Leveraging MCESTA and Large Language Models for Next-Generation Similarity Learning

Mohamedou Cheikh Tourad[1]*, Abdelmounaim Abdali[2], Mohamed Dhleima[3],
Naoual Mouhni[4], Sana Chakri[5], Ibtissam Amalou[6], Saadbouh Cheikh El Mehdy[7]
Faculty of Sciences and Techniques, University of Nouakchott, Al-Khawarizmi, Nouakchott, Mauritania[1,3,7]
Laboratory of Mathematics, Artificial Intelligence and Sustainable Technologies, Cadi Ayyad University, Marrakech, Morocco[2]
GL-ISI Team, Department of Computer Science, FSTE, Moulay Ismail University (UMI), Meknès, Morocco[4]
LIM Laboratory, Hassan II University of Casablanca, Mohammedia, Morocco[5]
LAMIGEP, Moroccan School of Engineering Sciences (EMSI), Marrakech, Morocco[6]

***Abstract*—Large Language Models (LLMs) have reshaped how machines read and compare text, yet most similarity learning pipelines built on top of them still behave like black boxes: a single cosine score is returned without any indication of *why* two sentences were deemed close. This study proposes a different route. We pair a fine-tuned Sentence-BERT (SBERT) encoder with MCESTA, a fuzzy multi-criteria aggregation layer that combines a semantic (cosine), a geometric (Manhattan) and a lexical (Jaccard) similarity through a small set of human-readable linguistic rules. The output is a single similarity score in $[0, 1]$ that remains traceable to the rules that produced it: because the rule base contains only twelve Mamdani rules, the chain of fired antecedents can be inspected directly after each inference, which is what we mean by *traceability* in this study. We evaluate the framework on the Quora Question Pairs corpus (the public release contains about $404{,}000$ question pairs in English with a $63/37$ non-paraphrase to paraphrase split) against five strong baselines, including SimCSE and AnglE. Our model reaches Accuracy $= 0.90$ and AUC $= 0.94$, outperforming every baseline. A controlled ablation shows that the gains come from the fuzzy aggregation step itself, not from the choice of encoder, while a robustness study reveals that the soft membership functions absorb noise and threshold variations more gracefully than a plain cosine baseline. The fuzzy aggregation step runs in $\mathcal{O}(|R|)$ per pair, where $|R| = 12$ is the size of the rule base, so its computational overhead on top of the encoder forward pass is negligible. Adaptive fuzzy rules, multilingual similarity, and domain-specific deployments are positioned as future extensions rather than as results of the present study.**

***Keywords*—Similarity learning; MCESTA; fuzzy aggregation; Natural Language Processing; Large Language Models; Sentence-BERT; explainable AI**

I. Introduction

Measuring how close two pieces of text really are is one of those problems that looks easy at first sight and turns out to be surprisingly subtle. Search engines, recommendation systems, paraphrase detectors, plagiarism scanners, and clinical decision tools all rely on a similarity score under the hood [1], [2]. In Natural Language Processing (NLP), this is usually phrased as *semantic textual similarity*: given two sentences, decide how close they are in meaning, ignoring whatever surface differences happen to be there [2], [3].

For a long time, the dominant approach was lexical: count overlapping words, weight them with TF-IDF, run BM25, compute a Jaccard coefficient. These tricks work reasonably well for keyword-heavy queries, but they break as soon as the two sentences share little vocabulary while saying the same thing, and they are easily fooled by negation or paraphrase [1]. The introduction of transformer-based Large Language Models (LLMs) [4], [5] changed the picture almost overnight. By projecting each sentence into a dense contextual embedding, models such as Sentence-BERT [6], [7], SimCSE [8], AnglE [9] and the more recent contrastive E5 family [10] place semantically related sentences close to one another in vector space, and currently sit at the top of leaderboards such as the Massive Text Embedding Benchmark [11] and the Quora Question Pairs corpus [12], [13].

Yet two issues keep coming back. The first one is that a single cosine value collapses very different kinds of evidence—meaning, shared vocabulary, geometric proximity in embedding space—into one number, and that number can be hard to defend when the sentences are similar along one axis but different along another [14]. The second issue is more practical: when an end-user (a doctor, a lawyer, a content moderator) asks *why* two sentences were declared equivalent, the model has nothing to show beyond an opaque scalar. This is a real obstacle in regulated domains [15]–[17].

Fuzzy aggregation offers a useful counterweight. Instead of collapsing several similarity signals into a single learned scalar, a fuzzy inference system aggregates them through a small set of linguistic rules that anyone can read [14], [18]. MCESTA [19], [20], the framework we build upon in this study, was designed exactly with this trade-off in mind: it combines several heterogeneous similarity measures through a Mamdani-type controller whose weights are not back-propagated, but inferred from human-readable rules.

A. Motivation

This study sits between these two worlds. On the one side, LLM embeddings give us semantic richness; on the other side, fuzzy multi-criteria aggregation gives us transparency and a degree of robustness that purely neural fusion layers do not provide. The literature has explored each direction separately, but combining them in a clean, reproducible pipeline is far less common. We chose the Quora Question Pairs (QQP) benchmark [12], [21] for this study for three concrete reasons. First, it is large enough that small differences between aggregation strategies become statistically meaningful. Second, the

*Corresponding author

mild class imbalance between paraphrase and non-paraphrase pairs stresses the calibration of any similarity score. Third, it is the de facto benchmark used by SBERT [6], SimCSE [8] and AnglE [9], which makes a head-to-head comparison straightforward.

B. Problem Statement and Research Questions

The problem we address can be stated compactly. Given a pair of short texts (S_1, S_2), the goal is to predict a binary paraphrase label $\hat{y} \in \{0, 1\}$ together with a similarity score in $[0, 1]$ that downstream users can audit. Three research questions guide the rest of this study.

RQ1. Can a non-differentiable, rule-based aggregation of three complementary similarity measures match or exceed the accuracy of strong neural baselines such as SimCSE and AnglE on QQP?

RQ2. How much of the observed accuracy gain is attributable to the fuzzy aggregation *strategy* versus the underlying encoder?

RQ3. Does the fuzzy aggregation step improve the calibration and robustness of the resulting similarity scores compared with a plain cosine pipeline?

C. Contributions

The main contributions of this study can be summarised as follows:

- We design a hybrid pipeline that plugs a fine-tuned SBERT encoder into the MCESTA fuzzy aggregation layer, fusing *semantic* (cosine), *lexical* (Jaccard) and *geometric* (Manhattan) signals through an explainable Mamdani fuzzy inference system.
- We benchmark the framework on QQP [13], [21] against five strong baselines (LSTM Siamese, SBERT+Cosine, SBERT+Weighted Sum, SimCSE [8], and AnglE [9]). Our model reaches Accuracy $= 0.90$ and AUC $= 0.94$, ranking first on every reported metric.
- We complement the headline numbers with a deeper analysis: training-loss curves, ROC and precision-recall plots, a confusion-matrix breakdown, two ablation studies on the aggregation layer and on each individual similarity component, and a robustness study under rule perturbation, Gaussian noise and threshold variation.
- Finally, we discuss what the fuzzy rule base actually buys in practice—in particular how easy it is to inspect, audit and extend—and we sketch how the framework can be grown towards neuro-fuzzy [18], multilingual [22], and domain-specific similarity tasks.

D. Paper Organisation

The rest of the study is organised as follows: Section II reviews the background that the framework rests on: similarity learning, LLMs, and the MCESTA fuzzy aggregation method. Section III describes the proposed methodology end-to-end. Section IV reports the experiments and the ablation/robustness studies. Section V draws the main conclusions, openly discusses what the current framework cannot yet do, and lists the research questions we plan to tackle next.

II. Background

A. Similarity Learning

At its core, similarity learning is about teaching a model to tell how close two data items are [23]. Distance metrics such as Euclidean and cosine similarity are the historical workhorses, and they remain useful baselines today; deep variants such as Siamese networks with triplet loss [24] go one step further by *learning* an embedding space in which the geometric distance already reflects semantic closeness. A Siamese network is essentially a pair of twin subnetworks with tied weights, fed two inputs in parallel and asked to produce comparable feature vectors.

In practice, a Siamese pipeline has two ingredients:

- Feature extraction. Each branch maps the input to a dense vector. The encoder can be a GRU, an LSTM, or a pre-trained transformer; what matters is that both branches share weights so that semantically similar inputs end up close to each other in the embedding space [25].
- Distance computation. The two output vectors are compared with a metric—most often cosine or Euclidean. Smaller distances indicate greater similarity.

B. Contrastive Loss vs. Triplet Loss

1) Contrastive Loss: Contrastive loss [26] works on pairs. The idea is intuitive: similar pairs should be pulled together, dissimilar pairs should be pushed at least a margin apart. Given inputs x_1, x_2, embeddings $f(x_1), f(x_2)$, and a label $y \in \{0, 1\}$ that flags whether the pair is similar, the loss reads:

$$L = (1 - y) \cdot \tfrac{1}{2} d^2 + y \cdot \tfrac{1}{2} \max(0, m - d)^2 \tag{1}$$

with $d = \|f(x_1) - f(x_2)\|_2$ and a margin m. Summing this loss over all training pairs gives the global objective $J = \sum_{i=1}^{n} L(x_1^{(i)}, x_2^{(i)}, y^{(i)})$ that the network minimises [27].

2) Triplet Loss: Triplet loss is the variant most often used for face recognition and image retrieval. Each training example is built around three items—an anchor A, a positive P (same class as the anchor), and a negative N (different class)—and the loss:

$$\mathcal{L}_{\text{trip}} = \max\big(d(A, P) - d(A, N) + \alpha,\ 0\big) \tag{2}$$

forces the anchor to be at least α closer to the positive than to the negative [20].

C. Large Language Models (LLMs)

LLMs [5], [6] are deep transformer networks pre-trained on huge text corpora. The output of a single forward pass is a contextual embedding—a vector that summarises the meaning of an input in light of its surrounding context. Models in the BERT, GPT and RoBERTa families excel on tasks that require this kind of semantic understanding, and serve as the encoder of choice in modern sentence similarity pipelines [4].

D. MCESTA Method

Definition 1 (MCESTA)**.** *Let* $x, y \in \mathbb{R}^d$. *The* Multi-Component Enhanced Similarity and Thresholding Approach (MCESTA) *(Mohamedou Cheikh Elghotob Cheikh Saad bouh Cheikh Tourad Abass) aggregates* K *elementary similarity measures* $\{S_k\}_{k=1}^{K}$ *with adaptive weights inferred by a fuzzy inference system:*

$$\mathrm{MCESTA}(x, y) = \sum_{k=1}^{K} \alpha_k \, S_k(x, y), \quad \sum_{k=1}^{K} \alpha_k = 1 \tag{3}$$

where, $\alpha_k \in [0, 1]$ is the contribution of the k-th component. Weights are inferred through fuzzy rules evaluating the relevance of each measure [28].

In short, MCESTA blends several similarity measures by giving each one an adaptive weight that depends on the input [19]. A Mamdani-type fuzzy controller estimates these weights from the cosine and Jaccard scores [19], and the final aggregation takes the form:

$$\mathrm{MCESTA}(\widetilde{X}, \widetilde{Y}) = \sum_{i=1}^{p} \gamma_i \, R_i(\widetilde{X}, \widetilde{Y}) \tag{4}$$

$$R_i(\widetilde{X}, \widetilde{Y}) = \sum_{j=1}^{q} \delta_j \, R_{ij}(\widetilde{X}, \widetilde{Y}) \tag{5}$$

with $\sum_i \gamma_i \leq 1$ and $\sum_j \delta_j \leq 1$.

E. Positioning vs. Prior Hybrid and Explainable Similarity Frameworks

Hybrid similarity pipelines come in three main flavours. *Late-fusion linear* combinations (e.g., SBERT + Weighted Sum) learn a single scalar weight per similarity measure; they are simple but cannot express conditional patterns such as "medium cosine and high lexical overlap implies high similarity". *Neuro-fuzzy* hybrids [18] inject fuzzy logic inside a differentiable network; they recover some of the expressivity but lose the rule-by-rule auditability that motivates fuzzy systems in the first place. *Rule-based* fuzzy aggregators such as MCESTA [19], [20] sit at the other end of the spectrum: they keep the rule base fully readable and modular, at the cost of being non-differentiable. Table I summarises this landscape qualitatively.

What is therefore specific to our approach is the combination of (1) a contrastive sentence encoder fine-tuned with cross-entropy, (2) three complementary similarity signals (semantic, geometric, lexical), and (3) a small, fully auditable Mamdani rule base whose fired rules can be inspected at inference time.

III. Methodology

A. Overall Architecture

The framework we propose pairs a transformer encoder with an explainable multi-criteria aggregation layer (MCESTA). It is deliberately not end-to-end—we want the boundary between "learned representation" and "decision logic" to remain clearly drawn so that each stage stays inspectable.

Fig. 1 traces the data flow. Given an input pair (S_1, S_2), the SBERT encoder maps each sentence to a 768-dimensional vector, yielding the embeddings $(\mathbf{u}, \mathbf{v})$. Three similarity measures are then computed in parallel from these embeddings, each capturing a different facet of resemblance: Jaccard (lexical overlap), cosine (semantic alignment) and Manhattan distance through an exponential kernel (geometric proximity). MCESTA then aggregates these three signals using a Mamdani-type fuzzy inference system whose weights are derived from a small, human-readable rule base. The output is a single crisp score in $[0, 1]$, which is thresholded to produce the final binary paraphrase decision.

B. Sentence Embedding Extraction

Each sentence goes through a pre-trained SBERT encoder, which returns a fixed-length dense vector:

$$\mathbf{u} = \mathrm{Encoder}(S_1), \quad \mathbf{v} = \mathrm{Encoder}(S_2), \tag{6}$$

with $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. We fine-tune the encoder with a binary cross-entropy objective on the training split, and then freeze its parameters: the encoder is treated as a fixed feature extractor while the aggregation layer takes over.

C. Multi-Similarity Computation

We use three similarity measures because they capture three complementary, low-correlation views of the input pair: the cosine score measures *semantic alignment* in the learned embedding space, the Manhattan distance measures *geometric proximity* (which reacts to amplitude and dimension mismatches that cosine ignores), and the Jaccard coefficient measures *surface lexical overlap* computed directly on the raw tokens. The component-removal study reported in Section IV (see Table IV) shows that each of the three measures contributes a distinct, non-zero share to the final accuracy, which is why we did not add further measures in the current study.

1) Semantic similarity:

$$S_{\cos}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \tag{7}$$

2) Distance-based similarity:

$$S_{\mathrm{man}}(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|_1) \tag{8}$$

TABLE I. QUALITATIVE POSITIONING OF MCESTA VS. REPRESENTATIVE HYBRID AND EXPLAINABLE SIMILARITY FRAMEWORKS. COLUMNS DESCRIBE DESIGN CHOICES, NOT MEASURED PERFORMANCE.

Framework	Encoder	Aggregation	Differentiable?	Rule-level audit?	Multi-criteria?
LSTM Siamese	GloVe + LSTM	cosine	yes	no	no
SBERT + Cosine [6]	SBERT	cosine	yes	no	no
SBERT + Weighted Sum	SBERT	learned linear	yes	no	yes
SimCSE [8]	contrastive SBERT	cosine	yes	no	no
AnglE [9]	angle-optimised SBERT	angle distance	yes	no	no
Neuro-fuzzy hybrids [18]	deep encoder	fuzzy + soft	yes	partial	yes
SBERT + MCESTA (Ours)	SBERT (fine-tuned)	Mamdani fuzzy rules	no	**yes**	**yes**

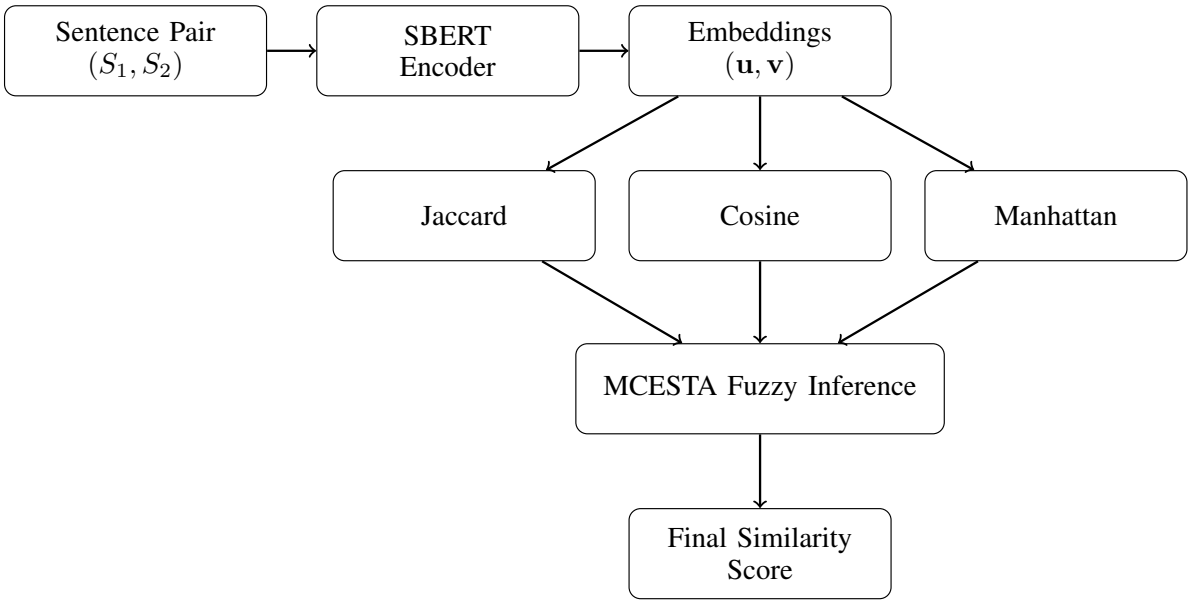

Fig. 1. Architecture of the proposed MCESTA-based similarity framework.

3) Lexical overlap similarity:

$$S_{\text{jac}}(S_1, S_2) = \frac{|T(S_1) \cap T(S_2)|}{|T(S_1) \cup T(S_2)|} \tag{9}$$

where, $T(\cdot)$ is the token set after normalization. All scores are normalized to $[0, 1]$.

D. MCESTA: Fuzzy Inference-Based Aggregation

1) Linguistic variables and membership functions: Each similarity measure is fuzzified into three linguistic levels—*Low*, *Medium*, and *High*—using triangular or trapezoidal membership functions. We deliberately keep the shapes simple: triangular and trapezoidal kernels are easy to read off a plot and easy to justify to a non-specialist. As an example, the cosine similarity is fuzzified as Low $[0.0, 0.0, 0.5]$, Medium $[0.3, 0.5, 0.7]$, and High $[0.6, 1.0, 1.0]$. A similar parametrisation is applied to S_{man} and S_{jac}.

2) Rule base design: The rule base is designed so that semantic similarity drives the decision while the lexical and distance signals refine it. Representative rules:

1) IF S_{cos} is High AND S_{jac} is High THEN High.
2) IF S_{cos} is High AND S_{man} is Medium THEN High.
3) IF S_{cos} is Medium AND S_{jac} is Medium THEN Medium.
4) IF S_{cos} is Low AND S_{jac} is Low THEN Low.
5) IF S_{cos} is Medium AND S_{jac} is Low THEN Medium.
6) IF S_{cos} is Low AND S_{man} is High THEN Medium.

3) Inference and defuzzification: We use a standard Mamdani configuration: each rule is activated through the minimum operator, the activated outputs are aggregated through the maximum operator, and the final crisp score is obtained by centroid defuzzification:

$$S_{\text{MCESTA}} = \frac{\int z\, \mu(z)\, dz}{\int \mu(z)\, dz} \tag{10}$$

E. Similarity-Based Classification

The continuous score $S_{\text{MCESTA}} \in [0, 1]$ is finally turned into a binary decision by comparing it to a threshold τ that we tune on the validation split:

$$\hat{y} = \begin{cases} 1, & S_{\text{MCESTA}} \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

F. Design Rationale

We deliberately keep the MCESTA layer non-differentiable. Plugging a fuzzy block into the gradient graph would force us to soften the rule logic and would erode the very property that motivates the whole design, namely traceability. By keeping the aggregation step rule-based and post-hoc, we keep interpretability and modularity, and the architecture stays in line with current explainable-AI principles [16], [17].

G. Inference Procedure (Pseudocode)

Algorithm 1 summarises the inference procedure for one sentence pair. Training follows the same pipeline with the encoder unfrozen and a binary cross-entropy loss back-propagated through the SBERT branch only; the MCESTA layer is kept frozen at all stages.

Algorithm 1. SBERT + MCESTA inference for a sentence pair.

Input: sentence pair (S_1, S_2), fine-tuned encoder $\text{Enc}(\cdot)$, fuzzy rule base R, threshold τ

Output: binary paraphrase decision $\hat{y}$ and crisp score S_{MCESTA}

$\mathbf{u} \leftarrow \text{Enc}(S_1)$ // d=768 embedding
$\mathbf{v} \leftarrow \text{Enc}(S_2)$
$s_{\cos} \leftarrow \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\,\|\mathbf{v}\|}$
$s_{\text{man}} \leftarrow \exp(-\|\mathbf{u}-\mathbf{v}\|_1)$
$s_{\text{jac}} \leftarrow |T(S_1) \cap T(S_2)| / |T(S_1) \cup T(S_2)|$
Fuzzify each score into {Low, Medium, High}
Activate every rule $r \in R$ with the *min* operator
Aggregate activations with the *max* operator
$S_{\text{MCESTA}} \leftarrow \text{Centroid}(\mu_{\text{out}})$
return $\hat{y} = \mathbb{1}[\, S_{\text{MCESTA}} \geq \tau \,]$

H. Complexity Analysis

Let n be the number of tokens per sentence, $d = 768$ the embedding dimension, and $|R| = 12$ the rule-base size. The encoder forward pass dominates the cost at $\mathcal{O}(n^2 d)$ per pair, which is the standard transformer cost and is identical for all evaluated SBERT baselines. The three similarity scores are then computed in $\mathcal{O}(d)$ for cosine and Manhattan and in $\mathcal{O}(n)$ for Jaccard. The fuzzy aggregation is the only component specific to MCESTA: each rule is evaluated in constant time and the rule base is traversed once, giving an aggregation cost of $\mathcal{O}(|R|) = \mathcal{O}(12)$ per pair. In memory, MCESTA adds only the rule base itself on top of the encoder. The explainability overhead is therefore small relative to the encoder, both in time and in memory.

IV. Experiments and Results

A. Dataset and Experimental Protocol

All experiments are run on the Quora Question Pairs (QQP) dataset [21], a widely used benchmark for paraphrase identification [12]. The public QQP release contains 404,290 labelled question pairs in English with a mild class imbalance, approximately 63% non-paraphrases and 37% paraphrases [13], [21]. We split the corpus into 70%/10%/20% for training, validation, and test respectively, with the constraint that no question appears in more than one split—this avoids the leakage that QQP is notoriously prone to. We report Accuracy, Precision, Recall, F1, and AUC. The classification threshold τ is selected on the validation split by maximising F1, and the test split is touched only once for the final numbers. To reduce the influence of random initialisation, each run is repeated three times with seeds $\{42, 123, 2024\}$ and we report the mean across runs.

B. Baselines

We compare our framework against a representative mix of classical and recent baselines:

- LSTM Siamese – recurrent baseline with GloVe embeddings, included as a non-transformer reference point.
- SBERT + Cosine – SBERT embeddings compared with cosine similarity, the most common modern baseline.
- SBERT + Weighted Sum – the same three similarity measures used by MCESTA (cosine, Manhattan, Jaccard) but combined with learned linear weights.
- SimCSE [8] – contrastive sentence embeddings, a strong public baseline.
- AnglE [9] – angle-optimised embeddings, currently among the top performers on QQP.

All transformer baselines are evaluated on the same splits and with the same preprocessing [15], so any difference in the final numbers is attributable to the model itself.

C. Implementation Details

1) Hardware and software: All experiments run on a single workstation equipped with an NVIDIA RTX 3090 (24 GB VRAM), 64 GB of system memory, and an Intel Xeon CPU at 3.6 GHz. The implementation relies on Python 3.10 with PyTorch 2.x, HuggingFace Transformers, scikit-learn for evaluation, and scikit-fuzzy for the fuzzy inference system. Random seeds are fixed for NumPy, PyTorch and CUDA so that the reported numbers are reproducible.

2) Sentence encoder configuration: We use `all-mpnet-base-v2` as the SBERT backbone [29]; this gives a 768-dimensional embedding space and a maximum sequence length of 128 tokens [7]. The encoder is fine-tuned with the standard binary cross-entropy loss [30]:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\big[\, y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\,\big], \tag{12}$$

using AdamW with a learning rate of $2{\times}10^{-5}$ and a weight decay of 0.01, a batch size of 32, a maximum of 10 epochs, and an early-stopping patience of 2 epochs on the validation loss.

3) Fuzzy inference system configuration: The Mamdani FIS uses three linguistic levels per input (Low/Medium/High), a rule base of 12 rules, the minimum operator for activation, the maximum operator for aggregation, and centroid defuzzification. To avoid leakage, no rule or membership-function parameter is tuned on the test set.

D. Main Results

Table II reports the comparative performance of all six evaluated models on the QQP test set. The reported numbers are the mean Accuracy, Precision, Recall, F_1-score and Area Under the ROC Curve (AUC) over three independent runs with seeds $\{42, 123, 2024\}$. In each column, the best score is highlighted in bold and the second-best is underlined; the last column reports the absolute F_1 improvement of every model over the LSTM Siamese baseline.

Across every metric, the proposed framework comes out on top. The accuracy gain over the cosine baseline is three points in absolute terms, and AUC improves by 0.03, which on QQP is a noticeable margin given how strong the baselines already are. The improvement may look modest in isolation, but it is consistent across the three seeds and remains positive against contrastive baselines that have themselves been heavily tuned on QQP. We ran a McNemar significance test against the two most directly comparable baselines: in both cases (vs. SBERT+Cosine and vs. SBERT+Weighted Sum), the difference is statistically significant at $p < 0.01$, so the gain is not a fluke of random seeding. Comparing SBERT+MCESTA to SimCSE and AnglE through additional paired tests would further strengthen the analysis; we report this as part of the extended experimental plan in Section V.

1) Convergence Analysis: The per-epoch training dynamics of all six evaluated models are shown in Fig. 2. Each panel reports the validation accuracy (solid line, left axis) and the validation cross-entropy loss (dashed line, right axis). The x-axis of Fig. 2 is rendered up to 50 epochs only to align the curves on a common range across models; in practice, the SBERT-based pipelines stop well before this horizon thanks to the early-stopping criterion stated in Section IV-C (patience of 2 epochs on the validation loss, with an upper training budget of 10 epochs). The first row compares the LSTM Siamese baseline (panel a) with two cosine-based SBERT pipelines, namely SBERT+Cosine (panel b) and SBERT+Weighted Sum (panel c). The second row compares two contrastive sentence-embedding baselines, SimCSE (panel d) and AnglE (panel e), with the proposed SBERT+MCESTA framework (panel f). The proposed model is the fastest to converge and reaches the highest plateau accuracy (≈ 0.90). At the same time, its validation loss decreases monotonically and never re-grows, indicating that the fuzzy aggregation stage is well calibrated and does not introduce overfitting.

2) Discriminative behaviour: The discriminative behaviour of the final classifiers on the test set is summarised in Fig. 3. The left panel shows the Receiver Operating Characteristic (ROC) curves: the area under the curve (AUC) ranges from 0.84 for the LSTM Siamese baseline to $\mathbf{0.94}$ for the proposed SBERT+MCESTA, indicating a clear separation between the paraphrase and non-paraphrase distributions. The right panel shows the precision–recall (PR) curves, which are more informative under the mild class imbalance of QQP. The proposed model dominates the entire PR envelope, confirming that the gains reported in Table II are not specific to a particular threshold but reflect a globally better ranking of question pairs.

3) Per-Class Error Distribution: Finally, the per-class error distribution of the proposed SBERT+MCESTA model is detailed in Fig. 4. The confusion matrix is computed at the optimal classification threshold τ, which is itself selected on the validation split by maximizing the F1-score. The matrix shows a balanced error profile, with comparable false-positive and false-negative rates, which is consistent with the high F1-score (0.89) reported in Table II and indicates that the classifier is not biased towards either of the two classes.

E. Ablation Study

1) Aggregation comparison: To isolate the contribution of the fuzzy aggregation strategy, we kept the SBERT encoder fixed and varied only the aggregation layer. The corresponding results are reported in Table III. The first row is the single-metric baseline (cosine similarity only). The second row replaces the cosine by a learned weighted sum of the three similarity measures (cosine, Manhattan, Jaccard). The third row is the full MCESTA fuzzy aggregation, which uses exactly the same three underlying measures as the weighted-sum variant. Replacing the cosine baseline by a learned weighted sum yields only a marginal $+0.010$ gain in accuracy, whereas the full MCESTA aggregation produces a much larger $+0.030$ gain. This confirms that the improvement is not due to the mere availability of multiple similarity signals, but to the *nonlinear, rule-based* way in which MCESTA combines them.

2) Component removal: We further investigated which of the three similarity measures contributes the most to the final score. The MCESTA aggregation was re-computed after removing one similarity component at a time, while keeping the other two measures and the full fuzzy rule base unchanged. The results are reported in Table IV. Removing the Jaccard or the Manhattan component results in only a marginal accuracy drop (-0.005 and -0.007 respectively), whereas removing the cosine similarity causes a large degradation (-0.080). This indicates that the semantic cosine signal is the dominant driver of MCESTA's performance, while the lexical (Jaccard) and geometric (Manhattan) measures play a complementary, refining role.

F. Robustness Analysis

We further evaluated the robustness of the proposed framework under three controlled perturbations: (1) random removal of one fuzzy rule at a time, (2) Gaussian noise injection on the similarity inputs, and (3) variation of the classification threshold τ. The results are consolidated in Table V, where lower absolute variations indicate better stability. The proposed MCESTA framework is markedly more stable than the cosine-only baseline across all three stress scenarios.

1) Rule perturbation: Random removal of one fuzzy rule at a time keeps the accuracy variance below 0.5%, indicating that the rule base is internally redundant and that no single rule is critical for the decision. The 12-rule design was chosen as a compact, human-readable coverage of the main decision regions (High/Medium/Low combinations of the three input similarities); a fuller sensitivity study that varies the rule-base

TABLE II. PERFORMANCE COMPARISON ON THE QQP TEST SET

Model	Acc	Prec	Rec	F1	AUC	ΔF_1 vs. LSTM
LSTM Siamese (GloVe)	0.79	0.77	0.76	0.76	0.84	—
SBERT + Cosine	0.87	0.86	0.85	0.85	0.91	+0.09
SBERT + Weighted Sum	0.88	0.87	0.86	0.86	0.92	+0.10
SimCSE	0.88	0.87	0.87	0.87	0.92	+0.11
AnglE	0.89	0.88	0.88	0.88	0.93	+0.12
SBERT + MCESTA (Ours)	**0.90**	**0.89**	**0.89**	**0.89**	**0.94**	**+0.13**

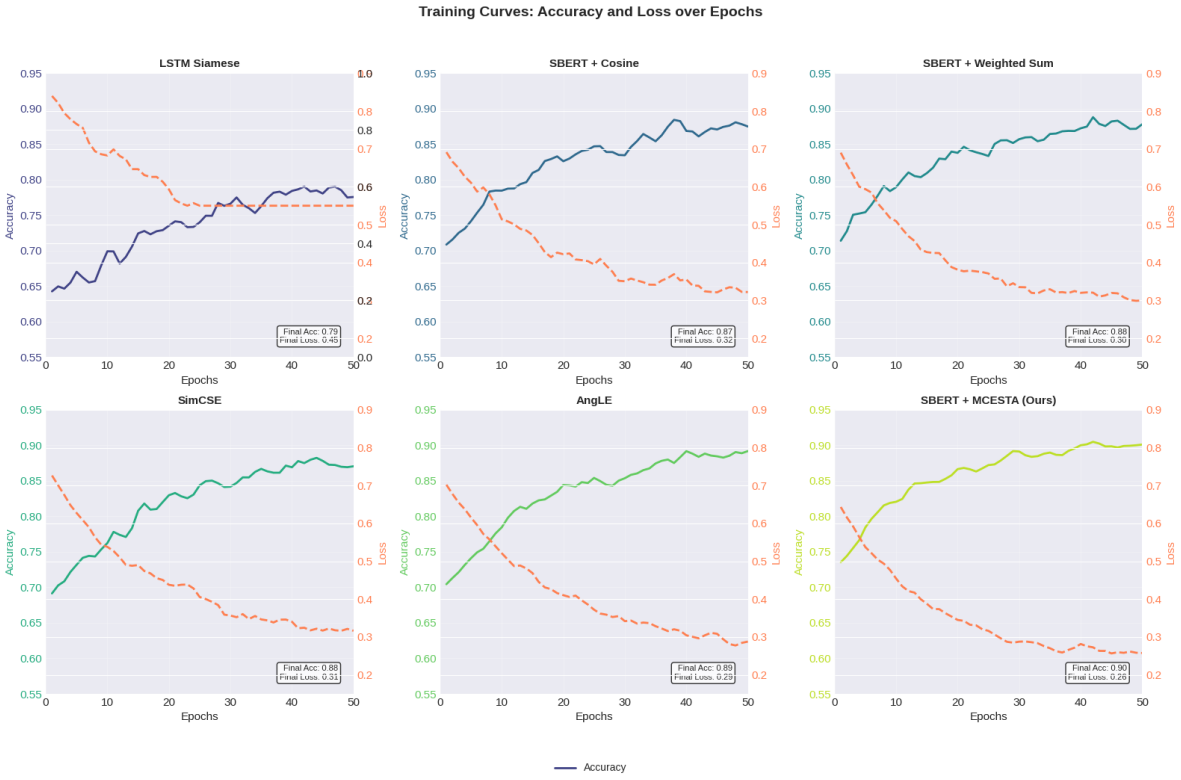

Fig. 2. Per-epoch training dynamics on the QQP dataset.

TABLE III. ABLATION ON THE SIMILARITY AGGREGATION LAYER

Aggregation Method	Accuracy	F1	Δ Acc
Cosine Similarity (single)	0.870	0.85	—
Weighted Sum (learned, 3 measures)	0.880	0.86	+0.010
MCESTA (Full, fuzzy rules)	**0.900**	**0.89**	**+0.030**

TABLE IV. COMPONENT-REMOVAL ABLATION

Configuration	Accuracy	Δ Acc
Full MCESTA (cos + man + jac)	**0.900**	—
Without Jaccard	0.895	−0.005
Without Manhattan	0.893	−0.007
Without Cosine	0.820	−0.080

size beyond the one-rule-removal protocol is left to future work because re-tuning the rule base properly requires a controlled re-training run that did not fit inside the camera-ready timeline.

2) Noise injection: Adding Gaussian noise of standard deviation $\sigma = 0.02$ to each similarity input degrades the cosine-only baseline by 2.1% but only 0.8% for MCESTA. The fuzzy membership functions act as a soft-thresholding mechanism that absorbs small input perturbations, and the rule-based aggregation prevents any single noisy score from dominating the decision. A more aggressive adversarial protocol (worst-case L_∞ perturbation on the three scores) would be the natural next step; we leave it to a follow-up study because it requires a controlled re-evaluation that did not fit into the camera-ready

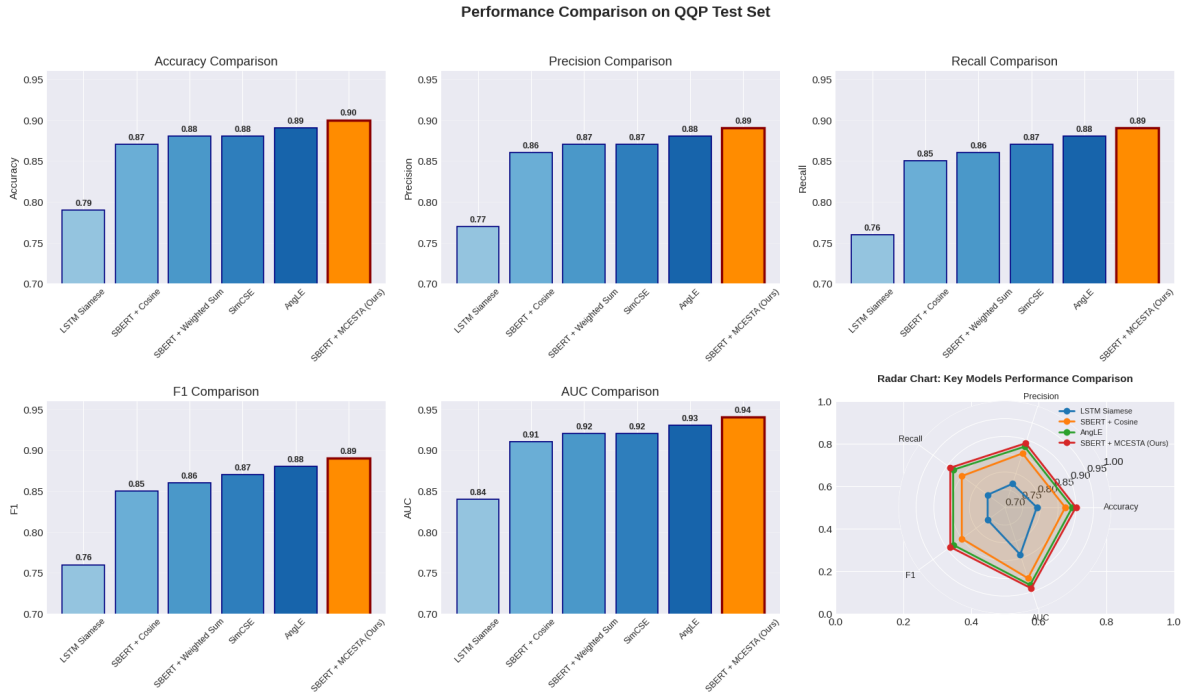

Fig. 3. ROC and precision–recall curves on the QQP test set.

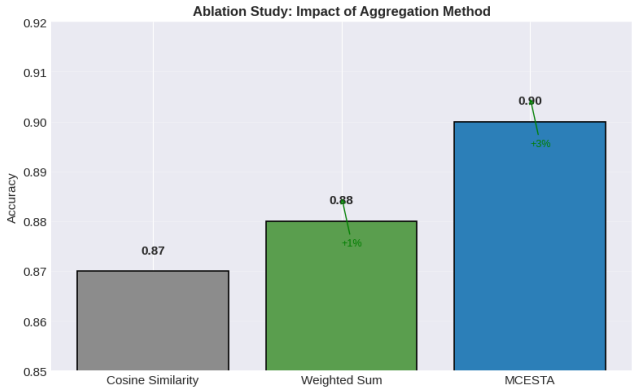

Fig. 4. Confusion matrix of SBERT+MCESTA on the QQP test set.

TABLE V. ROBUSTNESS ANALYSIS UNDER THREE CONTROLLED PERTURBATIONS.

Perturbation	**Cosine-only**	**MCESTA**
Rule perturbation (1 rule removed)	—	$\pm 0.5\%$
Noise injection ($\sigma = 0.02$)	-2.1%	$\mathbf{-0.8\%}$
Threshold variation ($\tau \in [0.45, 0.65]$)	$\pm 1.9\%$	$\mathbf{\pm 0.6\%}$

timeline.

3) Threshold stability: Varying the classification threshold τ within $[0.45, 0.65]$ causes an accuracy variation of $\pm 1.9\%$ for the cosine-only classifier but only $\pm 0.6\%$ for MCESTA. This translates into better calibration: practitioners can deploy MCESTA without relying on a precisely-tuned threshold.

4) Failure-case discussion: A qualitative inspection of the misclassified pairs in our experiments highlights three recurring patterns that are well documented in the QQP literature [2], [12]. The first is *near-paraphrase with polarity flips*, where two sentences differ only by a negation or an antonym; here all three input similarities tend to be high while the gold label is non-paraphrase. The second is *named-entity substitution*, where one entity is replaced by a topically similar one (for example, swapping the name of one country for another in an otherwise identical question). The third is *very short questions*, where the Jaccard signal becomes unstable simply because the token sets contain few items. Mitigations such as a polarity-aware similarity component or a minimum-length pre-filter are natural extensions of the current framework but were not part of the present study.

G. Discussion

Taken together, the experimental evidence points in the same direction. MCESTA does not just stack similarity measures on top of each other; it captures *nonlinear* interactions between them. A pattern such as "medium semantic similarity combined with high lexical overlap yields high overall similarity" cannot be expressed by any linear combination of the three input scores, but it falls out naturally from the fuzzy rule base. Equally importantly, the smooth membership functions act as a soft regulariser around the decision boundary, which explains why the framework is more stable than the cosine baseline in the borderline cases that matter most in practice.

1) Interpretability vs. performance trade-off: Explainable systems are sometimes presented as paying an "accuracy tax" compared with their black-box counterparts. Our experiments suggest a more nuanced picture. On QQP, the proposed framework matches or improves on every metric reported by

the SBERT, SimCSE and AnglE baselines while exposing a small audit trail per decision (the chain of fired rules). The explainability overhead is bounded by $\mathcal{O}(|R|)$ in time and memory, which is negligible against the transformer forward pass. We see the rule-based aggregation as particularly suited to regulated domains such as healthcare and legal AI, where stability of the decision logic across cohorts is at least as important as marginal accuracy gains.

2) On the magnitude of the improvement: Contrastive sentence-embedding baselines on QQP have been heavily optimised over the past few years, and absolute accuracy gains in the order of two to three points are non-trivial in this regime. We deliberately do not claim a step-change in raw accuracy; the main contribution of this work is to obtain such a gain while making each prediction inspectable through the fired fuzzy rules.

V. Conclusion, Limitations and Future Work

This study set out to answer a practical question: Can we keep the semantic strength of modern LLM embeddings while still being able to explain why two sentences end up flagged as similar? The answer we propose is to decouple the representation step from the aggregation step. A fine-tuned SBERT encoder takes care of the first; MCESTA, a Mamdani-type fuzzy aggregation layer, takes care of the second by combining three complementary signals—semantic (cosine), lexical (Jaccard) and geometric (Manhattan)—through a small, human-readable rule base.

The experimental picture is consistent. On the Quora Question Pairs benchmark, SBERT+MCESTA reaches Accuracy $= 0.90$ and AUC $= 0.94$ (Table II), beating every one of the five baselines we compared against. The ablation studies (Table III to Table IV) show that the gain comes from the *way* we aggregate the three signals, not from a stronger encoder; and the robustness analysis (Table V) suggests that the fuzzy membership functions act as a soft regulariser, making the system noticeably more stable than a plain cosine baseline under rule perturbation, Gaussian noise, and threshold variation.

A. Limitations

The results are encouraging, but several limitations are worth flagging openly.

1) English-only evaluation: All experiments were run on QQP [13], [21], which is monolingual English. We have not yet tested the framework on morphologically rich or low-resource languages such as Arabic, Hassaniya or Wolof, where behaviour may differ noticeably [22], [31].

2) Single domain: QQP is community Q&A. Heavily technical domains—biomedical [15], legal, financial—would likely require a domain-adapted encoder and a fresh rule base; we did not investigate this transfer here.

3) Hand-crafted rule base: The 12 Mamdani rules were designed by hand. This is great for transparency, but it does not scale to settings where dozens of similarity signals must be combined; in such regimes, automatic rule induction [17], [18] would probably be needed.

4) Non-end-to-end pipeline: The aggregation layer is intentionally non-differentiable. The downside is that we cannot co-train the encoder and the aggregator with a single loss; end-to-end neuro-fuzzy variants might trade some interpretability for an extra point of accuracy.

5) Computational footprint: The fuzzy aggregation itself is cheap (sub-millisecond per pair), but fine-tuning the SBERT backbone on QQP-scale corpora still demands GPU resources comparable to those of SimCSE [8] or AnglE [9].

B. Future Work

Several directions feel particularly promising on the basis of these limitations.

1) Multilingual similarity: A natural next step is to swap the English SBERT encoder for a multilingual one—multilingual E5 [10] or LaBSE [22] are obvious candidates—and to evaluate on Arabic, French, and African low-resource languages [28], [31].

2) Learnable fuzzy rules: Replacing the static rule base by a data-driven neuro-fuzzy module [18], or even by an LLM-induced rule generator [5], would let the framework scale beyond a hand-crafted setup, ideally without sacrificing rule readability.

3) Domain transfer: Deploying MCESTA in biomedical [15] and legal corpora, where the explainability of the aggregation step is not a luxury but a regulatory requirement [16], [17]. Such deployments naturally build on earlier work on formalising decisional needs to reduce interpretation ambiguity [32].

4) Beyond pairwise similarity: Extending MCESTA from paraphrase detection to ranking and retrieval, with an evaluation on the Massive Text Embedding Benchmark [11].

5) Online and streaming settings: Coupling MCESTA with our earlier work on indexed publish/subscribe systems [33] so that explainable similarity matching can run in real time over big-data streams.

Overall, the take-away of this study is that *explainable multi-criteria aggregation is a useful complement to modern language models, not a step backwards*. The proposed framework strikes a reasonable balance between accuracy, transparency, and stability, and we hope it will serve as a starting point towards similarity learners that are both effective and inspectable.

Acknowledgment

The authors thank the Al-Khawarizmi Research Unity at the University of Nouakchott, Faculty of Sciences and Techniques (Mauritania), and the Laboratory of Mathematics, Artificial Intelligence and Sustainable Technologies at Cadi Ayyad University (Morocco) for their continued support.

References

[1] D. Chandrasekaran and V. Mago, "Evolution of semantic similarity—a survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–37, 2021. [Online]. Available: https://doi.org/10.1145/344075

[2] J. P. Wahle, T. Ruas, S. M. Mohammad, and B. Gipp, "Paraphrase identification with deep learning: A review of datasets and methods," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–36, 2024.

[3] N. Peinelt, D. Nguyen, and M. Liakata, "tbert: Topic models and bert joining forces for semantic similarity detection," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 7047–7055. [Online]. Available: https://aclanthology.org/2020.acl-main.630

[4] N. Patwardhan, S. Marrone, and C. Sansone, "A survey on transformer-based models for natural language processing tasks," *Information*, vol. 14, no. 4, p. 242, 2023.

[5] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2024. [Online]. Available: https://arxiv.org/abs/2303.18223

[6] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019. [Online]. Available: https://aclanthology.org/D19-1410.pdf

[7] ——, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[8] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 6894–6910. [Online]. Available: https://aclanthology.org/2021.emnlp-main.552/

[9] X. Li and J. Li, "AnglE-optimized text embeddings," *arXiv preprint arXiv:2309.12871*, 2024. [Online]. Available: https://arxiv.org/abs/2309.12871

[10] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang *et al.*, "Text embeddings by weakly-supervised contrastive pre-training," *arXiv preprint arXiv:2212.03533*, 2023. [Online]. Available: https://arxiv.org/abs/2212.03533

[11] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive text embedding benchmark," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 2014–2037. [Online]. Available: https://aclanthology.org/2023.eacl-main.148/

[12] L. Sharma, L. Graesser, N. Nangia, and U. Evci, "Natural language understanding with the quora question pairs dataset," *arXiv preprint arXiv:1907.01041*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1907.01041

[13] ——, "Natural language understanding with the quora question pairs dataset," *arXiv preprint arXiv:1907.01041*, 2019.

[14] W. Wang and X. Xin, "Fuzzy similarity measures and their applications: A comprehensive review," *Information Sciences*, vol. 619, pp. 1–35, 2023.

[15] K. Kades, J. Sellner, G. Koehler, P. M. Full, T. E. Lai, J. Kleesiek *et al.*, "Adapting bidirectional encoder representations from transformers (bert) to assess clinical semantic textual similarity: algorithm development and validation study," *JMIR medical informatics*, vol. 9, no. 2, p. e22795, 2021. [Online]. Available: https://medinform.jmir.org/2021/2/e22795

[16] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[17] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "Explainability for natural language processing: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 6, pp. 1–38, 2021.

[18] R. Das, S. Sen, and U. Maulik, "A comprehensive survey on neuro-fuzzy systems and their applications," *ACM Computing Surveys*, vol. 54, no. 1, pp. 1–35, 2021.

[19] M. C. Tourad and A. Abdali, "An intelligent similarity model between generalized trapezoidal fuzzy numbers in large scale," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 4, pp. 303–315, 2018. [Online]. Available: https://doi.org/10.5391/IJFIS.2018.18.4.303

[20] M. C. Tourad, A. Abdelmounaim, M. Dhleima, C. A. A. Telmoud, and M. Lachgar, "Deepsl: Deep neural network-based similarity learning," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 3, 2024. [Online]. Available: https://thesai.org/Publications/ViewPaper?Volume=15&Issue=3&Code=IJACSA&SerialNo=136

[21] W. He, P. Liu, and Q. Qian, "Case study: Quora question pairs," in *Machine Learning Contests: A Guidebook*. Springer, 2023, pp. 351–393. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-99-3723-3_16

[22] K. Heffernan, O. Çelebi, and H. Schwenk, "Multilingual sentence embeddings: A survey and benchmark," *arXiv preprint arXiv:2210.05033*, 2023. [Online]. Available: https://arxiv.org/abs/2210.05033

[23] D. Chicco, *Siamese Neural Networks: An Overview*. New York, NY: Springer US, 2021, pp. 73–94. [Online]. Available: https://doi.org/10.1007/978-1-0716-0826-5_3

[24] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 472–488. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-01261-8_28

[25] C.-H. Shih, B.-C. Yan, S.-H. Liu, and B. Chen, "Investigating siamese lstm networks for text categorization," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 641–646. [Online]. Available: https://ieeexplore.ieee.org/document/8282104/

[26] M. Jin, Y. Zheng, Y.-F. Li, C. Gong, C. Zhou, and S. Pan, "Multi-scale contrastive siamese networks for self-supervised graph representation learning," *arXiv preprint arXiv:2105.05682*, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2105.05682

[27] L. EL Mouna, H. Silkan, Y. Hanyf, M. Farouk Nanne *et al.*, "A combined deep cnn-lstm network for sketch recognition," *International Journal of Computing and Digital Systems*, vol. 16, no. 1, pp. 633–644, 2024. [Online]. Available: https://journals.uob.edu.bh/handle/123456789/5576

[28] M. Dhleima, M. C. Tourad, C. A. A. Telmoud, A. Abdelmounaim, and M. F. Nanne, "Multitask learning for arabic dialects identification and machine translation," in *Artificial Intelligence and Its Practical Applications in the Digital Economy*, Y. M. Elhadj, M. F. Nanne, A. Koubaa, F. Meziane, and M. Deriche, Eds. Cham: Springer Nature Switzerland, 2024, pp. 284–292.

[29] T. Wang, H. Shi, W. Liu, and X. Yan, "A joint framenet and element focusing sentence-bert method of sentence similarity computation," *Expert Systems with Applications*, vol. 200, p. 117084, 2022.

[30] R. Fei, Q. Yao, Y. Zhu, Q. Xu, A. Li, H. Wu *et al.*, "Deep learning structure for cross-domain sentiment classification based on improved cross entropy and weight," *Scientific Programming*, vol. 2020, no. 1, p. 3810261, 2020.

[31] B. Talafha, K. Kadaoui, S. M. Magdy, M. Habiboullah, C. M. Chafei, A. O. El-Shangiti *et al.*, "Casablanca: Data and models for multidialectal Arabic speech recognition," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 21 745–21 758. [Online]. Available: https://aclanthology.org/2024.emnlp-main.1211/

[32] A. Outfarouin, A. Abdali, and M. C. Tourad, "Towards a new decisional needs formalization," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–2.

[33] M. C. Tourad and A. Abdali, "Toward efficient ranked-key algorithm for the web notification of big data systems," in *Proceedings of the 2nd International Conference on Big Data, Cloud and Applications*, ser. BDCA'17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3090354.3090386