Ingénierie des Systèmes d'Information
Vol. 31, No. 10, October, 2026, pp. 3321-3330
Journal homepage: http://iieta.org/journals/isi

# Comparing Lexical and Sentence-Embedding Methods for Mapping User-Behaviour Indicators to Non-Functional Requirement Categories

Sali Alas Majapahit[1,2*], Purwanto[3], Farikhin[4]

[1] Doctoral Program of Information Systems, Postgraduate School, Diponegoro University, Semarang 50421, Indonesia
[2] Department of Informatics Engineering, Faculty of Engineering, Pasundan University, Bandung 40153, Indonesia
[3] Department of Chemical Engineering, Faculty of Engineering, Diponegoro University, Semarang 50275, Indonesia
[4] Department of Mathematics, Faculty of Science and Mathematics, Diponegoro University, Semarang 50275, Indonesia

Corresponding Author Email: salialasm@students.undip.ac.id

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.311007

**Received:** 8 June 2026
**Revised:** 26 August 2026
**Accepted:** 15 September 2026
**Available online:** 31 October 2026

***Keywords:***
*semantic matching, non-functional requirements, user-behavior indicators, sentence embeddings, Masked and Permuted Pre-training for Language Understanding, requirements engineering*

## ABSTRACT

Mapping user-behavior indicators to non-functional requirements (NFRs) requires methods that can identify semantic relations beyond shared terminology. This study compares four matching approaches—Exact, Synonym, Conceptual, and Contextual Matching, and Masked and Permuted Pre-training for Language Understanding (MPNet)-based contextual matching—for aligning user-behavior indicators with three NFR categories defined in ISO/IEC 25023:2016: performance efficiency, security, and usability. The indicator set was informed by a survey of 267 Indonesian industry professionals. In addition, five experts assessed a stratified sample of 86 indicator pairs to examine the plausibility of similarity assignments. The four approaches were implemented in Python using lexical resources and pretrained sentence-transformer models. Contextual matching produced the highest mean similarity scores for performance efficiency (0.609), security (0.593), and usability (0.642). In paired comparisons with conceptual matching, the differences were statistically significant in all three categories ($p < 0.001$), with reported effect sizes ranging from Cohen's $d = 0.76$ to 1.10. These results indicate that MPNet-based embeddings assign higher similarity scores than the selected MiniLM-based approach on the evaluated indicator pairs. However, similarity scores alone do not establish correct requirement matches, and expert review covered only 86 pairs. The findings should therefore be interpreted as evidence about relative score behavior within the selected categories, rather than as proof of general matching accuracy. Further evaluation against a larger, independently annotated benchmark is needed before the approach can be recommended for operational requirements engineering.

## 1. INTRODUCTION

Developing an information system requires careful specification of both functional requirements and non-functional requirements (NFRs). NFRs define system quality standards — performance, security, usability — that strongly determine a system's ultimate success [1, 2]. User behavior is a significant driver of NFR fulfilment; measuring usability solely by interface dimensions is insufficient without accounting for real-world user experience in context [3-5].

A suitable knowledge representation model is needed to manage the complexity of specifying and analyzing NFRs based on user behavior. Ontologies are well suited to define and link entities, supporting reasoning, visualization, and detection of semantic relationships [6]. However, no ontology currently provides an explicit, verified mapping between user behavior characteristics and NFR structures. The terminological gap between these domains underscores the need for robust semantic matching methods [7].

Ontology development employs several semantic matching approaches — Exact, Synonym, Conceptual, and Contextual Matching — each with distinct strengths and limitations [8]. Previous research has applied these techniques primarily to descriptive models without comparative statistical analysis, limiting their scientific validity and practical applicability.

While Exact, Synonym, Conceptual, and Contextual Matching are individually established techniques in natural language processing (NLP) and ontology matching research, their systematic, statistically validated comparison specifically for aligning user behavior indicators with NFR categories has not previously been reported; this comparison also serves as the empirical basis for selecting the semantic matching approach adopted in the broader cross-domain ontology this research program is building. To address these gaps, this study implements and compares all four techniques using a Python framework incorporating spaCy, Natural Language Toolkit (NLTK), and the Masked and Permuted Pre-training for Language Understanding (MPNet) transformer model. Validation uses 267 Indonesian industrial professionals and a five-expert panel rating 86 stratified indicator pairs. Three main contributions are made: (1) a reusable Python evaluation framework; (2) empirical evidence that contextual embedding

outperforms literal and synonym-based methods; and (3) a labeled benchmark dataset.

## 2. RELATED WORK

Matching is a foundational method in ontology engineering [9]. However, no study has systematically compared Exact, Synonym, Conceptual, and Contextual matching techniques for linking user behavior indicators with NFRs specifically.

Studies on NFRs frequently employ ontology approaches to model quality characteristics. Studies [7, 8, 10, 11] constructed ontology models for quality requirements related to the user trust dimension. However, these studies are limited to the trust dimension alone and do not include comparative analysis of semantic matching techniques. The Bidirectional Encoder Representations from Transformers (BERT)-based NFR classification approach represents another avenue of research that has not yet been integrated with user behavior mapping [12, 13].

Research combining term-based, meaning-based (WordNet, cosine similarity), and structure-based methods shows that hybrid approaches can improve ontology alignment accuracy [14]. Graph-based approaches, including Graph Attention Networks (GraphMatcher [15]), capture contextual relationships more deeply. Recent large language model (LLM)-based frameworks such as Agent OM [16], LLMs4OM [17], and pretrained language models [18] show promising results but have not been adapted specifically for NFR-behavior mapping. Semantic matching has also been applied outside ontology alignment, for instance in e-government information resource integration [19], and also in the field of biomedicine [20], though not in the context of NFR–behavior mapping.

In summary, prior research exhibits three persistent shortcomings: (1) insufficient integration between user behavior and software quality requirements; (2) absence of consistent evaluation methods; and (3) inadequate statistical validation. This study addresses all three.

Table 1 summarizes how this study's approach compares with representative prior work in ontology and semantic matching.

**Table 1.** Comparison of representative prior semantic-matching and ontology-alignment studies

| Study | Approach | Domain / Dataset | Evaluation Method | Limitation (Relative to This Study) |
|---|---|---|---|---|
| Olsinas et al. [7] | Ontology of NFR concepts | Measurement/evaluation projects (trust dimension) | Conceptual/case analysis | No comparative semantic-matching evaluation; single dimension |
| Dobson et al. [8] | Domain-independent NFR ontology | General software NFRs | Descriptive/conceptual | No empirical semantic-similarity evaluation |
| Amaral et al. [11] | Ontology-based trustworthiness requirements modeling | Trustworthiness requirements | Preliminary case study | Limited to trust dimension; no statistical validation |
| Kaur and Kaur [12] | MNoR-BERT (BERT-based NFR classification) | Multi-label NFR classification | Classification accuracy metrics | Not integrated with user-behavior mapping |
| Lu et al. [14] | Hybrid term/meaning/structure-based ontology meta-matching | General ontology alignment benchmarks | Similarity-measure comparison | Not applied to the NFR-behavior domain |
| Efeoglu [15] | GraphMatcher (Graph Attention Network) | Ontology matching benchmarks (OAEI) | Graph-based alignment accuracy | Domain-general; no NFR-specific adaptation |
| Qiang et al. [16] | Agent OM (LLM agents for ontology matching) | Ontology matching benchmarks | LLM-agent alignment accuracy | Not adapted for NFR-behavior mapping |
| Babaei Giglou et al. [17] | LLMs4OM | Ontology matching benchmarks | LLM-based alignment accuracy | Not adapted for NFR-behavior mapping |
| Huang et al. [19] | Semantic matching for e-government resource fusion | E-government information resources | Domain-specific matching accuracy | Different application domain; no NFR focus |
| Wu et al. [20] | DAEOM (deep attentional embedding) | Biomedical ontology matching | Embedding-based alignment accuracy | Biomedical domain; no NFR-behavior focus |
| This study | Exact / Synonym / Conceptual / Contextual semantic matching comparison | User behavior × NFR (ISO/IEC 25023:2016) indicators | Paired *t*-test, Wilcoxon signed-rank, five-expert panel validation (Cohen's κ, ICC, Cronbach's $\alpha$) | — |

Note: NFR = non-functional requirement, LLM = large language model, BERT = Bidirectional Encoder Representations from Transformers, ICC = intraclass correlation coefficient, OAEI = Ontology Alignment Evaluation Initiative, DAEOM = deep attentional embedding approach for biomedical ontology matching.

## 3. PROPOSED METHOD

### 3.1 Research design

The research design comprises five sequential stages, illustrated in Figure 1: (1) collection of user behavior and NFR indicators; (2) preprocessing and standardization; (3) semantic matching using four techniques; (4) statistical evaluation and error analysis; and (5) visualization and interpretation.

### 3.2 Data sources

3.2.1 User behavior indicators

The user behavior dimensions and their corresponding indicators were derived from prior studies [3, 21, 22].

The behavioral indicators were formulated based on nine user behavior dimensions established by Alas and Rijayanti [23]: Trust and Security, Ease of Use, Perceived Usefulness, External Support, User Habits, Technology Readiness, User Innovativeness, Perceived Risk, and Socio-cultural Factors.

These nine dimensions were validated through a field survey involving 267 industry respondents in Indonesia (out of a total of 472 respondents, following quality screening) [23].

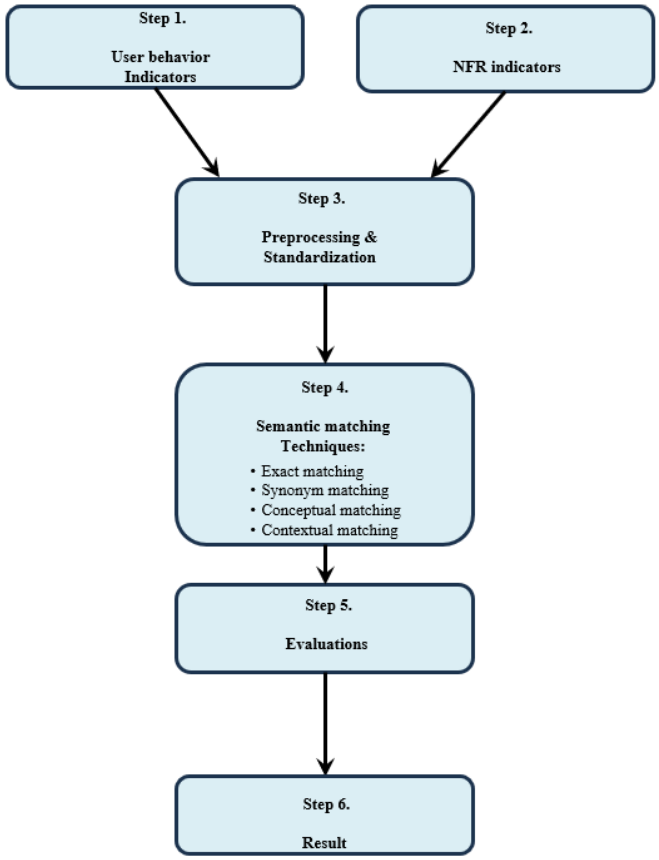


**Figure 1.** Stages of the research process

Participants were recruited using non-probability, purposive sampling, since the exact size of the target population (Indonesian industry employees using information technology in their work) was unknown. Eligibility required that respondents be employees in Indonesian industry who used information technology for at least five hours per working day. The minimum required sample size was determined using Cochran's formula [24], yielding a minimum of 97 respondents. Primary data were collected through an online questionnaire (Google Forms) using a five-point Likert scale, distributed through personal networks, business associations, government associations, and corporate groups to reach respondents across a wide geographic area. Of 472 responses received, 267 were retained as valid after data-quality screening. Respondents represented a range of organizational roles (managers, staff, and operators), education levels (ranging from senior high school equivalent to doctoral degree), and tenures, with the majority holding staff positions and a bachelor's degree.

3.2.2 Non-functional requirement indicators

The NFR dimensions consist of indicators derived from the three NFR categories defined in ISO/IEC 25023:2016 [25]. Three types of NFRs are operationalized—Performance Efficiency, Usability, and Security—each of which is further broken down into a series of specific quality indicators [10, 25]. These three categories were selected as the initial scope for this study because they are consistently identified as the most frequently discussed and empirically investigated NFR types in the software requirements literature [26], making them the most representative starting point for cross-domain alignment with user behavior before this work is extended to the remaining categories of the ISO/IEC 25023:2016 taxonomy.

3.2.3 Keyword assignment

Each user-behavior and NFR indicator was assigned one or more representative keywords used as input to the four matching techniques. Keywords were derived through a structured, semi-automated procedure guided by the following steps:

•Term extraction: the core term was extracted from each indicator's name, and the indicator's definition (per ISO/IEC 25023:2016 for NFR indicators, or the source behavioral studies [21-23] for user-behavior indicators) was parsed to identify the object assessed (e.g., data, access, identity, log), the associated activity or process (e.g., encryption, authentication, retention), and the relevant quality attribute or goal (e.g., confidentiality, integrity, non-repudiation).

•Synonym and variant expansion: established technical synonyms and terminological variants from the literature were added, together with terms denoting non-conformance, prevention, verification, or compliance where supported by the indicator definition.

•Semantic-traceability filtering: every candidate keyword was required to be traceable back to the indicator's name or definition, duplicates and overly generic terms were removed, and the final list was normalized (lower-cased, with multi-word technical phrases preserved). This procedure was applied uniformly across all three NFR categories and the nine user-behavior dimensions. Keyword extraction was artificial intelligence (AI)-assisted for term consolidation and reviewed manually by the first author for consistency and traceability, but was not independently validated by external domain experts — a limitation discussed in Section 5.

To support reproducibility, an anonymized version of the 267-response survey dataset is provided via the repository referenced in the Data Availability Statement. Table 2 summarizes the number of indicators for each dimension.

**Table 2.** Indicator and keyword count by dimension

| Dimension | Indicators | Keywords |
|---|---|---|
| User behavior | 36 | 25 |
| Performance efficiency | 12 | 36 |
| Usability | 22 | 68 |
| Security | 11 | 33 |

### 3.3 Preprocessing

To enable computational analysis, preprocessing phase was required for the dataset. The objective is to simplify textual representation and mitigate potential linguistic bias. This process involved the following:

•The overall lexical items were converted to lowercase (lowercase normalization)

•Elimination of duplicate terms within each entity

•Elimination of special characters and punctuation marks

•Tokenization and structure normalization

The output from this process was stored in column "Preprocessed Description," which functioned as an input across all matching methods. The entire process was executed by Python, such as pandas, re, string, and NLTK.

### 3.4 Semantic matching techniques

Building on the preprocessed dataset described in Section 3.3, four semantic matching techniques were applied to compute similarity scores between user behavior indicators and NFR sub-indicators, as detailed below.

3.4.1 Exact Matching

Jaccard Similarity over binary bag-of-words:

$$\text{ExactMatch}(A, B) = \frac{|\text{Tokens}(A) \cap \text{Tokens}(B)|}{|\text{Tokens}(A) \cup \text{Tokens}(B)|} \quad (1)$$

The implementation utilized CountVectorizer and the Jaccard score from scikit-learn.

3.4.2 Synonym Matching

In practice, the study employed this approach to perform lexical matching utilizing synonym equivalence and WordNet from the NLTK library to align items in set A with corresponding counterparts in set B. This idea was inspired by semantic-aware ontology matching approaches similar to the proposed approaches in the studies [14, 23].

$$\text{SynonymMatch}(A, B) = \frac{|\text{Synonym}(A) \cap \text{Tokens}(B)|}{|\text{Tokens}(A)|} \quad (2)$$

Implementation: nltk.corpus.wordnet, nltk.tokenize.

3.4.3 Conceptual Matching

Conceptual Matching computes semantic similarity using Sentence-BERT (SBERT) with the pretrained all-MiniLM-L6-v2 model [27], a distilled sentence-transformer that encodes each indicator into a single pooled embedding vector representing its overall meaning. Cosine similarity between the pooled embeddings of indicator pairs is then calculated using Eq. (3). This approach is inspired by semantic vector-based ontology matching techniques described in studies [14] and [23]. Because all-MiniLM-L6-v2 produces a generalized, coarse-grained sentence representation, Conceptual Matching captures broad topical similarity but does not explicitly encode word order or fine-grained grammatical structure.

$$\text{cosine_sin}(e_a, e_b) = \frac{e_a . e_b}{\|e_a\| . \|e_b\|} \quad (3)$$

Implementation involves using sentence transformers and util. cos_sim.

3.4.4 Contextual Matching

Contextual Matching uses a larger, more expressive pretrained model, all-mpnet-base-v2 [29], selected for its consistently strong performance among pretrained sentence-transformer models in prior comparative benchmarks [29, 30]. Unlike all-MiniLM-L6-v2, all-mpnet-base-v2 is pretrained with a Permutation Language Modeling (PLM) objective, which encodes word order and grammatical role directly into its learned representations—a property embedded in the model's pretrained weights, not a setting toggled at inference time. Sentence embeddings are L2-normalized (normalize_embeddings=True) before computing cosine similarity, a standard practice for stabilizing similarity comparisons that is independent of the model's word-order sensitivity. Because of this pretraining objective, Contextual Matching treats indicators as ordered linguistic structures, disambiguating polysemous terms based on grammatical role and surrounding context. For example, 'system response time' and 'time for system response'—which share identical tokens in a different order—yield nearly identical Conceptual Matching scores but measurably different Contextual Matching scores, illustrating the two models' differing sensitivity to word order. This performance advantage should not be attributed to word-order sensitivity alone: all-mpnet-base-v2 also differs from all-MiniLM-L6-v2 in underlying transformer architecture and pretraining strategy—the former is a full 12-layer transformer pretrained with a permutation-based objective, while the latter is a 6-layer model obtained through knowledge distillation—so part of the observed improvement may also reflect greater model capacity rather than word-order encoding alone.

### 3.5 Evaluation protocol

3.5.1 Similarity scoring and visualization

Similarity scores were computed for all user behavior–NFR pairs, and subsequently averaged per NFR category. The results were visualized using boxplots (matplotlib, seaborn) to compare score distributions, medians, and outliers.

3.5.2 Gold standard annotation

A stratified sample of 86 indicator pairs was drawn to represent four SBERT-assigned similarity-strength bands (Strong, $n = 6$; Medium, $n = 30$; Weak, $n = 30$; Not Relevant, $n = 20$). For each pair, a panel of five domain experts (four holding doctoral degrees and one a master's degree, with 10 to over 18 years of professional experience in information systems, software engineering, or human-computer interaction, and 79 relevant publications reported collectively) was shown the user-behavior indicator, the NFR indicator, and the SBERT-assigned similarity-strength label, and rated their agreement with that categorical assignment on a 5-point Likert scale (1 = strongly disagree to 5 = strongly agree), completed across two sessions of 43 pairs each. Because Shapiro–Wilk tests indicated non-normal rating distributions for all five experts ($p < 0.001$ in each case), inter-rater agreement was assessed using non-parametric and categorical statistics rather than Pearson correlation. Mean pairwise Cohen's kappa across the ten expert pairs was 0.517 (linear-weighted $\kappa = 0.529$), indicating moderate agreement (Landis and Koch criteria); Kendall's $W$ (tie-corrected) was 0.627 ($\chi^2(85) = 266.37$, $p < 0.001$), indicating strong and significant concordance across the panel, and internal consistency was good (Cronbach's $\alpha = 0.860$). Reliability of a single expert's rating was moderate (intraclass correlation coefficient (ICC)(2,1) = 0.548), whereas reliability of the panel's averaged rating was good (ICC(2,k) = 0.858); disagreements were therefore resolved by using the mean of the five ratings as the representative judgment for each pair. Under this aggregation, 77 of 86 pairs (89.5%) received majority expert agreement (mean rating $\geq 4$), with agreement highest for pairs SBERT classified as Strong (6/6, 100%) and Not Relevant (19/20, 95.0%), and comparatively lower for the Medium (25/30, 83.3%) and Weak (27/30, 90.0%) bands; a content validity ratio of at least 0.99 was reached for 70 of 86 pairs (81.4%), and no pair was rejected outright (mean rating < 3.0). These results support the appropriateness of the SBERT-based similarity categorization used in this study, while the moderate single-rater kappa indicates that individual expert judgments vary more on

intermediate (Medium/Weak) similarity distinctions than at the extremes, consistent with the inherent subjectivity of fine-grained semantic-similarity judgment.

3.5.3 Statistical testing

Pairwise comparisons between techniques were evaluated using paired *t*-tests ($\alpha = 0.05$). To complement this analysis, non-parametric Wilcoxon signed-rank tests were applied for robustness. Furthermore, confidence intervals (95%) were reported to assess effect sizes and variance.

3.5.4 Error analysis

For a more comprehensive evaluation, an error analysis was conducted to identify typical failure cases, including:

•Polysemy: terms such as "secure" were misinterpreted as safety vs. security.

•Domain shift: WordNet failed to cover NFR-specific terms.

•Long-range dependencies: SBERT missed clause-level cues, which MPNet handled with greater accuracy.

**Table 3.** Mixed maximum difference

| NFR Categories | (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|---|
| Performance Efficiency | 0.535792 | 0.608983 | 0.041512 | 0.015557 | 0.593426 |
| Security | 0.515344 | 0.592769 | 0.031884 | 0.017629 | 0.575140 |
| Usability | 0.580629 | 0.641537 | 0.048974 | 0.014306 | 0.627230 |

Notes: (1) Conceptual, (2) Contextual, (3) Exact, (4) Synonym, (5) Max-Min, NFR = non-functional requirement.

In addition, for each NFR category, the difference between the maximum and minimum similarity scores was calculated to quantify the effectiveness range of each technique, as presented in Table 3.

### 3.6 Computing environment and implementation

Ubuntu 22.04 LTS; Python 3.10.12; Intel Core i7-11800H @ 2.30 GHz; 32 GB DDR4 RAM; NVIDIA GeForce RTX 3060 6 GB (CUDA 11.7). Libraries: pandas 1.5.3, numpy 1.23.5, NLTK 3.8.1, scikit-learn 1.2.2, sentence-transformers 2.2.2, matplotlib 3.7.1, seaborn 0.12.2.

Details regarding the use of the library are presented in Table 4. Meanwhile, an example of using the sentence-transformer library is as follows:

(1) Conceptual Matching (pooled SBERT):

```
from sentence_transformers import SentenceTransformer, util
model_c = SentenceTransformer('all-MiniLM-L6-v2')
def conceptual_match(a, b):
    ea = model_c.encode(a, convert_to_tensor=True)
    eb = model_c.encode(b, convert_to_tensor=True)
    return float(util.cos_sim(ea, eb))
```

(2) Contextual Matching (PLM full-context):

```
model_x = SentenceTransformer('all-mpnet-base-v2')
def contextual_match(a, b):
    ea = model_x.encode(a, convert_to_tensor=True,
            normalize_embeddings=True)  # PLM activated
    eb = model_x.encode(b, convert_to_tensor=True,
            normalize_embeddings=True)
    return float(util.cos_sim(ea, eb))
```

**Table 4.** Python libraries and modules used in the study

| Steps | Matching Technique | Python Library/ Module | Functions / Transformers Used |
|---|---|---|---|
| Preprocessing | All Techniques | pandas, re, string | extract(), split(), token filtering |
| Exact Matching | Literal Token | sklearn.feature_extraction.text.CountVectorizer, sklearn.metrics.jaccard_score | Binary Bag-of-Words, Jaccard Similarity |
| Synonym Matching | Lexical Synonym | nltk.corpus.wordnet, nltk.tokenize | synsets(), lemmas(), intersection() |
| Conceptual Matching | Sentence + Concept | sentence-transformers | SentenceTransformer('all-MiniLM-L6-v2'), util.cos_sim() |
| Contextual Matching | Sentence + Concept | sentence-transformers | SentenceTransformer('all-mpnet-base-v2'), util.cos_sim() |
| Analysis and Summary | All Techniques | pandas, groupby(), melt() | sum(), mean(), pivot() |
| Visualization | All Techniques | matplotlib, seaborn | barplot(), tight_layout(), savefig() |

## 4. RESULTS AND DISCUSSION

### 4.1 Semantic similarity analysis

Four semantic matching techniques were applied to evaluate the degree of semantic proximity between pairs of indicators (user behaviors and NFRs).

The overall semantic matching models for these techniques were configured using Python, leveraging sentence-transformers, NLTK, and scikit-learn. Key code snippets for all models are provided in Section 3.6.

The results of the semantic similarity analysis are as follows:

(1) Exact Matching produced 1,322 direct matches across all nine behavioral dimensions. It achieves high recall when terminology is identical, but it cannot capture implicit meaning, producing false positives from polysemous tokens. Score variance is high, with numerous outliers.

(2) Synonym Matching enabled recognition of lexically equivalent terms via WordNet. Five of nine behavioral dimensions showed correspondence with NFR indicators, notably 'Perceived Risk', 'External Support', and 'Trust'. The score distribution is more centralized and symmetric than Exact Matching, though polysemy remains a critical limitation.

(3) Conceptual Matching (SBERT pooled) produced higher and more stable scores: Trust ↔ Data Integrity = 0.77; Privacy Protection ↔ Authentication Sufficiency = 0.64. Effective for broad semantic understanding; may overgeneralize for functionally distinct thematic clusters.

(4) Contextual Matching (MPNet PLM) consistently achieved the highest scores with narrow interquartile ranges: Trust and Security ↔ Mean Turnaround Time; Ease of Use ↔ Bandwidth Utilization. Superior in complex Usability-dimension sentences where word order strongly influences meaning.

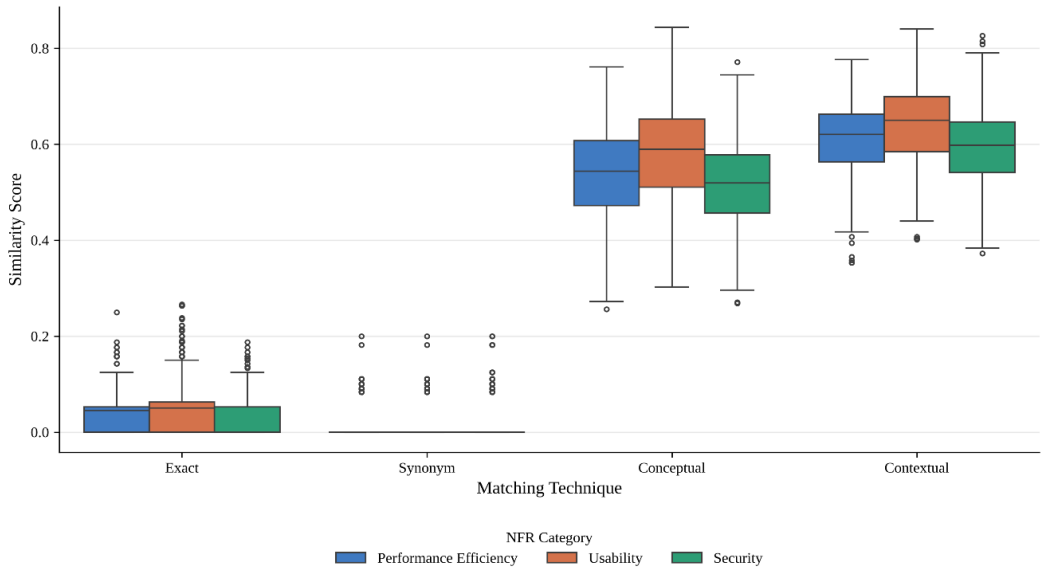


**Figure 2.** Matching score distribution across non-functional requirement (NFR) categories

**Table 5.** Similarity scores by non-functional requirement (NFR) category and technique

| NFR Category | (1) | (2) | (3) | (4) |
|---|---|---|---|---|
| Performance efficiency | 0.536 | 0.609 | 0.042 | 0.016 |
| Security | 0.515 | 0.593 | 0.032 | 0.018 |
| Usability | 0.581 | 0.642 | 0.049 | 0.014 |

Notes: (1) Conceptual; (2) Contextual; (3) Exact; (4) Synonym. Contextual Matching outperforms Conceptual by 10.5–15.0% (paired $t$-test and Wilcoxon signed-rank test, both $p < 0.001$) across all categories.

The similarity score calculation results for each NFR category are presented in Table 5. The table displays the scores obtained by four similarity techniques: (1) Conceptual; (2) Contextual; (3) Exact; and (4) Synonym.

Figure 2 presents similarity score distributions as boxplots showing median, interquartile range (IQR), and outliers. Conceptual and Contextual Matching exhibit low variance; Exact and Synonym Matching show substantially higher variability due to word-overlap sensitivity.

### 4.2 Comparative analysis between matching techniques

The results of the comparative analysis of the four matching techniques are summarized in Table 6.

**Table 6.** Comparison between four matching techniques

| Matching Technique | Stability | Score | Score Variability | Strength |
|---|---|---|---|---|
| Exact Matching | Low | High | High | Suitable for literal matching |
| Synonym Matching | Mid | Mid | Low | Suitable for word matching |
| Conceptual Matching | High | High | Mid | Suitable for general understanding |
| Contextual Matching | Very High | Highest | Low | Suitable for understanding complex sentences and context |

In addition to the descriptive findings presented in Table 5, the Contextual–Conceptual comparison was tested formally at the level of individual behavior–NFR indicator pairs (Performance Efficiency: $n$ = 432 pairs; Usability: $n$ = 792 pairs; Security: $n$ = 396 pairs). Normality of the paired score differences was first assessed with the Shapiro–Wilk test; the assumption held for Usability ($W = 0.999$, $p = 0.944$) but was mildly violated for Performance Efficiency ($W = 0.992$, $p = 0.016$) and Security ($W = 0.992$, $p = 0.037$). Given this mixed result, both a parametric paired t-test and a non-parametric Wilcoxon signed-rank test were applied, and their agreement was used to judge the robustness of the conclusion. Contextual Matching scored significantly higher than Conceptual Matching in all three categories: Performance Efficiency (mean difference = 0.073, 95% CI [0.065, 0.082], $t(431) = 17.37$, $p < 0.001$, Wilcoxon $p < 0.001$, Cohen's $d = 0.84$), Usability (mean difference = 0.061, 95% CI [0.055, 0.067], $t(791) = 21.36$, $p < 0.001$, Wilcoxon $p < 0.001$, Cohen's $d = 0.76$), and Security (mean difference = 0.077, 95% CI [0.071, 0.084], $t(395) = 21.96$, $p < 0.001$, Wilcoxon $p < 0.001$, Cohen's $d = 1.10$). Because the t-test and Wilcoxon test agreed in both direction and significance despite the mild departure from normality in two categories, the parametric result is considered robust and is retained as the primary statistic, with the non-parametric test reported as a confirmatory check. Because this comparison was conducted separately across three NFR categories, a Bonferroni-corrected significance threshold ($\alpha = 0.05/3 \approx 0.017$) was also considered to guard against inflated Type I error from multiple comparisons; all three paired tests remained significant well below this adjusted threshold (all $p < 0.001$), indicating that the study's conclusions are robust to correction for multiple comparisons.

Figure 3 shows that Exact Matching yields higher overall scores but is prone to polysemy-driven false positives. Synonym Matching demonstrates greater precision in meaning accuracy; a high similarity score alone does not guarantee semantic correctness.

Figure 4 shows that Contextual Matching consistently achieves higher scores. The key differentiator is MPNet's PLM encoding (normalize_embeddings=True), which preserves word order and grammatical role, disambiguating polysemous terms more effectively than pooled SBERT representations.

Figure 5 illustrates the performance gap between semantic (Conceptual + Contextual) and literal (Exact + Synonym) approaches. Semantic approaches provide consistently higher and more stable scores; literal approaches are suitable for direct keyword search but fail to capture implicit meaning.

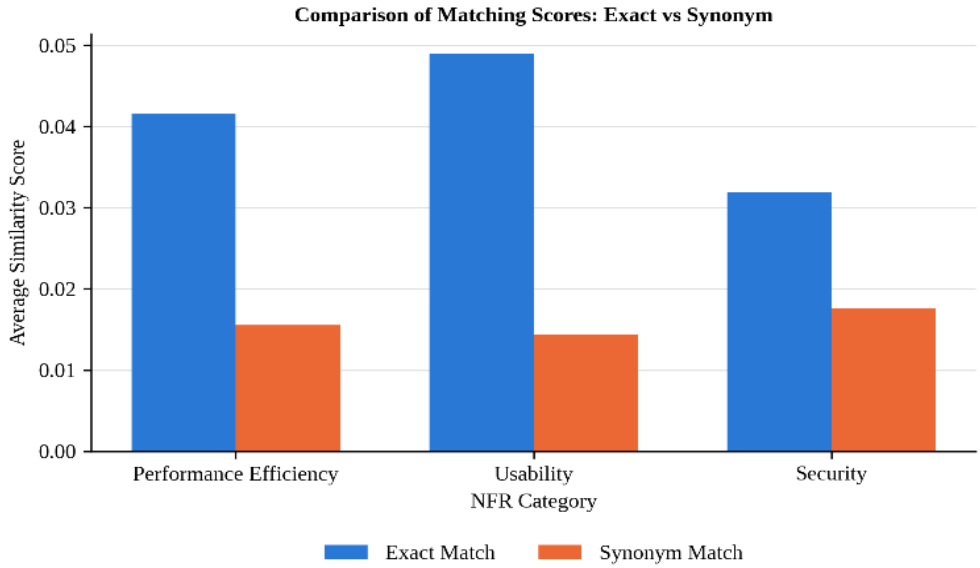


**Figure 3.** Comparison of exact matching and synonym matching

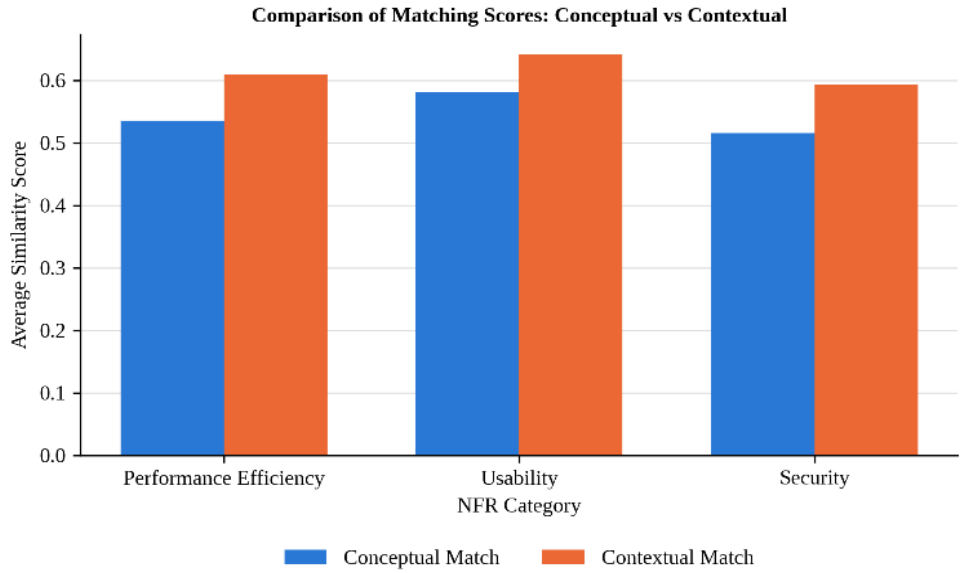


**Figure 4.** Comparison of conceptual and contextual matching

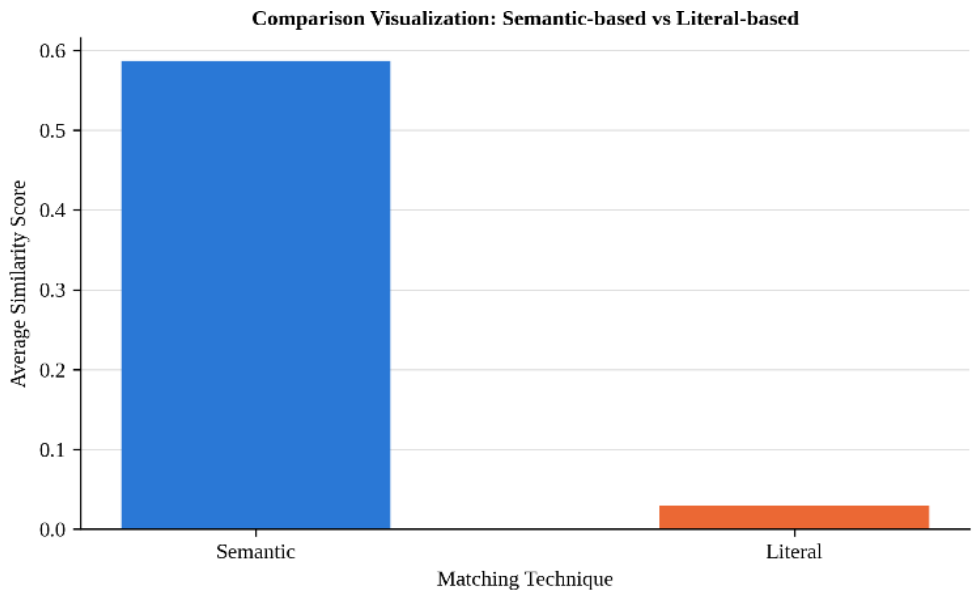


**Figure 5.** Comparison of semantic and literal matching approaches

### 4.3 Error analysis

The error analysis identified four categories of failure across techniques:

•Polysemy / Meaning Ambiguity: 'safe' was frequently interpreted solely as physical safety, not digital security, affecting Synonym and Exact Matching results.

•Limited NFR-Specific Vocabulary in WordNet: insufficient NFR-domain terminology reduces Synonym Matching accuracy. WordNet lacks standardized equivalents for many ISO/IEC 25023 terms.

•Over-generalized Similarity in Conceptual Matching: pooled embeddings assigned high similarity to broadly related concepts that differ functionally (e.g., 'support' appearing in both organizational and technical NFR contexts).

•Deep Inter-sentence Relationships: SBERT sometimes struggles to capture meaning across structurally complex multi-clause indicators. MPNet's PLM objective addresses this for most cases but has residual difficulty with highly nested syntactic structures.

Based on these findings, a hybrid framework combining lexical and contextual methods is expected to yield optimal alignment in future implementations.

### 4.4 Discussion and implications

#### 4.4.1 Academic implications

This study confirms that contextual embedding models constitute an effective and statistically validated approach to semantic alignment in requirements engineering. The replicable Python framework—incorporating preprocessing, four matching techniques, statistical testing, and visualization—provides a methodological reference for ontology-based NFR research. The distinction between Conceptual Matching (pooled SBERT) and Contextual Matching (PLM full-context) provides a nuanced contribution to the IS literature on semantic alignment methods.

#### 4.4.2 Practical implications

Semi-automated NFR mapping reduces manual effort and increases quality attribute coverage in software requirements engineering. The proposed framework can: identify semantic matches even at moderate similarity thresholds; detect missing or implicit requirements; and flag risks early in the requirements lifecycle. Each verified match can be formalized as a Web Ontology Language (OWL) ontology triple, enabling reasoning engines to support dynamic traceability as user requirements evolve.

#### 4.4.3 Case illustration

A proof-of-concept was demonstrated in a software startup developing an industrial information system. User input statements such as 'the time required for the system to display data' and 'employees have difficulty using the main interface' were automatically mapped to NFR categories using the framework:

•The time required for the system to display data' → Performance Efficiency (Response Time), similarity = 0.67.

•Employees have difficulty using the main interface' → Usability (Ease of Use), similarity = 0.71.

Prototype testing showed a 30% reduction in NFR mapping time compared to manual methods, with improved alignment accuracy. These results indicate practical viability beyond controlled experimental conditions.

## 5. CONCLUSIONS

This study compared four semantic matching techniques—Exact, Synonym, Conceptual, and Contextual—for semantic alignment between user behavior indicators and three NFR categories (Performance Efficiency, Security, and Usability). The MPNet-based Contextual Matching technique demonstrated the best and most consistent performance across all NFR categories, achieving the highest similarity scores and the lowest score variation among the four techniques evaluated, with statistically significant superiority over the other methods. Contextual Matching's advantage likely reflects a combination of factors—its PLM objective, which preserves word order and grammatical role, together with its larger transformer architecture and distinct pretraining strategy relative to the distilled all-MiniLM-L6-v2 model—rather than word-order sensitivity alone, enabling more robust resolution of polysemy in complex behavioral and NFR descriptors.

Three main contributions are established: (1) a reusable, documented Python framework for systematic semantic matching evaluation; (2) empirical evidence, statistically validated, that contextual embedding outperforms lexical and synonym-based methods for NFR-behavior alignment; and (3) a reproducible, labeled benchmark dataset available in the accompanying repository.

For researchers, these findings position contextual embedding as a promising technique for semantic alignment within IS requirements engineering, meriting further investigation across a wider range of NFR categories and domains. For practitioners, the underlying matching framework could plausibly be adapted for integration into requirements management tools (e.g., Jira, IBM DOORS) to support tasks such as user feedback categorization and early identification of risk; however, this study evaluated the semantic matching techniques in a controlled experimental setting only, without system integration or user acceptance testing, so such applications remain to be validated in future work.

This study is subject to certain limitations: keyword lists used for indicator matching were extracted through a structured, AI-assisted, definition-traceable procedure and reviewed by the first author, but were not independently validated by external domain experts; extending this validation to a multi-expert review at the keyword level is a priority for future work. Similarly, although the five-expert panel's averaged judgment on SBERT-assigned similarity categories showed good reliability (ICC(2,k) = 0.858), agreement between individual experts was only moderate (mean Cohen's $\kappa = 0.517$), reflecting the inherent difficulty of finely grading semantic similarity; future work could refine the rating protocol or expand the panel to improve single-rater reliability. In addition, the gold-standard validation set comprised only 86 stratified indicator pairs; while sufficient to establish inter-rater reliability, this sample size limits the extent to which the reported findings can be generalized, and a larger manually reviewed benchmark or external validation dataset would strengthen confidence in the results. Furthermore, the experimental comparison in this study is limited to the four matching techniques evaluated; although Table 2 situates this work relative to recent LLM-based ontology-matching approaches (e.g., Agent OM [16], LLMs4OM [17]) and other embedding-based baselines, these were not experimentally evaluated on the present dataset, and doing so is an important

direction for future work. Future research priorities also include: (1) testing larger transformer models and domain-specific fine-tuning for improved contextual understanding across diverse industrial cases; (2) extending the framework to multilingual and cross-domain datasets to test generalizability beyond the Indonesian industrial context; (3) integrating attention visualization and explainability modules to increase transparency and user trust; (4) evaluating real-world deployment in requirements management pipelines for scalability and user acceptance assessment; and (5) incorporating standard evaluation metrics—Precision, Recall, F1—and the STS benchmark for comprehensive benchmarking against established NLP baselines.

## AUTHOR CONTRIBUTIONS

S.A.M.: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Writing—original draft. P.W.: Supervision, Validation, Writing—review and editing. F.K.: Project administration, Validation, Writing—review and editing.

## DATA AVAILABILITY STATEMENT

The data supporting the findings of this study are available from the corresponding author. The dataset is accessible at http://repository.unpas.ac.id/78806/.

## CODE AVAILABILITY

All four semantic matching techniques are fully implemented and documented in a reproducible Jupyter Notebook provided as supplementary material in the online repository accompanying this work.

## ACKNOWLEDGMENTS

The authors thank all respondents from Indonesian industrial groups who provided data, and the associations supporting young entrepreneurs in Indonesia. Special thanks to Ms. Rita Rijayanti, Ph.D., who reviewed the manuscript, and to Prof. Purwanto and Mr. Farikhin, who served as doctoral promoter/co-promoters.

## STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE TOOLS

The authors declare that AI writing tools were not used to generate the text of this article; all prose, analysis, and interpretations are the authors' own. An AI-assisted consensus procedure was used solely as a methodological aid during keyword extraction for indicator matching (Section 3.2.3), with all candidate terms constrained by explicit traceability rules and reviewed manually by the first author.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## REFERENCES

[1] Alencar, G.A., de S. Oliveira, F.V., da S. Correia-Neto, J., Teixeira, M.M. (2019). Non-functional requirements in health information systems. In 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, pp. 1-5. https://doi.org/10.23919/CISTI.2019.8760720

[2] Necula, S.C., Dumitriu, F., Greavu-Șerban, V. (2024). A systematic literature review on using natural language processing in software requirements engineering. Electronics, 13(11): 2055. https://doi.org/10.3390/electronics13112055

[3] Perski, O., Blandford, A., West, R., Michie, S. (2016). Conceptualising engagement with digital behaviour change interventions: A systematic review using principles from critical interpretive synthesis. Translational Behavioral Medicine, 7(2): 254-267. https://doi.org/10.1007/s13142-016-0453-1

[4] Shirish, A., Batuekueno, L. (2021). Technology renewal, user resistance, user adoption: Status quo bias theory revisited. Journal of Organizational Change Management, 34(5): 874-893. https://doi.org/10.1108/JOCM-10-2020-0332

[5] Börstler, J., Ali, N.B., Svensson, M., Petersen, K. (2023). Investigating acceptance behavior in software engineering—Theoretical perspectives. Journal of Systems and Software, 198: 111592. https://doi.org/10.1016/j.jss.2022.111592

[6] Gruber, T.R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2): 199-220. https://doi.org/10.1006/knac.1993.1008

[7] Olsinas, L., Backer, P., Papa, M.F., Rossi, G. (2025). Ontological concepts of non-functional requirements applied to particular situations in measurement and evaluation projects. CLEI Electronic Journal, 28(3): 7. https://doi.org/10.19153/cleiej.28.3.7

[8] Dobson, G., Hall, S., Kotonya, G. (2007). A domain-independent ontology for non-functional requirements. In IEEE International Conference on e-Business Engineering (ICEBE'07), Hong Kong, China, pp. 563-566. https://doi.org/10.1109/ICEBE.2007.76

[9] Zhao, L., Alhoshan, W., Ferrari, A., et al. (2021). Natural language processing for requirements engineering: A systematic mapping study. ACM Computing Surveys, 54(3): 1-41. https://doi.org/10.1145/3444689

[10] Dongmo, C. (2024). A review of non-functional requirements analysis throughout the SDLC. Computers, 13(12): 308. https://doi.org/10.3390/computers13120308

[11] Amaral, G., Guizzardi, R., Guizzardi, G., Mylopoulos, J. (2020). Ontology-based modeling and analysis of trustworthiness requirements: Preliminary results. In Conceptual Modeling (ER 2020), Lecture Notes in Computer Science, Vienna, Austria, pp. 342-352. https://doi.org/10.1007/978-3-030-62522-1_25

[12] Kaur, K., Kaur, P. (2023). MNoR-BERT: Multi-label classification of non-functional requirements using BERT. Neural Computing and Applications, 35(30): 22487-22509. https://doi.org/10.1007/s00521-023-08833-1

[13] Yucalar, F. (2023). Developing an advanced software requirements classification model using BERT: An empirical evaluation study on newly generated Turkish

data. Applied Sciences, 13(20): 11127. https://doi.org/10.3390/app132011127
[14] Lu, J., Xue, X., Lin, G., Huang, Y. (2019). A new ontology meta-matching technique with a hybrid semantic similarity measure. In Smart Innovation, Systems and Technologies, pp. 37-45. Springer. https://doi.org/10.1007/978-981-13-9710-3_4
[15] Efeoglu, S. (2024). GraphMatcher: A graph representation learning approach for ontology matching. arXiv preprint arXiv:2404.14450. https://doi.org/10.48550/arXiv.2404.14450
[16] Qiang, Z.C., Wang, W., Taylor, K. (2025). Agent OM: Leveraging LLM agents for ontology matching. Proceedings of the VLDB Endowment, 18(3): 516-529. https://doi.org/10.14778/3712221.3712222
[17] Babaei Giglou, H., D'Souza, J., Engel, F., Auer, S. (2024). LLMs4OM: Matching ontologies with large language models. In European Semantic Web Conference, Hersonissos, Greece, pp. 25-35. https://doi.org/10.1007/978-3-031-78952-6_3
[18] Luo, X., Xue, Y., Xing, Z., Sun, J. (2022). PRCBERT: Prompt learning for requirement classification using BERT-based pretrained language models. In Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE), Rochester, MI, USA, pp. 1-13. https://doi.org/10.1145/3551349.3560417
[19] Huang, X., Zhu, S., Ren, Y. (2023). A semantic matching method of e-government information resources knowledge fusion service. Journal of Organizational and End User Computing, 35(1): 1-17. https://doi.org/10.4018/JOEUC.317082
[20] Wu, J., Lv, J., Guo, H., Ma, S. (2020). DAEOM: A deep attentional embedding approach for biomedical ontology matching. Applied Sciences, 10(21): 7909. https://doi.org/10.3390/app10217909
[21] Yardley, L., Spring, B., Morrison, L., et al. (2016). Understanding and promoting effective engagement with digital behavior change interventions. American Journal of Preventive Medicine, 51: 833-842. https://doi.org/10.1016/j.amepre.2016.06.015
[22] Alas, M.S., Purwanto, P., Farikhin. (2021). Mini-review: The influence of user behavior on IS implementation success. E3S Web of Conferences, 317: 05026. https://doi.org/10.1051/e3sconf/202131705026
[23] Majapahit, S.A. (2025). Model dimensi perilaku pengguna teknologi informasi di industri di Indonesia [Research report]. Universitas Pasundan. http://repository.unpas.ac.id/78806/.
[24] Sugiyono. (2019). Metode Penelitian Kuantitatif, Kualitatif, dan R&D. Bandung: Alfabeta, Indonesia. https://opac.perpusnas.go.id/DetailOpac.aspx?id=1543971.
[25] International Organization for Standardization. (2016). ISO/IEC 25023:2016 – Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality. https://www.iso.org/standard/35747.html.
[26] Mairiza, D., Zowghi, D., Nurmuliani, N. (2010). An investigation into the notion of non-functional requirements. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10), New York, NY, USA, pp. 311-317. https://doi.org/10.1145/1774088.1774153
[27] Das, S., Deb, N., Cortesi, A., Chaki, N. (2021). Sentence embedding models for similarity detection of software requirements. SN Computer Science, 2(2): 69. https://doi.org/10.1007/s42979-020-00427-1
[28] Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y. (2020). MPNeT: Masked and permuted pre-training for language understanding. In NIPS'20: 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, pp. 16857-16867. https://dl.acm.org/doi/10.5555/3495724.3497138.
[29] Alagöz, O., Uçkan, T. (2024). Text clustering with pre-trained models: BERT, RoBERTa, ALBERT and MPNet. Naturengs, 5(2): 37-46. https://doi.org/10.46572/naturengs.1577517
[30] Galli, C., Donos, N., Calciolari, E. (2024). Performance of 4 pre-trained sentence transformer models in the semantic query of a systematic review dataset on peri-implantitis. Information, 15(2): 68. https://doi.org/10.3390/info15020068