric p^* and the editing distance support the robustness of the model, and confusion matrices confirm the system's ability to clearly differentiate between relevant and irrelevant responses. Thus, the model is versatile and can be easily adjusted depending on the purpose of the assessment – either formative or summative.

References

[1] Bacon D R 2003 Assessing learning outcomes: A comparison of multiple-choice and short-answer questions in a marketing context, Journal of Marketing Education 25(1) 31–36

[2] Balcau C 2004 Maxentropic reconstruction of some homogeneous Markov chain in the countable case, Math. Rep. (Bucur.) 6(56) 9–19

[3] Brew C and Leacock C 2013 Automated short answer scoring: Principles and prospects, Handbook of Automated Essay Evaluation 136–152 Routledge

[4] Burrows S, Gurevych I and Stein B 2015 The eras and trends of automatic short answer grading, International Journal of Artificial Intelligence in Education 25 60–117

[5] Cocianu C, Constantin D and Sararu C 2009 Partially Supervised Approach in Signal Recognition, Informatica Economica 13(3) 153

[6] Constantin D and Hamzh E 2012 An Improved Algorithm for Dynamic Independent Component Analysis, AL-Qadisiyah Journal For Administrative and Economic Sciences 14(4)

[7] Kizlik B 2012 Measurement, Assessment and Evaluation in Education Robert Kizlik and Associates, Boca Raton, Florida. Retrieved from http://www.adprima.com

[8] Krouska A, Troussas C and Sgouropoulou C 2019 Fuzzy logic for refining the evaluation of learners' performance in online engineering education, European Journal of Engineering and Technology Research 4(6) 50–56

[9] Machado M A S, Moreira T D R G, Gomes L F A M, Caldeira A M and Santos D J 2016 A fuzzy logic application in virtual education, Procedia Computer Science 91 19–26

[10] Nijloveanu D, Bold N and Popescu I A 2016 Model of evaluation using questions with specified solving time, eLearning & Software for Education (1)

[11] Popescu D A, Tiţa V and Bold N 2016 The development of a web application for assessment by tests generated using genetic-based algorithms. In CEUR Workshop Proceedings 2121 Paper 5, pp. 37–46)

[12] Preda V, Balcau C and Niculescu C 2012 On multiobjective fractional programming involving generalized d-type-I relative to bifunctions and related functions, Mathematical Reports 14(1) 95–106

[13] Zadeh L A 1988 Fuzzy logic, Computer 21(4) 83–93

[14] Zhang S, Hu Y and Bian G 2017 Research on string similarity algorithm based on Levenshtein Distance, IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) 2247–2251

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665