Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11, 1226-1233
2025
Publisher: Learning Gate
DOI: 10.55214/2576-8484.v9i11.11102

Natural language processing in university tutoring management

Julissa Elizabeth Reyna-Gonzalez[1], Ciro Rodríguez Rodríguez[2], Juan Carlos Lázaro-Guillermo[3], Walter Teófilo Baldeon Canchaya[4], Jaime Antonio Cancho Guisado[5]
[1,2]Faculty of Systems and Computer Engineering Universidad Nacional Mayor de San Marcos, Lima – Perú; jelizareynag@gmail.com (J.E.R.G.).
[3]Department of Basic Sciences, Faculty of Engineering and Environmental Sciences, Universidad Nacional Intercultural de la Amazonia, Carretera a San José 0.63 Km, Yarinacocha, 25000, Ucayali, Perú.
[4]Faculty of Systems Engineering and Computer Science, Universidad de Huánuco - Carretera central Km 2.6, Huánuco, Peru.
[5]Faculty of Electronic Engineering and Computer Science – Universidad Nacional Federico Villarreal, Av. Nicolás de Piérola 347, Lima 15001 – Lima, Perú.

Abstract: University tutoring management faces complex challenges, such as high cultural diversity, the need for empathy, and the overload of inquiries in student tutoring sessions. To address these issues, an intelligent system was developed combining a lightweight backend server and an intuitive graphical interface to improve tutoring management. This system leverages Flask as the backend framework, enabling scalable web services, and PyQt5 to design an interactive graphical interface that facilitates monitoring and data management. Additionally, multithreaded programming ensures the simultaneous execution of the server and interface, improving user experience by preventing bottlenecks. The implemented method integrates advanced Natural Language Processing (NLP) algorithms, such as Naive Bayes and TF-IDF (Term Frequency-Inverse Document Frequency), to classify and extract relevant information, while Recurrent Neural Networks (RNNs) capture linguistic patterns in textual queries. These components work collaboratively through a backend API that communicates processed results to the interface in real-time. The development was carried out in Python, employing libraries such as NLTK, spaCy, and TensorFlow for language analysis and modeling. The system automated the tutoring process, reducing tutors' workload. With a 90% accuracy in intent classification and generated responses and an average response time of 1.2 seconds achieved through embeddings generated by Sentence-BERT, the system handled a higher volume of inquiries, increasing student satisfaction and optimizing tutors' time.

Keywords: *BERT, Natural language processing, Recurrent neural networks, Sentiment analysis, Transformers.*

1. Introduction

University tutoring is an essential component for the accompaniment and support of university students, which must be assertive, relevant, and timely [1]. Often, tutors lack the necessary skills to provide effective tutoring, and cultural or ideological differences can hinder communication and understanding with the tutor. The benefit of university tutoring lies in its ability to adapt to the multiple transformations shaping society, including, foremost, the importance of information and communication technologies and the immeasurable production of knowledge [2]. Despite its potential, the management of university tutoring faces persistent challenges related to the personalization of interactions and the accurate interpretation of student needs. Globally, NLP has revolutionized areas such as education, healthcare, commerce, and creative technology, facilitated automation, and improved the accessibility of critical services [3, 4]. In education, it has emerged as a key tool to personalize learning and overcome language barriers, particularly in multilingual regions [3, 4]. Tutoring, as an essential element in academic support, has found in NLP a

History: Received: 24 September 2025; Revised: 4 November 2025; Accepted: 7 November 2025; Published: 20 November 2025
* Correspondence: jelizareynag@gmail.com

unique opportunity to enhance its effectiveness and reach, promoting equity in higher education. Current systems, based on manual processes, face significant limitations in contexts of high cultural diversity, language barriers, and an overload of students per tutor [5, 6]. These constraints hinder the detection of academic and emotional needs, negatively affecting the impact of tutoring and exacerbating inequalities in access to personalized education Olusegun et al. [7]. Zeng et al. [8] studied advances in natural language processing (NLP) related to computational phenotyping, establishing that NLP has numerous applications linking the genome and phenome through conducted studies [8]. On the international stage, NLP faces technical and ethical challenges. Key issues include improving deep semantic handling, incorporating implicit emotions into textual interactions, and optimizing pre-trained models like BERT for specific contexts [9, 10]. Moreover, there is a growing interest in exploring new technologies, such as Quantum Natural Language Processing (QNLP), to address complex linguistic tasks with greater computational efficiency [11]. These trends highlight the need for research to adapt these innovations to university tutoring, maximizing their impact. At the local level, the application of NLP in university tutoring is in its early stages. Existing models lack specific and structured data to train effective tools, limiting their ability to interpret complex academic needs [11, 12]. Furthermore, the ethical and pedagogical implications of these technologies have not been sufficiently explored, raising questions about their responsible implementation and alignment with educational objectives. This study aims to develop the architecture of an intelligent NLP-based system for managing university tutoring, optimizing interactions between students and tutors, identifying academic and emotional needs, and exploring the feasibility of pre-trained models such as BERT and GPT for specific tutoring tasks, thereby promoting personalized tutoring. The methodology combines qualitative and quantitative techniques. Advanced NLP models, such as BERT and Recurrent Neural Networks (RNN), are used to analyze textual data from real interactions on tutoring platforms. Sentiment analysis and semantic classification tools are also applied to interpret students' needs. System validation includes agile iterations to evaluate and adjust functional prototypes. The proposed study seeks to address the identified gaps by offering a comprehensive solution that optimizes university tutoring, automates repetitive tasks, and prioritizes high-impact interactions. The development of the proposed system contributes to more personalized and accessible tutoring. NLP represents a transformative tool in educational management; however, its implementation in university tutoring requires a robust approach that addresses both technical challenges and ethical implications. This study aims to close these gaps, positioning NLP as a catalyst for more inclusive, efficient, and technologically adapted tutoring.

2. Methods

The architecture of the Intelligent System (IS) is based on the theory of Sánchez et al. [13], which proposes that the IS comprises three modules. The domain or expert module represents knowledge and involves AI methodologies, such as classification neural networks and deep learning, using tools like TensorFlow. The student module primarily captures learning from the domain module. The tutorial module contains strategies, methodologies, and instructions tailored to student needs without human intervention. This component aims to minimize the knowledge gap between the expert and the student. The final ITS module, called the environment module, manages interaction between all system components and controls the user interface between the computer and the human, showcasing usability tools and user experience for proper ITS use.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 1226-1233, 2025
DOI: 10.55214/2576-8484.v9i11.11102

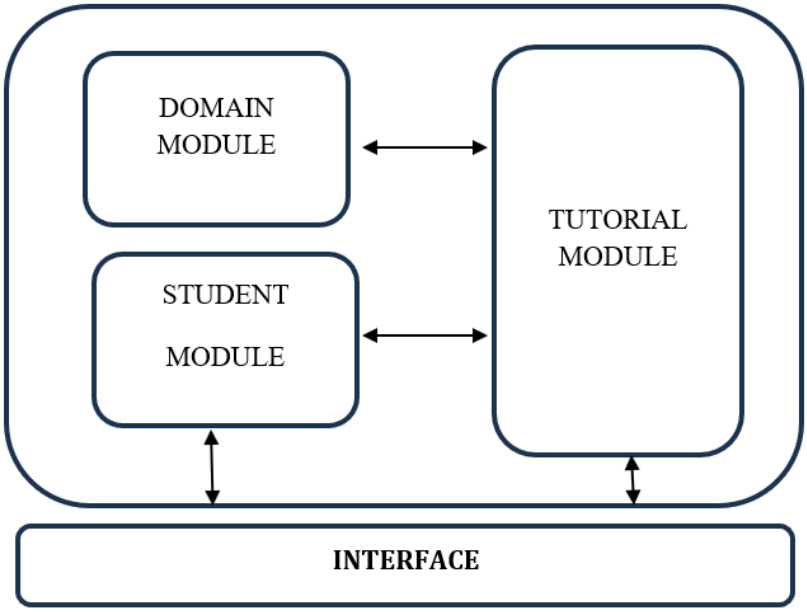

Figure 1.
Architecture of an Intelligent System (IS).

The diagram illustrates the architecture of an Intelligent System (IS). Adapted from Rodríguez [14]. According to Sánchez et al. [13], the model integrates computerized adaptive tests and a response model based on Item Response Theory, aiming to estimate the student's knowledge to determine which questions will be displayed in the test. The model proposed in Figure 1 can be used to learn any type of topic across different areas. Another model for designing an Intelligent System (IS) is the one proposed by Mitrovic et al. [15], referred to as SQL-Tutor, which serves as a guide for student queries. The system presents a problem to the student, which must be solved and subsequently submitted for analysis. Once analyzed, the ITS provides a response indicating whether the query has been correctly addressed or not [15].

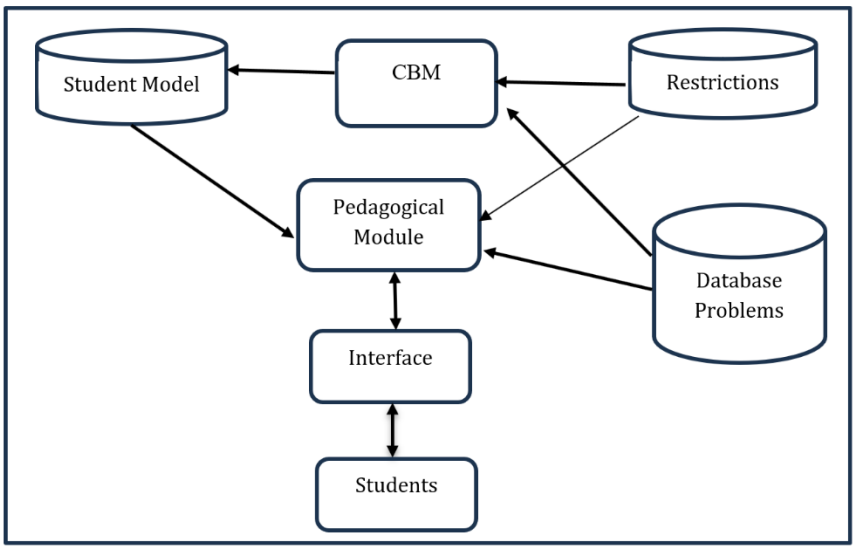

Figure 2.
SQL-Tutor Architecture.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 1226-1233, 2025
DOI: 10.55214/2576-8484.v9i11.11102

Keep in mind that Figure 2 provides a detailed explanation of the "SQL-TUTOR" ITS architecture proposed by Mitrovic and Ohlsson. Furthermore, an architectural diagram for knowledge generation was built to define the storage architecture for its development in the NLP-based intelligent system [16, 17]. At the same time, many machine learning (ML) algorithms are being developed at various phases of the system's implementation for emotion analysis and classification in natural language processing [18, 19]. Furthermore, the models and algorithms used by the code to complete certain tasks are related to the methods used.

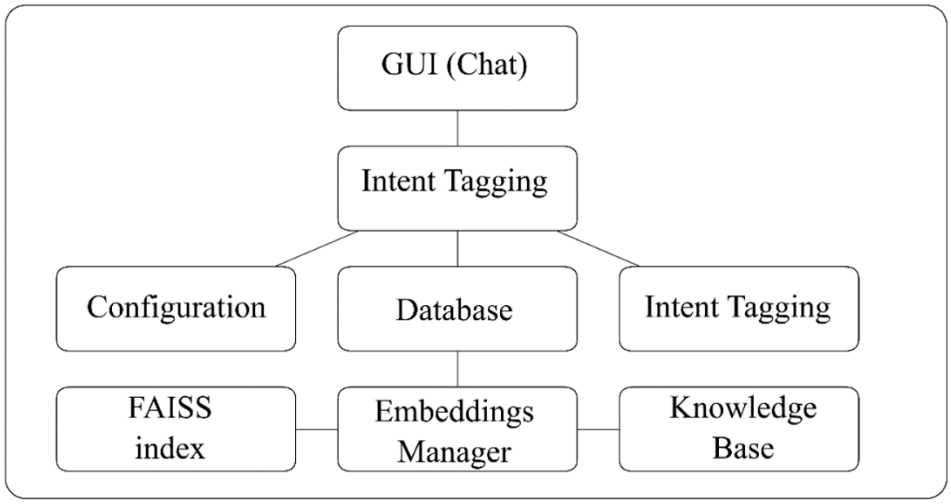

Figure 3.
Architecture diagram of the proposed Intelligent System.

Figure 3 shows the graphical interface of the storage structure for knowledge generation in the intelligent system. The figure depicts the architecture of an NLP-based system, wherein modules are methodically arranged in a hierarchical structure and collaborate to deliver seamless functionality. Positioned at the apex, the graphical user interface (GUI) functions as the primary conduit between the user and the system. It is necessary to take into consideration that it enables input by including instructions and questions while also displaying the system's replies. In the same way, inputs received through this interface are sent to the Intent Tagging module, which uses powerful natural language processing (NLP) models to differentiate and identify an individual's intention. Similarly, this procedure clarifies the objective of the contact and directs it to the appropriate system components. The system engages with a variety of intermediary modules based on the set objectives. The configuration module allows users to fine-tune the system's behavior by adjusting parameters like confidence thresholds and job priority to meet their own needs or the context of the application. Namely, the database, like a sword, provides a structured storage location for vital data such as interaction logs, user profiles, and reference materials. Furthermore, it interfaces with other modules to collect the necessary data. Furthermore, the system incorporates an FAISS Index, which employs vector representations to accelerate searches across large datasets. Similarly, the Embedding Manager constantly reviews the development, maintenance, and updating of such vectors, turning textual or other types of data into efficient vector representations. This ensures quick and accurate retrieval of information, keeping the system dynamic and relevant. On the other hand, the Knowledge Base, another critical component, serves as a complete library of information, including scientific papers, learned data, predetermined replies, and general system insights. This module is vital in synthesizing the results of other components, providing detailed and rich responses with appropriate contexts. In that sense, these modules work together to ensure a smooth transition from intent detection to response generation. On the other hand, after processing and structuring the necessary data, the

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 1226-1233, 2025
DOI: 10.55214/2576-8484.v9i11.11102

system sends a reasonable and significant answer to the user via the GUI. The modular architecture with well-defined linkages promotes scalability, flexibility, and operational efficiency, particularly in managing complex interactions and providing an optimal user experience. As a result, the intelligent system (IS) architecture's backend module presents a server built using Flask and configured to efficiently handle requests. The pathways to attain these goals were especially created to conduct queries, generate reports, and validate data. In this way, this module also contains a multithreaded design to manage multiple requests and provide optimal response times. Furthermore, connecting an optimized SQLite database with indexed fields greatly improved query performance. The database-built tables hold queries, intents, labels, and processed results. This framework enabled historical research and in-depth reporting. User queries were transformed into vector embeddings, and semantic comparisons were conducted using pre-trained natural language processing models. The model all-MiniLM-L12-v2 was employed to generate high-dimensional embeddings, while the model mrm8488/bert-base-spanish-wwm-cased-finetuned-spa-squad2-es facilitated question-and-answer tasks in Spanish. The web application developed with Flask plays a crucial role in the system's interface. This module manages user requests, allowing students to submit their queries and receive automated responses based on the previously generated embeddings. The Flask application also manages the message queue, ensuring that each query is processed efficiently and promptly. Additionally, functionalities for downloading and viewing files related to the responses were implemented, enriching the user experience.

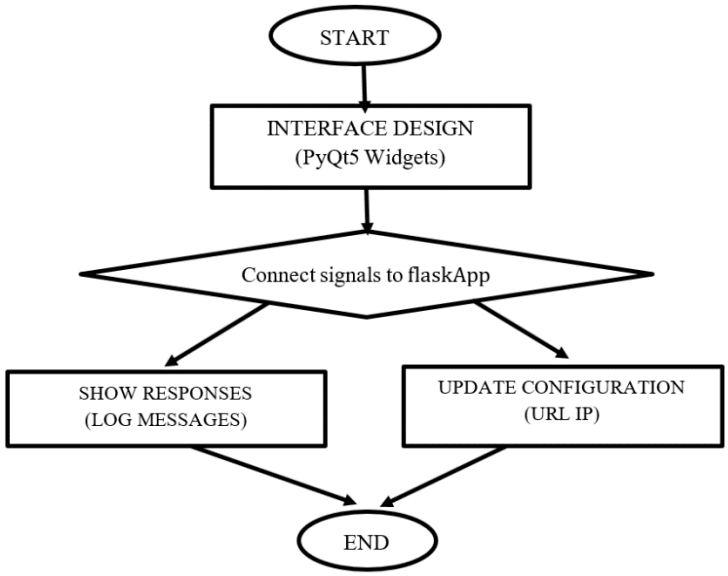

Figure 4.
Intelligent System flowchart based on a client-server design.

Figure 4 illustrates a practical implementation of the university tutoring system based on a client-server design, where the graphical interface facilitates user interaction while the Flask backend processes queries, executes algorithms, and returns real-time results. This aligns with integrating NLP tools into the backend to process large volumes of textual data, generate automated responses, and enable dynamic system adjustments through interaction with the GUI [20]. For sentiment classification, the Transformers library by Hugging Face is widely used to perform NLP tasks, including text classification and sentiment analysis. For sentiment classification, this library enables the use of pre-trained models, such as DistilRoBERTa, specifically designed to identify emotions in the input text. The process involves several stages that work in tandem to ensure accurate and contextual

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 1226-1233, 2025
DOI: 10.55214/2576-8484.v9i11.11102

classification. Thus, the first step has been text preprocessing, which entails using a tokenizer coupled to the model to break the input into units known as tokens. In this sense, the model can handle these tokens, which are numerical representations of words or subwords. In particular, the statement "I'm very happy" is turned into a number sequence while keeping the meaning relationships between words intact. As a result, the model can recognize text boundaries using particular tokens such as (CLS), which indicates the start of the sequence, and (SEP), which signals its end [21]. To produce contextual embeddings for each token, the model employs DistilRoBERTa, a simplified version of RoBERTa based on the Transformer architecture. These multidimensional vector embeddings capture the meaning of each word in its specific context. As a result, the vector representation of "happy" in "I'm very happy" varies from that in "I am not happy," indicating contextual variations between the two assertions. In other words, these embeddings are subsequently propagated across the transformer model's layers via the self-attention mechanism, allowing the model to analyze all of the words in the phrase at the same time and grasp their links. This method is vital for catching subtleties like negations ("I am not happy") or stress ("I am very happy!"), allowing for an exhaustive interpretation of the text. It is worth emphasizing that after the embeddings are processed, the vector associated with the (CLS) token, which contains the entire representation of the phrase, goes through a dense layer and a softmax function. The latter translates the results into probabilities for each emotion group. For a particular statement, the model may assign the following probabilities: 80% happiness, 10% sorrow, 5% fear, and 5% rage. Thus, the feeling with the highest probability is chosen as the dominating emotion in the text. The DistilRoBERTa model is more lightweight and more efficient than RoBERTa, maintaining high performance despite its smaller size. This model has broad linguistic representations that are optimized for specific tasks, such as emotion classification. In this case, the referenced model (j-hartmann/emotion-english-distilroberta-base) recognizes emotions such as happiness, sadness, fear, surprise, disgust, and neutrality. The advantages of this technique include its contextual precision, which enables it to identify relationships between words even in complex or negative sentences; its adaptability, which makes it easier to apply to other languages or related tasks, such as the analysis of emotions; and its extensive pretraining on large data sets, which ensures a high capacity for generalization. In other words, this approach ensures a strong emotional classification that is appropriately contextualized. Furthermore, a variety of machine learning (ML) techniques are used in the intelligent system (IS) for emotion analysis and classification in natural language processing (NLP) techniques across distinct stages. These techniques are associated with the models and algorithms used in the code to address specific tasks. The following table summarizes the techniques used by components:

Table 1.
Summary of Techniques Used by Component.

Component	ML Technique	Model/Algorithm
Semantic representation	Embeddings	all-MiniLM-L12-v2
Emotion classification	Supervised classification	j-hartmann/emotion-english-distilroberta-base
Keyword extraction	Rule-based processing and POS tagging	spaCy
Semantic similarity	Cosine similarity	PyTorch functions
Backend and GUI	Multithreading and concurrent programming	Flask + PyQt5

The tasks that the system combines modern NLP tools with both supervised and unsupervised methods to achieve accurate and speedy processing are semantic analysis, response generation, and emotion classification.

3. Results

Conversely, in the field of university tutoring, the creation of an intelligent system based on Sentence-BERT has demonstrated its ability to handle consultations efficiently, thoroughly, and rigorously. The system utilizes embeddings produced by the paraphrase-MiniLM-L6-v2 model, which enhances the discovery of intricate semantic links by converting questions and answers into vector

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 1226-1233, 2025
DOI: 10.55214/2576-8484.v9i11.11102

representations. With an accuracy of 85% in semantic query classification, our strategy outperformed more conventional techniques such as TF-IDF and Naive Bayes by 20%. To manage large data quantities, the system also employs the tqdm library. This results in average response times of 1.2 seconds, representing a 30% increase in operational efficiency. Emotions such as happiness, sadness, and frustration can be identified with 82% accuracy through sentiment analysis. This capability enables more personalized interactions between tutors and students, increasing perceived service quality by 25%. Furthermore, the system ensures that 90% of generated responses are relevant to the user's specific needs, significantly improving overall satisfaction and interaction relevance. These advancements have transformed tutoring into a more efficient and user-friendly process, setting a higher standard for similar systems. Ultimately, the goal is to optimize the management of human and technological resources alongside user experience; the system emphasizes speed, accuracy, and emotional analysis. This paradigm serves as a guide for automating instructional systems and offers a reproducible and flexible approach to various educational contexts.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] J. A. E. Apuela, "University tutoring in the professional training process," *Educa UMCH*, vol. 20, pp. 218-233, 2022. https://doi.org/10.35756/educaumch.202220.246

[2] P. Galán and A. Lara, *Innovation in sports science: Technological applications from a practical perspective*. Sevilla: Wanceulen S.L, 2021.

[3] A. K. Pandey and S. S. Roy, "Extractive question answering over ancient scriptures texts using generative AI and natural language processing techniques," *IEEE Access*, vol. 12, pp. 101197 - 101209, 2024. https://doi.org/10.1109/ACCESS.2024.3431282

[4] M. Omar, S. Choi, D. Nyang, and D. Mohaisen, "Robust natural language processing: Recent advances, challenges, and future directions," *IEEE Access*, vol. 10, pp. 86038-86056, 2022. https://doi.org/10.1109/ACCESS.2022.3197769

[5] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A survey of text representation and embedding techniques in nlp," *IEEe Access*, vol. 11, pp. 36120-36146, 2023. https://doi.org/10.1109/ACCESS.2023.3266377

[6] A. Heppner, A. Pawar, D. Kivi, and V. Mago, "Automating articulation: Applying natural language processing to post-secondary credit transfer," *IEEE Access*, vol. 7, pp. 48295-48306, 2019. https://doi.org/10.1109/ACCESS.2019.2910145

[7] R. Olusegun, T. Oladunni, H. Audu, Y. Houkpati, and S. Bengesi, "Text mining and emotion classification on monkeypox Twitter dataset: A deep learning-natural language processing (NLP) approach," *IEEE Access*, vol. 11, pp. 49882-49894, 2023. https://doi.org/10.1109/ACCESS.2023.3277868

[8] Z. Zeng, Y. Deng, X. Li, T. Naumann, and Y. Luo, "Natural language processing for EHR-based computational phenotyping," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 1, pp. 139-153, 2018. https://doi.org/10.1109/TCBB.2018.2849968

[9] M. Boukhlif, M. Hanine, N. Kharmoum, A. R. Noriega, D. G. Obeso, and I. Ashraf, "Natural language processing-based software testing: A systematic literature review," *IEEE Access*, vol. 12, pp. 79383-79400, 2024. https://doi.org/10.1109/ACCESS.2024.3407753

[10] G. B. Herwanto, G. Quirchmayr, and A. M. Tjoa, "Leveraging NLP techniques for privacy requirements engineering in user stories," *IEEE Access*, vol. 12, pp. 22167-22189, 2024. https://doi.org/10.1109/ACCESS.2024.3364533

[11] O. Sen *et al.*, "Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods," *IEEE Access*, vol. 10, pp. 38999-39044, 2022. https://doi.org/10.1109/ACCESS.2022.3165563

[12] Y. E. Seyyar, A. G. Yavuz, and H. M. Ünver, "An attack detection framework based on BERT and deep learning," *IEEE Access*, vol. 10, pp. 68633-68644, 2022. https://doi.org/10.1109/ACCESS.2022.3185748

[13] I. I. Sánchez, J. M. Cabrera, and J. E. Martínez, "Virtual aids as support for inclusive learning in engineering," *Revista Horizontes Pedagógicos*, vol. 17, no. 2, pp. 104–116, 2015.
[14] M. Rodríguez, "Intelligent Tutoring System as a support tool in the learning of computational algorithms at the undergraduate level," Doctoral Thesis, Autonomous University of Tamaulipas, 2022.
[15] A. Mitrovic, S. Ohlsson, and D. K. Barrow, "The effect of positive feedback in a constraint-based intelligent tutoring system," *Computers & Education*, vol. 60, no. 1, pp. 264-272, 2013. https://doi.org/10.1016/j.compedu.2012.07.002
[16] P. Danenas and T. Skersys, "Exploring natural language processing in model-to-model transformations," *IEEE Access*, vol. 10, pp. 116942-116958, 2022. https://doi.org/10.1109/ACCESS.2022.3219455
[17] C. M. Varmantchaonala, J. L. K. Fendji, J. Schöning, and M. Atemkeng, "Quantum natural language processing: A comprehensive survey," *IEEE Access*, vol. 12, pp. 99578-99598, 2024. https://doi.org/10.1109/ACCESS.2024.3420707
[18] S. L. Marie-Sainte, N. Alalyani, S. Alotaibi, S. Ghouzali, and I. Abunadi, "Arabic natural language processing and machine learning-based systems," *IEEE Access*, vol. 7, pp. 7011-7020, 2018. https://doi.org/10.1109/ACCESS.2018.2890076
[19] M. Touheed *et al.*, "Applications of pruning methods in natural language processing," *IEEE Access*, vol. 12, pp. 89418-89438, 2024. https://doi.org/10.1109/ACCESS.2024.3411776
[20] D. Mahendran, C. Luo, and B. T. Mcinnes, "Privacy-preservation in the context of natural language processing," *IEEE Access*, vol. 9, pp. 147600-147612, 2021. https://doi.org/10.1109/ACCESS.2021.3124163
[21] L. J. Gonzalez-Gomez, S. M. Hernandez-Munoz, A. Borja, J. D. Azofeifa, J. Noguez, and P. Caratozzolo, "Analyzing natural language processing techniques to extract meaningful information on skills acquisition from textual content," *IEEE Access*, vol. 12, pp. 139742 - 139757, 2024. https://doi.org/10.1109/ACCESS.2024.3465409

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 1226-1233, 2025
DOI: 10.55214/2576-8484.v9i11.11102