computers

MDPI

Article

Fuzzy Memory Networks and Contextual Schemas: Enhancing ChatGPT Responses in a Personalized Educational System

Christos Troussas *, Akrivi Krouska, Phivos Mylonas, Cleo Sgouropoulou and Ioannis Voyiatzis

Department of Informatics and Computer Engineering, University of West Attica, 12243 Egaleo, Greece; akrouska@uniwa.gr (A.K.); mylonasf@uniwa.gr (P.M.); csgouro@uniwa.gr (C.S.); voyageri@uniwa.gr (I.V.)
* Correspondence: ctrouss@uniwa.gr

Abstract: Educational AI systems often do not employ proper sophistication techniques to enhance learner interactions, organize their contextual knowledge or even deliver personalized feedback. To address this gap, this paper seeks to reform the way ChatGPT supports learners by employing fuzzy memory retention and thematic clustering. To achieve this, three modules have been developed: (a) the Fuzzy Memory Module which models human memory retention using time decay fuzzy weights to assign relevance to user interactions, (b) the Schema Manager which then organizes these prioritized interactions into thematic clusters for structured contextual representation, and (c) the Response Generator which uses the output of the other two modules to provide feedback to ChatGPT by synthesizing personalized responses. The synergy of these three modules is a novel approach to intelligent and AI tutoring that enhances the output of ChatGPT to learners for a more personalized learning experience. The system was evaluated by 120 undergraduate students in the course of Java programming, and the results are very promising, showing memory retrieval accuracy, schema relevance and personalized response quality. The results also show the system outperforms traditional methods in delivering adaptive and contextually enriched educational feedback.

Keywords: fuzzy logic; adaptive learning systems; generative AI in education; ChatGPT; memory retention; programming education; personalized education

check for updates

Academic Editor: Paolo Bellavista

Received: 27 January 2025
Revised: 25 February 2025
Accepted: 3 March 2025
Published: 4 March 2025

Citation: Troussas, C.; Krouska, A.; Mylonas, P.; Sgouropoulou, C.; Voyiatzis, I. Fuzzy Memory Networks and Contextual Schemas: Enhancing ChatGPT Responses in a Personalized Educational System. *Computers* **2025**, *14*, 89. https://doi.org/10.3390/computers14030089

1. Introduction

Personalized education encompasses adapting to learners' personal needs and preferences, such as prior knowledge and learning style [1,2]. Specifically, learning computer programming, such as the Java language, presents several challenges due to the complexity of the language [3]. Indeed, Java has a rich feature set, including, among others, concepts of object-oriented programming, multithreading and exception handling, and thus requires learners to grasp foundational and advanced concepts in a structured manner. Beginners often face obstacles in understanding basic syntax or control structures, while more advanced learners may face difficulties in optimizing code and implementing design patterns. As such, the different learners' skill levels pose a barrier to traditional educational software to adapt to their individual needs, specifically in programming education. On top of that, in Java, concepts like problem solving or debugging demand contextualized feedback tailored to each learner's progress. So, the idea of "one size fits all" seems to not be valid, and even personalized learning approaches need refinement for further enhancing the adaptivity to learners and providing them with contextual support.

In particular, this necessity for adaptive systems that mimic human-like memory retention and contextual understanding becomes highly relevant within the modern framework of education, especially in programming education [4]. Human memory intrinsically has a dynamic nature, emphasizing the latest and most relevant information while naturally forgetting the outdated or less important details [5]. This natural process helps humans avoid distraction from current tasks by not having to pay attention to irrelevant information. This is again an essential capability of educational artificial intelligence in the pursuit of providing personalized learning experiences that work. The adaptive systems will need to remember some important user interactions and adapt their content delivery with progress in learning and provide responses contextualized to a history of interaction with the learner. Moreover, it will also, with the contextual understanding, allow those systems to provide some related topics structured and thematically approached in such a way that meaningful clustering should offer a proper manner of learning [6]. Such a combination-insured memory retention together with the contextual adaptability provide learners timely, relevant, and personalized support for better learning and improved learning engagement.

Most of the existing educational AI systems lack the power to prioritize recent and relevant interactions and lay a basis for remembering contextual knowledge for long periods in adaptive ways [7–10]. Many of them either rely on static memory structures or have no retention of long-term context, and responses are totally incapable of reflecting learner progress or evolution in their needs [11]. For example, interactions that were made on previous sessions are usually lost and thus cannot build upon the prior knowledge let alone detect recurring difficulties. Also, the mechanism for distinguishing between high-priority recent interactions and older ones with less relevance in current conversational flow eludes existing models [12]. This lack of adaptive memorization impairs the system's ability to tailor learning experiences, especially in rich domains such as programming education, where context and progress are paramount. These gaps can only be filled with dynamic memory systems that genuinely mimic human-like forgetting and prioritization to ensure that learners receive timely and contextually aware support [13].

This paper contributes the Fuzzy Memory Network that models memory retention through the application of fuzzy weights, hence simulating human-like forgetting to dynamically prioritize recent and relevant interactions. The system applies fuzzy logic in computing retention weights for user interactions regarding recency and contextual importance to thereby support the gradual decay of memory relevance over time. The fuzzy retention mechanism will ensure that the system stays focused on interactions that are most relevant to the learner's current progress, deprioritizing older or less relevant details without discarding them. The system further enhances contextual understanding by employing a Schema Manager to perform the thematic clustering of related interactions. This component groups user interactions into meaningful schemas based on semantic similarity, creating a structured and dynamic framework for knowledge organization. These thematic clusters enable the system to provide richer and more cohesive educational content, aligning retrieved materials with the learner's ongoing queries and prior interactions. The clustering approach ensures that the retrieval process is both efficient and contextually relevant, supporting personalized and progressive learning pathways. These components are tightly integrated with a generative AI model, specifically ChatGPT, for delivering personalized and context-aware responses. The Fuzzy Memory Network and Schema Manager collaboratively provide enriched input to ChatGPT, equipping it with user-specific contextual knowledge and prioritized interactions to enhance the quality of response. This integration will enable the generative model to build responses that incorporate not only historical learner interactions but also their present needs, closing the gap between static educational systems and adaptive learning environments. By incorporating fuzzy memory retention, thematic clustering, and generative AI capabilities, the proposed system addresses key shortcomings of educational frameworks in use today, allowing for a scalable, personalized, and contextually enriched learning experience that responds to the complexity of programming education.

2. Related Work

The concept of memory in AI systems plays a pivotal role in enabling long-term interaction and dynamic learning. In conversational AI systems [14–19], memory is often implemented as a static or short-term mechanism, where interactions are limited to a fixed context window. For example, ChatGPT can only store context up to the scope of the present session and has no memory to recollect previous sessions as soon as it is closed. This prevents continuity and continuity to the previous interaction of the users, making their effectiveness in personal education somewhat minimal. Traditional educational AI systems, like Intelligent Tutoring Systems (ITSs), depend on a predefined knowledge base and a static rule-based approach for the delivery of instructional content [20–27]. Since these systems focus more on structured lessons and guide learners through personalized guidance, they lack the adaptive memory mechanism, which may reflect recency, relevance, and user-specific learning trajectories. More significant shortcomings of standard models have spurred attempts to alleviate these issues: augmenting them with external memory modules either in the form of attention mechanisms or knowledge graphs that can retain interaction histories more elaborately. Nonetheless, these often fail to adequately model the important aspect of gradually forgetting the irrelevant information such as human memory. This gap underscores the need for more dynamic memory systems capable of retaining, prioritizing, and decaying information in a human-like manner, especially in domains like programming education where contextual progression is critical.

Fuzzy logic has emerged as one of the powerful tools in the modeling of AI that expresses gradual changes and manages uncertainties inherent in real-world scenarios. Unlike binary logic, relying on crisp true-or-false states, fuzzy logic allows systems to represent information with partial memberships. This enables nuanced decision making and adaptability. This feature makes fuzzy logic especially apt for applications in dynamic transitions like adaptive systems, user modeling, and decision support systems in various fields, including education [28–38]. One of the applications of fuzzy logic is in the PARSAT system [32], where weights with fuzzy application model student knowledge levels. PARSAT uses the trapezoidal membership function in defining states of knowledge and therefore allows it to dynamically adjust instructions as the learner moves along the knowledge chain. This will help capture the continuity of human learning, whereby transitions across knowledge levels are not abrupt but rather smooth. In addition to education, fuzzy logic has also found other applications in control systems, robotics, and natural language processing for handling uncertainty and variability. In the context of memory retention, fuzzy logic provides a promising framework for modeling time decay and prioritization, as it can dynamically assign relevance to interactions based on recency and other factors. Despite its potential, the application of fuzzy logic in personalized programming education remains underexplored, offering a fertile area for innovation.

Thematic clustering and contextual organization, often referred to as schema-based learning, have gained prominence as effective techniques for managing and retrieving information in AI systems [39–45]. Inspired by cognitive psychology, schemas are frameworks for organizing related knowledge into coherent structures, allowing both humans and machines to generalize and adapt to new situations. Schema-based learning has been highly used in many applications of AI, including recommendation systems, knowledge management, and dialogue systems. For instance, the use of knowledge graphs usually relies on clustering algorithms that group semantically similar concepts together so that a system can offer more contextualized responses. Educational AI systems have also considered the application of schemas in organizing instructional content to ensure related topics are presented sequentially and in context. For instance, a tutoring system might cluster concepts like "loops", "conditionals", and "iterations" into a "control structures" schema to enhance contextual learning. Advances in natural language processing have further facilitated schema formation by enabling systems to calculate semantic similarity between interactions using embedding techniques, such as cosine similarity. However, while schema-based learning has proven effective in enhancing contextual relevance, existing systems often lack integration with dynamic memory mechanisms. Thus, they lack responsiveness to shifting user requirements and also do not prioritize current important interactions that tend to reduce their usages in teaching programming.

There is still a big gap to fill between integrating fuzzy memory retention and schema-based contextual learning into personalized programming education [46]. Existing conversational and educational AI systems, as presented earlier, are either too rigid, with static knowledge bases, or too transient, discarding past interactions once a session ends. Although fuzzy logic has seen many applications in the context of adaptive learning systems, when modeling knowledge states, very few are those applications focusing on memory retention and time decay [47]. Likewise, schema-based clustering was successfully applied for organizing related content but is generally unaware of the prioritization mechanism concerning the memorization and therefore unable to adapt dynamically with regard to recency and relevance. The intersection of these approaches—combining fuzzy memory retention with thematic clustering—offers immense potential for addressing the diverse and evolving needs of programming learners. A system that integrates these elements could emulate human-like memory, prioritizing recent and important interactions while organizing knowledge contextually to facilitate a tailored and adaptive learning experience. This paper fills this gap by proposing a Fuzzy Memory Network that bridges these methodologies, delivering personalized and context-aware education for Java programming.

3. System Architecture

The proposed system involves three modules including the Fuzzy Memory Module, Schema Manager, and Response Generator. To generate context-rich personalized educational responses, modules integrate one with another such that human-like memory retention happens, combining knowledge within the blocks and response generation by selecting appropriate responding contexts to their learners. Therefore, the coordination of the modules is illustrated in Figure 1 and briefly described below:

- Fuzzy Memory Module:
  This module captures and holds user interactions by fuzzy weights that simulate memory decay and focus on recent relevant information. It emulates human-like forgetting by granting degrees of relevance to memories over time, so that the system concentrates on interactions which are most relevant to the query at hand.
- Schema Manager:
  The Schema Manager categorizes retrieved memories into thematic clusters called schemas. Clustering related interactions by contextual similarity allows the module to structure knowledge dynamically and create a strong representation of the learner's progress and areas of focus.
- Response Generator:
  The Response Generator synthesizes information from the Fuzzy Memory Module and Schema Manager to create personalized responses generated using a generative AI model (ChatGPT), hence ensuring relevance to the current query of the learner and enriched by past interactions.

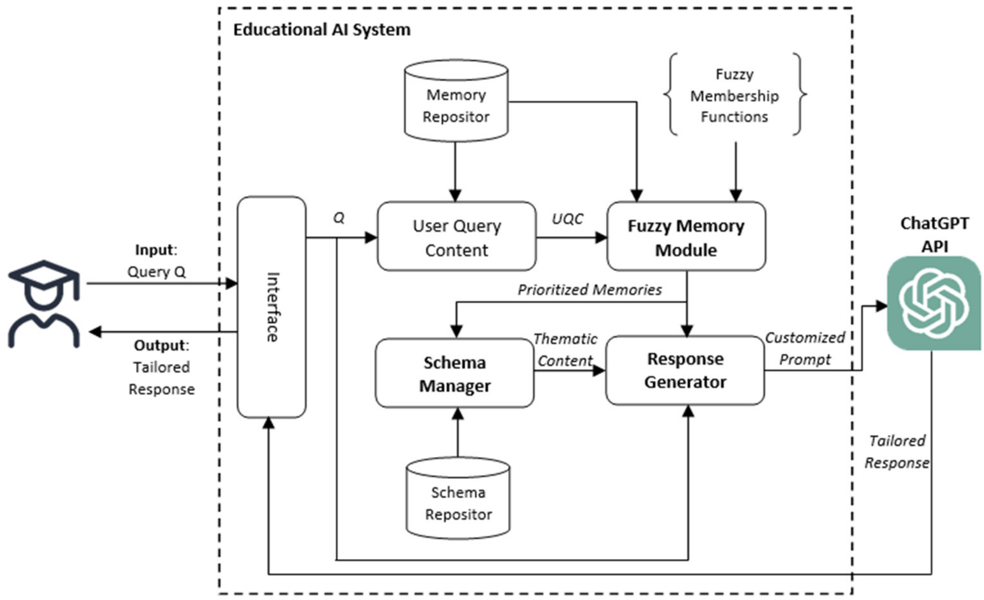

Figure 1. Logical architecture.

The system works together in an integrated process to provide interactive, personalized responses dynamically for training purposes. To better understand system functionality, suppose a learner posts the question to the system on how to "iterate over the array in Java". The Fuzzy Memory Module immediately identifies and recovers relevant interaction history. Based on values of their fuzzy weights, this reflection of interaction recency and relevance is put across, thus, ensuring that, for example, last queries for "loops" and "arrays" are given relevant importance.

The Schema Manager then takes over, organizing these retrieved interactions into meaningful thematic clusters or schemas. Here, relevant interactions are then grouped within a "Control Structures Schema" providing a structured context that aligns with the learner's query. This further helps the system understand the broader topic and the relationships among the pieces of information retrieved.

Finally, the Response Generator creates a customized response based on the result of the memory retrieval as well as the schema organization. Combining the immediate query with the historical context of the learner and his/her progress can give an enriched explanation such as detailed steps of applying for-loops with arrays in Java.

The integration of modules allows the system to be responsive to the learners' needs and provide support in a contextual manner, hence making the educational experience both personalized and responsive.

A three-module framework—Fuzzy Memory Module, Schema Manager, and Response Generator—is proposed for the design of such a system. Unlike traditional, rule-based Intelligent Tutoring Systems or LLM tutors which discard memory associations with previous instances, it is designed to constantly search for the most recent text or interaction and then systematically cluster these, by theme, into schemas It also adjusts them according to the different semantics that arise as learners progress from one section of a lesson to another. When external data retrieval methods (like memory-based models) lack natural continuity, relevance of response, and the capability for flexible adaptation, this leveled approach provides what retrieval-based AI models miss. But our system is not static like the analogs of 2024–2025, which retrieve without prioritizing externally generated knowledge, nor are these LLM tutors able extend long-term changes. In the personalization and semantic retrieval front, it presents an integration of fuzzy memory retention with scaffold retrieval systems. While this architecture increases the computational burden and relies on finely tuned similarity thresholds, it significantly enhances adaptability in learning, coherence in presentations, and the situationality of interactions, bridging the gulf between generative AI fluency and structured pedagogical support.

3.1. Fuzzy Memory Module

The system's backbone is the Fuzzy Memory Module, which models memory retention through fuzzy logic. This approach thus follows human memory dynamics in defining smooth transitions between states of recency, intermediate retention, and forgetting. Every interaction will be weighted with respect to time, using fuzzy membership functions that provide relevance in accord with when the interaction occurred. The system assigns a value to a memory based on metadata captured during the interaction with the user. This value captures when the interaction occurred and is thus captured as part of the memory record within the system's database. Every time a user interacts with the system in quest of an answer to a query, by giving some feedback, or providing an answer to a query, the system captures that interaction as a "memory". Apart from the interaction content, the system logs metadata, including the precise timestamp (t_i), i.e., the date and time of the occurrence, the interaction type, i.e., query, clarification request, or feedback, and other contextual information, such as the topic or schema to which the interaction is associated, i.e., "Control Structures" or "Object-Oriented Programming". These interactions are committed to a structured memory repository that holds metadata such as a unique identifier, the timestamp, and contextual embeddings that capture the semantic content of the interaction. When a new query is processed, the Fuzzy Memory Module recalls relevant memories from the repository by extracting the elapsed time ($\Delta t = t - t_i$, where t is the current time) and, if needed, computing semantic similarity to further filter relevance. This elapsed time is taken as input for the predefined membership functions—Recent, Intermediate, and Forgotten—and evaluates the memory's relevance degree, thereby helping the system determine its priority for retrieval based on temporal thresholds and contextual significance. Figure 2 presents membership functions scheme.

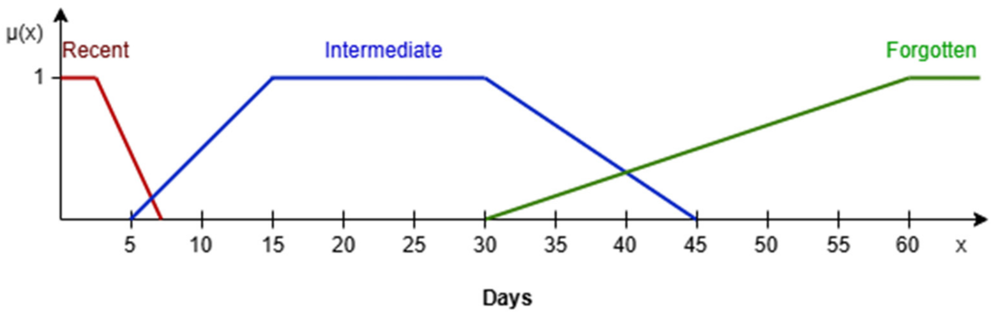

Figure 2. Membership functions scheme.

The Recent membership function manages to incorporate interaction relevance from when the events occur within the last 7 days. This gives full membership ($\mu_R = 1$) to interactions that occurred within the first 3 days, representing peak relevance. The relevance then decreases linearly from day 3 to day 7, after which the membership becomes zero.

$$\mu_R(t) = \begin{cases} 1,\ if\ t \leq 3 \\ \frac{7-t}{7-3},\ if\ 3 < t < 7 \\ 0,\ if\ t \geq 7 \end{cases}$$

where t is the number of days since the interaction occurred.

Between day 0 and day 3 ($t \leq 3$), the interaction is at its maximum importance (μ_R = 1). From day 3 to day 7, the interaction's relevance decreases in a linear way until it reaches μ_R = 0 at =7.

The Intermediate membership function ($\mu_I(t)$) denotes interactions that occurred between 5 and 45 days ago, reflecting medium-term retention. The relevance rises gradually from day 5 to day 15, stays at peak membership (μ_I = 1) from day 15 to day 30, and then decreases linearly until day 45.

$$\mu_I(t) = \begin{cases} 0,\ if\ t \leq 5 \\ \frac{t-5}{15-5},\ if\ 5 < t \leq 15 \\ 1,\ if\ 15 < t \leq 30 \\ \frac{45-t}{45-30},\ if\ 30 < t \leq 45 \\ 0,\ if\ t > 45 \end{cases}$$

The interaction begins to gain relevance from day 5 to day 15 as it transitions into the intermediate memory range. Between day 15 and day 30, the interaction achieves peak relevance (μ_I = 1), highlighting the role of medium-term retention in reinforcing knowledge. After day 30, relevance gradually diminishes, eventually reaching μ_I = 0 at t = 45.

The Forgotten membership function ($\mu_F(t)$) addresses interactions older than 30 days, representing long-term memory. The membership remains zero for interactions up to day 30 and then increases linearly between days 30 and 60. Beyond day 60, it stabilizes at its maximum value (μ_F = 1).

$$\mu_F(t) = \begin{cases} 0,\ if\ t \leq 30 \\ \frac{t-30}{60-30},\ if\ 30 < t \leq 60 \\ 1,\ if\ t > 60 \end{cases}$$

Memories begin to transition into the "Forgotten" range after day 30 with relevance gradually increasing as t approaches day 60. After day 60, the interaction is fully considered in the forgotten state (μ_F = 1), though it may still be retrieved if deemed relevant.

The specific time thresholds in the Fuzzy Memory Module are derived from empirical research on memory retention, notably the replication of Hermann Ebbinghaus's forgetting curve. In [48], it was shown that memory retention declines rapidly within the first few days of learning with a significant drop occurring in the initial 24 h. This sharp early decline justifies the inclusion of the "Recent" membership function, covering the first 7 days, to emphasize the critical period for memory reinforcement. Subsequently, the "Intermediate" membership function spans 5 to 30 days, reflecting a period where, although the rate of forgetting slows, ongoing review is essential for transitioning information into long-term memory. The "Forgotten" membership function is how the member is treated after 30 days—that is, the period when, if not refreshed periodically, information will be forgotten. The time periods are put together to simulate natural memory decay defined by Ebbinghaus [49] in this way so the system can rank information for retrieval in a manner consistent with human cognitive processes.

Thus, the calculation of fuzzy weights is an essential operation that defines how much a memory at any time t is relevant. It will depend on the output of the three membership functions Recent ($\mu_R(t)$), Intermediate ($\mu_I(t)$), and Forgotten ($\mu_F(t)$), which define how the system models the relevance of the memory's course of change in time. With the outputs of the mentioned membership functions, the system makes use of the maximum operator for ensuring the best state selection for this memory.

The fuzzy weight of a memory at time t is given by

$$M(t) = \max(\mu_R(t),\ \mu_I(t),\ \mu_F(t))$$

where $\mu_R(t)$ is the membership value for the Recent state, $\mu_I(t)$ is the membership value for the Intermediate state, and $\mu_F(t)$ is the membership value for the Forgotten state [49].

The primary reason for the application of triangular membership functions is their computational efficiency and simplicity as well as their educational similarity with the human memory decay models. The triangular function is performed piecewise linearly fast, while the sigmoidal or Gaussian function needs complex hyperparameter tuning, despite performing well on modeling gradual forgetting, as predicted by the Ebbinghaus curve. Their flexibility, along with low computational overhead, make them suitable for use in modeling memory retention processes in real time.

Let us demonstrate the procedure of the calculation of the fuzzy weight by taking into account an interaction that occurred 10 days ago. Its membership values are evaluated against three states: Recent, Intermediate and Forgotten. Starting with the Recent membership function ($\mu_R(t)$), which is active for interactions occurring between 0 and 7 days, the membership value decreases linearly from 1 at t = 3 to 0 at t = 7. Since t = 10 falls outside this range, the value of $\mu_R(10)$ is 0, indicating that the interaction is no longer recent.

For the Intermediate membership function ($\mu_I(t)$), which applies to interactions between 5 and 45 days, the membership value rises linearly from t = 5 to t = 15, remains at its peak between t = 15 and t = 30, and then decreases linearly until t = 45. At t = 10, the interaction lies within the initial rising phase of this function. Substituting into the formula, $\mu_I(10) = \frac{t-7}{15-7} = \frac{10-7}{8} = 0.375$. This indicates that the interaction holds moderate importance in the intermediate memory range.

The Forgotten membership function ($\mu_F(t)$) is relevant only for interactions occurring after 30 days. Since t = 10 lies outside this range, the value of $\mu_F(10)$ is 0, reflecting that the interaction has not yet transitioned into the forgotten state.

To calculate the fuzzy weight, the maximum operator is applied to the membership values:

$$M(10) = \max(\mu_R(10),\ \mu_I(10),\ \mu_F(10)) = \max(0,\ 0.375,\ 0) = 0.375$$

This result indicates that the interaction is moderately prioritized, which is primarily due to its intermediate relevance. While the interaction is no longer recent and has not yet reached long-term retention, its position within the Intermediate memory range makes it valuable for retrieval. This example demonstrates how the fuzzy weight calculation dynamically captures the relevance of an interaction over time, ensuring that the system prioritizes information based on its cognitive significance and time-decayed importance.

3.2. Schema Manager

The Schema Manager is tasked with organizing the retrieved memories into thematic clusters, known as schemas, to enrich contextual understanding and provide a structured framework for the learning experience. This process builds upon the output of the Fuzzy Memory Module, which delivers a prioritized set of high-weighted memories based on their time-decayed relevance. In this manner, using these prioritized memories, the Schema Manager maximizes the interactions of higher interest, grouping them in the thematic coherent clusters embodying the relationships and context among related pieces of knowledge. Organization in this manner enables the system to return responses not only relevant to the immediate question but also grounded in a more profound contextual understanding.

Schema formation begins with the retrieval of the top N memories from the Fuzzy Memory Module. These pre-fuzzy-weighted memories are the input to the Schema Manager's organization process. To identify relationships among the retrieved memories, the system calculates pairwise similarities using cosine similarity. This is a standard metric in natural language processing that measures the semantic closeness of two vectors. For two memory embeddings v_i and v_j, cosine similarity is calculated as

$$S_{ij} = \frac{v_i \cdot v_j}{||v_i||\,||v_j||}$$

where S_{ij} represents the similarity score, and v_i and v_j are the embedding vectors representing the semantic content of memories i and j [50]. A high similarity score of two memories will thus reflect that these are sharing either contextual or thematic overlap.

With these similarity scores, the system clusters memories into schemas by applying a preset likeness threshold. For example, if $S_{ij} > 0.8$, the memories i and j are contextually similar, so they are akin to being grouped within the same schema. This threshold limits the number of nearly related memories into a single group but keeps the thematic integrity within each schema. For example, interactions involving "for-loops", "while-loops", and "arrays" lead to a "Control Structures Schema", which is a coherent group of programming concepts. This type of clustering provides the learners with more detailed and contextually appropriate answers since it allows the system to answer not only the query but also the wider thematic associations surrounding the query.

The connection between the Fuzzy Memory Module and the Schema Manager is integral to the system's adaptive capabilities. The Fuzzy Memory Module ensures that the most relevant interactions are prioritized, reflecting the learner's recent activities and needs. The Schema Manager builds on this foundation by structuring these prioritized memories into logical groups, enabling a deeper understanding of the relationships between various interactions. This collaborative process of the system must ensure that such a system stays adaptable to what is immediately being needed while holding a robust framework for contextual learning. With regard to memory retention and thematic organization, the system allows for the rendition of personalized and precise responses that include contextual enrichment.

In the system, thematic schemas are created on the fly based on semantic similarity and hierarchical clustering, so that related interactions are more likely to be grouped into coherent knowledge structures. Although it first provides a similarity score to evaluate how semantically close two interactions are to each other (measured with cosine similarity), schema boundaries are ultimately defined through agglomerative hierarchical clustering that merges interactions into schemas using a constantly shifting similarity threshold. It is important to note that this threshold is adaptive: whether or not a new schema is formed is based on user interaction that does not have predetermined category boundaries. Schema names, like the Control Structures Schema, are not assigned manually; this is a hybrid automated generation process. First, clusters are assigned a label according to the most common keywords within the interactions in the cluster. The deep-learning based topic modeling labels are then optimized with embedding-based topic modeling so as to represent broad pedagogical resonances. Crucially, the system is not built upon injected seed schemas; rather, global thematic structures arise iteratively over time as interactions build up. Such a formation of specific schemata remains adaptive to distributed, scalable, and generalizable systems that together form capabilities beyond any pre-specified domain of action. We can also consider improvements in the future, such as reinforcement learning and user feedback systems, to enhance both schema clustering and boundary definitions.

3.3. Response Generator

The Response Generator is the final system module and is tasked with integrating inputs from both the Schema Manager and the Fuzzy Memory Module to produce contextually rich and personalized responses. Its main role is to transform structured data into coherent, pedagogically effective outputs tailored to the user's query. This process involves integrating prioritized memories, thematic clusters, and explicit guidance to ChatGPT, ensuring the responses are accurate and consistent with the user's learning history.

The Response Generator begins by gathering the relevant inputs:

- The Fuzzy Memory Module provides a ranked list of memories with fuzzy weights, representing their relevance and recency. Each memory m_i is associated with a fuzzy weight $M(m_i,t)$, which is computed using membership functions for Recent, Intermediate, and Forgotten states.
- The Schema Manager delivers the thematic context by organizing these memories into a selected Schema, S_j, which encodes related interactions and broader knowledge structures.

To ensure the response is thematically aligned, the query is first matched to the most relevant schema. This involves calculating the semantic similarity between the query embedding Q and the centroid embedding S_j of each schema. The similarity is calculated using cosine similarity:

$$Similarity(Q,\ S_j) = \frac{Q \cdot S_j}{||Q||\,||S_j||}$$

where Q is the query vector, and S_j is the centroid vector representing the average embedding of the memories within schema S_j [51]. A predetermined threshold τ ($\tau = 0.8$) determines whether a schema is relevant. The choice for $\tau = 0.8$ balances precision and recall, ensuring that only highly relevant schemas are selected while excluding irrelevant or weakly related ones. Only schemas with similarity scores exceeding this threshold are considered.

Once the most relevant schema is identified, the Response Generator retrieves individual memories within that schema, prioritizing those with higher fuzzy weights. The relevance of a memory m_i is calculated by combining its fuzzy weight $M(m_i,t)$ with its semantic similarity to the query:

$$Relevance\ Score(m_i) = \alpha \cdot M(m_i, t) + \beta \cdot Similarity(Q,\ m_i)$$

where the following apply:

- α and β are scaling factors that balance the influence of the fuzzy weight and similarity.
- $Similarity(Q,m_i)$ is the cosine similarity between the query embedding Q and the memory embedding m_i.

Memories are then ranked based on their relevance scores with the highest-ranked memories selected for the response. To ensure diversity and avoid redundancy, the system applies a penalty to memories with high pairwise similarity:

$$Penalty(m_i,\ m_j) = \max(0,\ Similarity(m_i,\ m_j) - \delta)$$

where δ is a threshold ($\delta = 0.9$) to ensure content diversity. We apply $\delta = 0.9$, because that can actually diversify the contents while penalizing only the memories that have a very high pairwise similarity.

After selecting the relevant memories and thematic context, the Response Generator structures the inputs into a formatted prompt for the generative AI model. The prompt typically includes the following:

- The user's query, explicitly stated.
- Annotated memories, such as

  "Related memory: Using for-loops with arrays."
  "Related memory: Nested loops in Java."

- Thematic insights from the matched schema, framing the response within the broader context.
- Explicit instructions for the AI model in terms of level of analysis of the related memory.

For example, if the query is *"How do I iterate over an array in Java?"*, the input to the AI would be the query, memories about loops, and thematic content on control structures with an instruction to focus on for-loops and provide a concrete example.

The hyperparameters τ, α, β, and δ were set by means of empirical tuning and ablation studies, while grid search optimization was applied for balancing memorization, schematization, retrieval accuracy and computational efficiency, respectively. Regarding memory decay factor (τ), a parameter for balancing the effect of old interactions where lower values keep information for a longer time, $\tau = 0.85$ was chosen to focus on more relevant interactions without focusing too much on old data. Now based on value of α (schema similarity threshold), the thematic clusters could be slight or too specific; the best tradeoff between specificity and coverage is for $\alpha = 0.7$, limiting excess fragmentation and schemas being too broad at the same time. The response weighting factor β balances between the old knowledge and the recent user demands with $\beta = 0.6$ being the optimal setting that maintains coherence whilst not emphasizing old responses too much. Here, δ (retrieval pruning threshold) filters low-relevance interactions, maintaining $\delta = 0.5$ for a limiting task time complexity with enough meaningful data content. All parameter ranges were based on previous work performed on contextual retrieval models and iteratively refined by testing alternative values ($\tau = \{0.7, 0.85, 0.95\}$, $\alpha = \{0.6, 0.7, 0.8\}$ as functions of precision, coherence, and schema stability). Although the present system provides hyperparameters that are fixed and tuned to maximize performance, future work could also implement adaptive tuning mechanisms, in which a dynamic adjustment of hyperparameters that are most effective for real-time interaction patterns can enhance flexibility and personalization even further.

As follows, the combined operation of the three modules is presented in a pseudocode (Algorithm 1) that shows how a user query will be processed through the Fuzzy Memory Module, Schema Manager, and Response Generator. Such details are going to define the overall process involved in the retrieval of pertinent memories, their organization into thematic schemas, and creation of enriched input to be used by ChatGPT for producing pertinent, contextually relevant responses.

Suppose a user asks "How do I iterate over an array in Java"? The query is matched to the "Control Structures Schema" with a similarity score of 0.85, exceeding the threshold τ. The Response Generator retrieves the top memories within this Schema, such as the following:

- Memory m_1: "Using for-loops with arrays" ($M(m_1,t) = 0.8$).
- Memory m_2: "Difference between for-loops and while-loops" ($M(m_2,t) = 0.5$). These memories are combined with thematic content from the schema and passed to the generative AI model, which produces a response like the following: "To iterate over an array in Java, you can use a for-loop. Here's an example: for (int i = 0; i < array.length; i++) {System.out.println(array[i]);}. This builds on what we discussed earlier about

for-loops and arrays."

Algorithm 1. GenerateResponse(UserQueryContent UQC)

1. Retrieve relevant memories:
   - For each memory in MemoryRepository:
     - Compute FuzzyWeight based on memory timestamp
     - Compute SimilarityScore between memory and UQC
     - Compute RelevanceScore using weighted combination of FuzzyWeight and SimilarityScore
     - If RelevanceScore exceeds threshold:
       -Add memory to RelevantMemories
2. Apply redundancy penalty:
   - For each memory pair (Mi, Mj) in RelevantMemories:
     - If similarity between Mi and Mj exceeds DiversityThreshold:
       - Reduce relevance score of Mj by a penalty factor
   - Rank memories by final relevance score
3. Organize ranked memories into thematic schemas:
   - For each memory in RankedMemories:
     - Assign it to the most relevant Schema
4. Select top N schemas:
   - Compute SchemaRelevance for each schema based on UQC
   - Select the top N schemas with the highest relevance scores
5. Construct structured input for ChatGPT:
   - Initialize InputToChatGPT with user query
   - For each selected schema:
     - Append schema name and relevant memories
     - Determine AnalysisLevel (Detailed or Brief) for each memory
     - If Detailed, append full memory content
     - If Brief, append summarized memory content
6. Generate final response:
   - Call ChatGPT with structured input
   - Return generated response

4. Results

This section presents the evaluation of the presented system, focusing on its performance in memory retrieval, schema relevance, and response quality. The evaluation employs quantitative metrics, qualitative assessments, and comparisons with baseline systems for confirming the proposed method's effectiveness. The study involved 120 undergraduate students interacting with the system over a four-week period, generating extensive data for analysis.

4.1. Evaluation Metrics

In this subsection, the evaluation metrics [52,53] that have been used are presented.

4.1.1. Memory Retrieval Accuracy

The Fuzzy Memory Module was assessed based on its ability to retrieve relevant memories efficiently. Two metrics were employed:

- Precision@N: This evaluates the proportion of relevant memories among the top-N retrieved. For instance, Precision@5 computes the fraction of relevant memories in the top five results. An improved Precision@5 measure indicates that the system is ranking relevant memories first.

- Mean Reciprocal Rank (MRR): It measures how high the first relevant memory ranks in the ranked retrieval list. For a given query, if the first relevant memory is at rank k, the reciprocal rank is $\frac{1}{k}$. The MRR is the average reciprocal rank across all the queries, which reflects the ability of the system to retrieve relevant information at an early rank.

4.1.2. Schema Relevance

The Schema Manager was evaluated on its ability to organize and retrieve thematically related clusters. Two metrics were used:

- Cluster Purity: This metric evaluates how homogeneous memories are within a schema. Purity is greater if memories within a schema pertain closely to the identical theme, signifying correct clustering.
- Query–Schema Match Precision: This measures the accuracy with which the system selects the most relevant schema for a query. It is determined by comparing the system's selected schema with a manually annotated gold standard.

4.1.3. Response Relevance and Quality

The Response Generator was tested on its ability to produce personalized, contextually rich responses:

- BLEU Score: It measures the overlap between the system-generated responses and predefined reference answers. Improved BLEU scores indicate greater adherence to the ground truth.
- Human Evaluation: Relevance, clarity, and personalization of system responses were rated on a scale of 1 (poor) to 5 (excellent) on a Likert scale by a panel of evaluators.

4.2. Experimental Setup

The study involved 120 second-year undergraduate students (75 males, 45 females) enrolled in a Java programming course at a public university in a capital city of the country. The course included basic topics like loops, arrays, and object-oriented programming. The students who were selected represented heterogeneous academic and demographic profiles.

Students used the system over four weeks in sessions designed to reflect real educational contexts, including the following:

- Concept Reinforcement: Students asked questions to revisit previously taught material.
- Exploration of New Topics: Queries were designed to extend knowledge, introducing new material related to past concepts.
- Feedback and Review: The system provided explanations, feedback on assignments, and guidance for further study.

The students received an average of 20 sessions, containing an average of 10 questions per session, for a total of 2000 interactions that were recorded.

The dataset used for the evaluation was designed to align with the Java programming curriculum and included the following:

- 1500 unique questions covering Java subjects such as loops, arrays, and object-oriented principles.
- 3000 interaction logs recording varied student inquiries, extending from basic syntax to conceptual rationale and code bits.
- Annotations: All sessions were annotated with timestamps, query details, and thematic labels to evaluate memory retention and schema construction.

To balance the dataset, half of the queries were directed toward reinforcing previously taught material, while the other half explored new concepts connected to earlier concepts. Semantic embeddings for queries and memories were precomputed using a transformer-based model to facilitate similarity calculations.

The system was compared against two baseline approaches:

- Baseline 1 (Keyword-Based Retrieval): A simple retrieval system that matches queries to stored memories based on keyword overlap without considering fuzzy weights or thematic clustering.
- Baseline 2 (ChatGPT-Only): ChatGPT that generates responses based solely on the user query, without leveraging memory prioritization or schema organization.

Forty students (25 males and 15 females) interacted with each one of the three systems (presented system, Baseline 1 and Baseline 2).

4.3. Results Analysis

4.3.1. Results for Memory Retrieval Accuracy

The performance of the Fuzzy Memory Module was quantified using Precision@5 and MRR. The results are presented in Table 1.

Table 1. Precision@5 and MRR results.

Metric	Presented System	Baseline 1 (Keyword-Based)	Baseline 2 (Chat-GPT)
Precision@5	88%	63%	71%
Mean Reciprocal Rank (MRR)	0.81	0.58	0.66

The Precision@5 value was calculated by taking the proportion of relevant memories retrieved within the top 5 ranked results for each query. To accomplish this, we manually labeled the relevance of memories in the dataset based on their overall correspondence to the user query and overall context. For a particular query, if five memories were retrieved and four of them were relevant, the Precision@5 score for that query was $\frac{4}{5} = 0.8$. The final Precision@5 value of 88% for the proposed system is the average across all queries, which was calculated as

$$Precision@5 = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{Number\ of\ relevant\ memories\ in\ Top\ 5}{5}$$

where $|Q|$ is the total number of queries evaluated.

The MRR measures how early in the ranked list the first relevant memory appears. For each query, if the first relevant memory is at rank k, the reciprocal rank is $\frac{1}{k}$. For example, if the first relevant memory for a query appears at rank 2, the reciprocal rank is $\frac{1}{2} = 0.5$. The final MRR value of 0.81 was calculated by averaging the reciprocal ranks across all queries:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{k_i}$$

where k_i is the rank of the first relevant memory for query i.

The proposed system outperformed both baselines significantly. The Precision@5 score of 88% indicates that nearly nine out of ten retrieved memories were relevant, reflecting the effectiveness of fuzzy weights in prioritizing recent and contextually appropriate interactions. The MRR score of 0.81 further highlights the system's ability to rank relevant memories near the top of the list.

Baseline 1 could not retrieve relevant memories since it relied on exact keyword matches and overlooked semantic nuances. Baseline 2 performed better, but without fuzzy weights, it was less effective in prioritizing recent interactions, which resulted in lower precision and MRR.

The high Precision@5 and MRR values indicate that the Fuzzy Memory Module effectively prioritizes recent and contextually relevant interactions, outperforming the baselines. Baseline 1 performed poorly in Precision@5 and MRR since it relied on exact keyword matching, which often failed to capture semantic nuances. Baseline 2 performed moderately better but lacked the prioritization provided by fuzzy weights, thus scoring lower.

4.3.2. Results for Schema Relevance

The Schema Manager's performance in clustering and selecting thematic groups was evaluated using cluster purity and query–schema match precision. Table 2 summarizes the results.

Table 2. Cluster purity and query–schema match precision results.

Metric	Presented System	Baseline 1 (Keyword-Based)	Baseline 2 (Chat-GPT)
Cluster Purity	0.91	0.68	0.75
Query–Schema Match Precision	85%	57%	70%

The cluster purity metric measures the thematic coherence of memories within each schema. For each cluster, purity was calculated as the fraction of memories that belong to the dominant theme within that cluster. For example, if a schema contained 10 memories and eight of them were correctly labeled as "Control Structures", the cluster purity for that schema would be $\frac{8}{10} = 0.8$. The overall purity score of 0.91 for the proposed system represents the weighted average of purity scores across all schemas, which was calculated as

$$Cluster\ Purity = \frac{\sum_{j=1}^{|S|} n_j \cdot Purity\ of\ Schema\ S_j}{\sum_{j=1}^{|S|} n_j}$$

where $|S|$ is the total number of schemas, and n_j is the number of memories in Schema S_j.

The query–schema match precision evaluates how often the Schema Manager selected the most relevant schema for a query. Relevance was determined by comparing the selected schema against a manually labeled ground truth. For example, if a query like "How do I use nested loops?" was matched to the "Control Structures Schema", and this schema was labeled as correct, it contributed to the precision. The overall precision of 85% was calculated as

$$Query - Schema\ Match\ Precision = \frac{Number\ of\ Correctly\ Matched\ Schemas}{Total\ Queries}$$

The proposed system demonstrated strong clustering capabilities, achieving a cluster purity score of 0.91. This indicates that the memories grouped within each schema were highly thematically coherent. The query–schema match precision of 85% reflects the system's ability to accurately match queries to the most relevant schema, enabling contextually enriched responses.

Baseline 1 failed to organize memories into meaningful clusters due to its lack of semantic understanding, resulting in poor cluster purity and match precision. Baseline 2 performed moderately well but lacked the structured thematic grouping provided by the Schema Manager.

The results demonstrate the Schema Manager's superior clustering and matching capabilities. Baseline 1 performed poorly due to its inability to group memories thematically, while Baseline 2 achieved moderate performance but lacked the structured organization provided by schemas.

4.3.3. Results for Response Relevance and Quality

The responses generated by the system were evaluated using BLEU scores and human ratings. Table 3 presents the results.

Table 3. BLEU scores and human ratings.

Metric	Presented System	Baseline 1 (Keyword-Based)	Baseline 2 (ChatGPT)
BLEU Score	0.79	0.53	0.67
Relevance (Human Score)	4.6/5	3.8/5	4.2/5
Clarity (Human Score)	4.5/5	3.7/5	4.0/5
Personalization (Human-Score)	4.4/5	3.5/5	3.9/5

The BLEU score measures the n-gram overlap between system-generated responses and reference responses. For each response, BLEU was computed using the following formula:

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n log p_n\right)$$

where BP is the brevity penalty, w_n is the weight for n-grams, and p_n is the precision for n-grams of size n. The BLEU score of 0.79 for the proposed system reflects a high level of similarity between generated responses and ground-truth answers. The lower BLEU scores for the baselines indicate that their responses were less aligned with the reference answers.

Human evaluation involved ten independent evaluators, all of whom are professors in the field of AI in education with over 25 years of experience. They rated responses on relevance, clarity, and personalization using a 5-point Likert scale. The evaluators have the user query, the response generated, and the context of the interaction to ensure a complete and expert assessment of the output by the system.

Ratings for relevance judged the level at which the response was able to address the query, whereas clarity referred to ease of understanding. Personalization evaluated to what extent a response was similar to what had been generated in the user's past learning history. The proposed system consistently obtained top ratings, since fuzzy weights and thematic clustering together generated responses fitting a user's context.

The proposed system achieved the highest BLEU score (0.79), indicating strong alignment with reference responses. Human evaluators praised the system's ability to incorporate prior context and tailor responses to the user's learning history, awarding average scores of 4.6/5 for relevance, 4.5/5 for clarity, and 4.4/5 for personalization.

Baseline 1 produced generic responses that lacked coherence and contextual depth, while Baseline 2 delivered more relevant answers but failed to personalize them effectively due to its reliance on query-only input. That means that Baseline 1 performed poorly across all metrics due to its generic and often irrelevant responses. Baseline 2 achieved moderate performance but failed to deliver personalized outputs, as it relied solely on the user query without considering prior interactions or thematic context.

5. Discussion

Through the evaluation of the proposed system, it will be able to prove that the latter has major contributions to personalized education technology. Apart from showing that it can generate responses which are relevant and tailor-made for the students' needs, the result can also confirm its design features, combining fuzzy weights with memory retention and thematic clustering in order to understand the context better. Further discussions on the implication of the results compared to existing results will provide knowledge about the deficit of the proposed system and would provide good avenues for future improvements.

The comparison results shown in Tables 1–3 prove the helpfulness of our system compared to generative AI models, especially ChatGPT-only tutoring (Baseline 2). Although generative AI generates fluent and syntactically correct responses, it lacks context remembering, personalized feedback and the structured retrieval of knowledge. In Table 1, we observe that the ChatGPT-only model achieves a Precision@5 of only 71%, which is much lower than our system (88%), indicating the limitations of LLMs as relevance rankers. In a similar vein, the MRR of 0.66 for the ChatGPT-centric tutoring indicates that the model struggled to retrieve the best prior interactions early in the ranking, in contrast with our system's 0.81 MRR, which profited from fuzzy memory ranking.

Schema-based thematic clustering significantly boosts the precision of query–schema match at 85% (as also demonstrated in Table 2), while generative AI models tend to return semantically relevant but contextually unrelated responses with only 70% match precision. Because generative AI depends on only the current prompt context with no way to structure longer-term memory, related concepts tend to clump less coherently together, resulting in declining response consistency across multiple interactions. The BLU score showing ChatGPT-only tutoring (0.67) was indeed competitive but lower than what had been obtained by our system (0.79, Table 3), which suggests that structured memory retrieval is advantageous in improving response relevance. In fact, human evaluation ratings show that generative AI models yield fluent but less personalized outputs scoring lower for clarity (4.0/5) and personalization (3.9/5), whereas their respective scores for our system are 4.5 and 4.4.

These findings illustrate that even though generative AI has the potential to produce very linguistically competent written outputs, it is highly unlikely to be able to dynamically manage the memory processes of prioritization, theme-based arrangement of conversation, and delivery of long-term contextual continuity. Through fuzzy retention and schema-based retrieval, the system is able to partially solve these problems and thus be more adaptive, structured, and context-aware than LLMs on their own, which increases the chances of true AI tutoring.

One of the main cruxes under the system encompasses the Fuzzy Memory Module through which it boasts high memory access accuracy. On coming across the precision of 88% and mean reciprocal rank score of 0.81, it clearly notified that in this module lies a highly relevant interaction on the time-decayed relevance. These results, consistent with some previous research on memory networks, such as [54], indicate the need for memory mechanisms that keep contextual relevance. However, unlike static approaches relying on either fixed weights or predefined rules, the fuzzy weights used here keep relevance dynamic by changing the interactions stored, and hence they are more similar to human forgetting and prioritization. This dynamic nature is particularly useful in educational settings, where the more recent queries frequently mirror changing learning needs. In contrast to more basic keyword-based systems or generative AI models without memory mechanisms, this method significantly enhances both the accuracy and contextual relevance of retrieved memories.

The Schema Manager builds on top of this foundation by structuring retrieved memories into thematic clusters and brings structure and depth to the learning process. Indeed, this can be confirmed by the elaborate results with a cluster purity of 0.91 and a query–schema match precision of 85%, which proves that the manager can expect effectiveness in the way it groups related concepts. These metrics illustrate that the system excels at capturing the relationships between memories, enabling it to deliver responses that are not only accurate but also contextually enriched. This thematic clustering is particularly relevant in programming education, where foundational concepts like loops and arrays are often interconnected with more advanced topics such as object-oriented programming. In that, the system ensures that responses are relevant to both the immediate query and its broader context by creating cohesive schemas. The approach also extends beyond the state-of-the-art in traditional conversational agents because it generalizes to those that do not have contextual organization mechanisms and actually strengthens findings from past research, which underlined the need for thematic understanding in dialogue systems.

Finally, in the Response Generator module, everything above is put together by generating custom responses based on prioritized memories and thematic context. Its quality is reflected through a BLEU score of 0.79, which is very high, meaning that almost all the returned responses are correctly aligned with reference responses predefined within the system. Furthermore, human judges continuously scored the system highly regarding relevance, clarity, and personalization, obtaining average scores of 4.6, 4.5, and 4.4 points out of 5, respectively. These experimental results highlight the effectiveness of the system at personalizing response generation based on prior interactions with learners and thematic knowledge from these interactions. Compared to baseline systems, the Response Generator produced richer outputs more indicative of its understanding about the user's learning trajectory. While simpler retrieval-based systems tend to produce generic or disjointed answers, and standard generative AI models fail to properly use historical context, this system's structured approach guarantees that the answers are both precise and meaningful.

This study's findings are thus in line with and extended from existing literature about conversational agents and personalized learning systems. For instance, in [55], the authors have pointed out that generative AI models lack coherence over long periods without explicit memory mechanisms. The proposed system addresses the gap of this sort with fuzzy weights on integrating dynamic memory prioritization. Similarly, in [56], the authors analyzed the direction of thematic clustering on knowledge graphs, but the focus of their work was static representation, whereas in this system, the Schema Manager dynamically updates clusters with evolving user interactions, thereby creating an adaptive and personalized framework. The findings also extend previous work [57], which showed the significance of contextual cues in dialogue systems but did not have a mechanism for organizing memories into thematic groups. The proposed system bridges these gaps by combining fuzzy memory retention and schema-based clustering, offering a novel approach to delivering personalized and contextually rich responses.

Our study makes a significant contribution to the research landscape by addressing gaps in the existing literature and offering practical, impactful advancements. Building upon the foundational work in [58], which explores the theoretical potential of fuzzy logic in customizing educational content, our research extends these findings by integrating dynamic memory retention and contextual organization. While the previous study laid the groundwork, it did not incorporate fuzzy memory retention or contextual organization. In contrast, our work fills this gap. The Fuzzy Memory Module dynamically prioritizes interactions using time-decayed fuzzy weights, while the Schema Manager organizes these interactions into thematic clusters. This enables ChatGPT to provide contextually enriched and personalized responses. Through these practical innovations, our study not only validates but also extends the theoretical insights of prior research, presenting a scalable and effective solution for adaptive learning in programming education.

The results of the study also focus on the capability of the system in resolving some major problems of educational technology. It is to reinforce previously taught concepts while teaching new material in a related context. Since recent and relevant interactions are the theme of the Fuzzy Memory Module, responses will reinforce prior knowledge. At the same time, the Schema Manager promotes novelty by embedding new information within larger thematic frameworks. If a user asks how to go through an array in Java, the system retrieves memories of for-loops and arrays, also relating them to other topics, such as loops inside loops. This ability to balance familiarity with novelty is key to creating effective learning.

Another strength of the system is personalization capacity. The system attempts to address some of the constraints of the traditional one-size-fits-all approaches that are typical for education. For example, the interaction history of learners with the system forms the basis of personalizing responses that are delivered. This is obvious in the results of human evaluation, where all the responses elicited very high ratings on both relevance and clarity. The reviewers pointed out that the system could refer back to previously taught concepts and adapt to the user's progress, which was a major differentiator. This is in line with the principles of adaptive learning, which stresses the importance of delivering content that meets learners where they are in their educational journey.

Despite these strengths, there are a number of weaknesses the system could further be researched to overcome. The interpretation of vague queries is a big challenge. Sometimes ambiguous queries may cause schema selection not to be optimal. For instance, "How can I optimize my code?" is an ambiguous query, which could link to multiple schemas. Those schemas include control structures, performance optimization, or even debugging. In such scenarios, the system may not always pick the most relevant schema based on semantic similarity scores. This can be addressed by including other disambiguation techniques, such as user feedback or multi-modal inputs. Another limitation is the dependence on predefined thresholds for similarity scores and diversity penalties. These thresholds were empirically optimized for this study but may not generalize across all educational contexts.

In programming education, where concepts often build on one another, this system offers a practical solution for helping learners navigate complex topics while maintaining continuity with their prior learning. Comparisons with existing systems further underscore the novelty and effective-ness of the proposed approach. Traditional models of conversation, including but not limited to a MemN2N model given by [59], have fixed memory in the sense it cannot adjust easily according to new dialogues coming while using a generation AI like that of GPT-3 who performs wonderfully creating fluent text but may go wrong on placing the historical context on its rightful occasions.

Recent advances in prompt engineering and in-context learning [60] have notably increased the inferential capabilities of LLMs, so they can deduce new tasks from a little context example without retraining prompts. However, while prompt engineering improves the LLM's output, our approach enhances in-context learning with structured memory hold and schema-guided organization. This enables retrievals to create coherent or pedagogically relevant responses rather than disjointed ones and turned out of order. The stark contrast between the traffic signal follower LLM and our scheme is that it uses fuzzy memory retention in time-restricted need circumstances to dynamically adjust what past interactions are considered most important, thereby lessening its dependence on prompts which are externally optimized. This in itself is consistent with evidence showing LLMs' emergent reasoning capabilities and suggests that through combining structured retrieval with optimally designed prompts, we may further improve the response consistency and adaptability of these models in future work. Future work might take adaptive prompt generation techniques as its entrance to dynamically generate optimal instructional guidance through real-time schema clustering and will offer users a personalized choice out of many LLM outputs.

Although the system proposed here personalizes based on the history of a learner, it is actually designed to prevent stagnation by means of thematic schema expansion and adaptive memory retrieval. While recommender systems are limited to previous topics, since both elements are a common area of knowledge, the Fuzzy Memory Module emphasizes experience mostly based on freshness and relevance but also adds a degree of diversity by changing memory weights on the fly to stimulate the exploration of new areas but still topical knowledge. In addition, the Schema Manager not only allows learners to unpack their learning history, but it also organizes interactions into greater thematic bundles, enabling them to follow topics that may be more advanced than the immediate history but related in thematic context. This helps stop the reinforcement of narrow content loops and encourages a broadening of knowledge. Future improvements could build on this by introducing novelty-based content suggestions, promoting stability but mixing it up in their learning journey.

The proposed system addresses these shortcomings by integrating fuzzy weights and thematic clustering into a unified architecture, enabling it to deliver responses that are both personalized and contextually enriched. The findings also highlight opportunities for future development. One promising direction is the incorporation of multi-modal inputs, such as voice queries, diagrams, or code files. This would allow the system to support a wider range of educational scenarios and enhance its applicability across diverse learning environments. Another area for exploration is the integration of real-time user feedback into the memory prioritization and schema selection processes.

This would let the system dynamically adjust its weighting and clustering mechanisms based on user feedback regarding whether a response was helpful or relevant. In conclusion, the proposed system represents an important advance in personalized educational technology. Its ability to prioritize relevant interactions, organize knowledge into thematic schemas, and deliver personalized responses positions it as a powerful tool for programming education and beyond. The results validate its potential to change the learning experience by providing the right, contextual support at appropriate times and adjusted to the user's needs. While some of the limitations have not been mitigated, this study provides an excellent foundation for future studies, offering the value of its insights into the design and evaluation of adaptive learning systems.

6. Conclusions

This work introduces a new framework for personalized educational AI, integrating fuzzy memory retention, thematic clustering, and generative AI to address important challenges in adaptive learning. Combining the strengths of three collaborative modules—the Fuzzy Memory Module, Schema Manager, and Response Generator—equipped with the generative power of ChatGPT, the system offers customized, contextually accurate responses that can dynamically adapt to the needs of each learner. It integrates fuzzy weights to ensure that recent and relevant interactions are prioritized, while the Schema Manager enhances contextual understanding by organizing memories into thematic clusters. The Response Generator synthesizes these inputs to guide ChatGPT to generate responses that reinforce prior knowledge, introduce new material, and align with the learner's evolving trajectory.

The system was evaluated in a real-world educational context, demonstrating significant improvements in memory retrieval accuracy, schema relevance, and response personalization compared to traditional methods and baseline models. These results validate the framework's effectiveness in delivering an engaging and effective learning experience, particularly in programming education. The combination of human-like memory retention and thematic knowledge organization sets a new standard for adaptive educational technologies, highlighting the potential for broader applications in diverse learning domains.

This study shows how well the proposed system works for adaptive tutoring and the adaptation of memory, but it has some limitations. The first limitation is that the evaluation was conducted only in this study domain (programming education), which can restrict generalizability across subjects with different cognitive structures. Second, even if the system implements some fuzzy memory maintenance rules and schema clustering, its personalization approach does not currently include long-term adaptive modifications of learner behavior that can help adjust knowledge dissemination through diversification and numbers. Third, although we had applied empirical tuning on parameters like α, β, and δ, a completely automated and adaptive optimization mechanism (for example reinforcement learning or genetic algorithms) has not been carried out.

Future work will be aimed at overcoming these limitations by scaling the system up to larger datasets and different educational domains (e.g., mathematics, language learning) and by providing domain-specific knowledge to make the system more versatile. Improvements will consist of adding multi-modal inputs (e.g., voice queries, diagrams) to enhance interpretation through richer representations of learner interactions, testing dynamic thresholding methods to stimulate the robustness of memory prioritization and schema clustering, and building real-time feedback systems for personalization optimization. Long-term tracking studies will be rolled out that will push the system to check how knowledge is retained and whether the learners remain engaged for longer periods which will eventually lead to developing an even advanced adaptive learning program.

Author Contributions: Conceptualization, C.T. and A.K.; methodology, C.T. and A.K.; software, C.T. and A.K.; validation, C.T. and A.K.; formal analysis, C.T. and A.K.; investigation, C.T. and A.K.; resources, C.T. and A.K.; data curation, C.T. and A.K.; writing—original draft preparation, C.T., A.K., P.M., C.S. and I.V.; writing—review and editing, C.T. and A.K.; visualization, C.T. and A.K.; supervision, C.T. and A.K.; project administration, C.T., A.K., P.M., C.S. and I.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval are not required for this study, as it exclusively involves the analysis of properly anonymized datasets obtained from past research studies through voluntary participation. This research does not pose a risk of harm to the subjects. All data were handled with the utmost confidentiality and in compliance with ethical standards.

Informed Consent Statement: Informed consent was obtained from all subjects at the time of original data collection.

Data Availability Statement: The data supporting the findings of this study are available upon request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xia, Y.; Shin, S.-Y.; Shin, K.-S. Designing Personalized Learning Paths for Foreign Language Acquisition Using Big Data: Theoretical and Empirical Analysis. *Appl. Sci.* **2024**, *14*, 9506. [CrossRef]
2. du Plooy, E.; Casteleijn, D.; Franzsen, D. Personalized adaptive learning in higher education: A scoping review of key characteristics and impact on academic performance and engagement. *Heliyon* **2024**, *10*, e39630. [CrossRef] [PubMed]
3. García, V. Analysis of the Learning Process of Computer Programming Logic in an 8-Year-Old Elementary School Student at Home through the Scratch Program. *Digital* **2024**, *4*, 69–91. [CrossRef]
4. Barbosa, P.L.S.; Carmo, R.A.F.D.; Gomes, J.P.; Viana, W. Adaptive learning in computer science education: A scoping review. *Educ. Inf. Technol.* **2024**, *29*, 9139–9188. [CrossRef]

5. Buschman, T.J.; Miller, E.K. Working Memory Is Complex and Dynamic, Like Your Thoughts. *J. Cogn. Neurosci.* **2022**, *35*, 17–23. [CrossRef]
6. Martin, F.; Chen, Y.; Moore, R.L.; Westine, C.D. Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018. *Educ. Technol. Res. Dev.* **2020**, *68*, 1903–1929. [CrossRef]
7. Zhai, C.; Wibowo, S.; Li, L.D. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: A systematic review. *Smart Learn. Environ.* **2024**, *11*, 28. [CrossRef]
8. Frick, T.W. Are We Dupes? Limitations of AI Systems: What Should Educators Do with Them? *TechTrends* **2024**, *68*, 14–26. [CrossRef]
9. Adewale, M.D.; Azeta, A.; Abayomi-Alli, A.; Sambo-Magaji, A. Impact of artificial intelligence adoption on students' academic performance in open and distance learning: A systematic literature review. *Heliyon* **2024**, *10*, e40025. [CrossRef]
10. Chen, L.; Chen, P.; Lin, Z. Artificial Intelligence in Education: A Review. *IEEE Access* **2020**, *8*, 75264–75278. [CrossRef]
11. Hilali, K.; Chergui, M.; Ammoumou, A. Adaptive Learning Systems: A Comprehensive Overview and Identification of Challenges. In Proceedings of the 2023 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), Rabat, Morocco, 22–24 November 2023; pp. 1–7. [CrossRef]
12. Kusal, S.; Patil, S.; Choudrie, J.; Kotecha, K.; Mishra, S.; Abraham, A. AI-Based Conversational Agents: A Scoping Review From Technologies to Future Directions. *IEEE Access* **2022**, *10*, 92337–92356. [CrossRef]
13. Jayram, T.S.; Bouhadjar, Y.; McAvoy, R.L.; Kornuta, T.; Asseman, A.; Rocki, K.; Ozcan, A.S. Learning to remember, forget and ignore using attention control in memory. *arXiv* **2018**, arXiv:1809.11087.
14. Zhang, Z.; Huang, M.; Zhao, Z.; Ji, F.; Chen, H.; Zhu, X. Memory-Augmented Dialogue Management for Task-Oriented Dialogue Systems. *ACM Trans. Inf. Syst.* **2019**, *37*, 34. [CrossRef]
15. Troussas, C.; Krouska, A.; Virvou, M. Integrating an Adjusted Conversational Agent into a Mobile-Assisted Language Learning Application. In Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, USA, 6–8 November 2017; pp. 1153–1157. [CrossRef]
16. Porcu, V. The Role of Memory in LLMs: Persistent Context for Smarter Conversations. *Int. J. Sci. Res. Manag. (IJSRM)* **2024**, *12*, 1673–1691. [CrossRef]
17. Dergaa, I.; Ben Saad, H.; Glenn, J.M.; Amamou, B.; Ben Aissa, M.; Guelmami, N.; Fekih-Romdhane, F.; Chamari, K. From tools to threats: A reflection on the impact of artificial-intelligence chatbots on cognitive health. *Front. Psychol.* **2024**, *15*, 1259845. [CrossRef] [PubMed]
18. Zheng, S.; He, K.; Yang, L.; Xiong, J. Memory Repository for AI NPC. *IEEE Access* **2024**, *12*, 62581–62596. [CrossRef]
19. Florindi, F.; Fedele, P.; Dimitri, G.M. A novel solution for the development of a sentimental analysis chatbot integrating ChatGPT. *Pers. Ubiquitous Comput.* **2024**, *28*, 947–960. [CrossRef]
20. Aleven, V. Rule-Based Cognitive Modeling for Intelligent Tutoring Systems. In *Advances in Intelligent Tutoring Systems*; Nkambou, R., Bourdeau, J., Mizoguchi, R., Eds.; Studies in Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2010; Volume 308, pp. 33–62. [CrossRef]
21. Bai, X.; Black, J.B. Enhancing Intelligent Tutoring Systems with the Agent Paradigm. In *Intelligent Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation*; Stankov, S., Glavinic, V., Rosic, M., Eds.; IGI Global Scientific Publishing: Hershey, PA, USA, 2011; pp. 46–69. [CrossRef]
22. Batsaikhan, B.; Correia, A.-P. The effects of Generative Artificial Intelligence on Intelligent Tutoring Systems in higher education: A systematic review. *Stud. Technol. Enhanc. Learn.* **2024**, *4*, 1–19. [CrossRef]
23. Kanetaki, Z.; Stergiou, C.; Bekas, G.; Troussas, C.; Sgouropoulou, C. A Hybrid Machine Learning Model for Grade Prediction in Online Engineering Education. *Int. J. Eng. Pedag. (IJEP)* **2022**, *12*, 4–24. [CrossRef]
24. Lin, C.C.; Huang, A.Y.Q.; Lu, O.H.T. Artificial intelligence in intelligent tutoring systems toward sustainable education: A systematic review. *Smart Learn. Environ.* **2023**, *10*, 41. [CrossRef]
25. Alfarra, A.H.; Samhan, L.F.; Mosa, M.J.; Alajrami, M.A.; El Kahlout, F.; Abu-Nasser, B.S.; Abu-Naser, S.S. AI-Driven Learning: Advances and Challenges in Intelligent Tutoring Systems. *Int. J. Acad. Appl. Res. (IJAAR)* **2024**, *8*, 24–29.
26. As'ad, M. Intelligent Tutoring Systems, Generative Artificial Intelligence (AI), and Healthcare Agents: A Proof of Concept and Dual-Layer Approach. *Cureus* **2024**, *16*, e69710. [CrossRef]
27. Alam, A.; Mohanty, A. Educational technology: Exploring the convergence of technology and pedagogy through mobility, interactivity, AI, and learning tools. *Cogent Eng.* **2023**, *10*, 2283282. [CrossRef]
28. Troussas, C.; Chrysafiadi, K.; Virvou, M. Machine Learning and Fuzzy Logic Techniques for Personalized Tutoring of Foreign Languages. In Proceedings of the Artificial Intelligence in Education. AIED 2018, London, UK, 27–30 June 2018; Penstein Rosé, C., Martínez-Maldonado, R., Ulrich Hoppe, H., Luckin, R., Mavrikis, M., Porayska-Pomsta, K., McLaren, B., du Boulay, B., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 10948, pp. 358–362. [CrossRef]
29. Jan, N.U.; Naqvi, S.; Ali, Q. Using Fuzzy Logic for Monitoring Students Academic Performance in Higher Education. *Eng. Proc.* **2023**, *46*, 21. [CrossRef]

30. Hnatchuk, Y.; Hnatchuk, A.; Pityn, M.; Hlukhov, I.; Cherednichenko, O. Intelligent Decision Support Agent Based on Fuzzy Logic in Athletes' Adaptive E-Learning Systems. In Proceedings of the 2nd International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS'2021), Khmelnytskyi, Ukraine, 24–26 March 2021; Volume 2853, pp. 1–8. Available online: https://ceur-ws.org/Vol-2853/paper4.pdf (accessed on 3 March 2023).
31. Ilahi, R.; Widiaty, I.; Abdullah, A.G. Fuzzy system application in education. In Proceedings of the 3rd Annual Applied Science and Engineering Conference (AASEC 2018), Bandung, Indonesia, 18 April 2018; Volume 434, p. 012308. [CrossRef]
32. Papakostas, C.; Troussas, C.; Krouska, A.; Sgouropoulou, C. PARSAT: Fuzzy logic for adaptive spatial ability training in an augmented reality system. *Comput. Sci. Inf. Syst.* **2023**, *20*, 1389–1417. [CrossRef]
33. Göktepe Körpeoğlu, S.; Filiz, A.; Göktepe Yıldız, S. AI-Driven Predictions of Mathematical Problem-Solving Beliefs: Fuzzy Logic, Adaptive Neuro-Fuzzy Inference Systems, and Artificial Neural Networks. *Appl. Sci.* **2025**, *15*, 494. [CrossRef]
34. Alzaid, M.; Fkih, F. Sentiment Analysis of Students' Feedback on E-Learning Using a Hybrid Fuzzy Model. *Appl. Sci.* **2023**, *13*, 12956. [CrossRef]
35. Doz, D.; Cotič, M.; Felda, D. Random Forest Regression in Predicting Students' Achievements and Fuzzy Grades. *Mathematics* **2023**, *11*, 4129. [CrossRef]
36. Albaity, M.; Mahmood, T.; Ali, Z. Analysis and Applications of Artificial Intelligence in Digital Education Based on Complex Fuzzy Clustering Algorithms. *Mathematics* **2023**, *11*, 3184. [CrossRef]
37. Hegazi, M.O.; Almaslukh, B.; Siddig, K. A Fuzzy Model for Reasoning and Predicting Student's Academic Performance. *Appl. Sci.* **2023**, *13*, 5140. [CrossRef]
38. Elfakki, A.O.; Sghaier, S.; Alotaibi, A.A. An Intelligent Tool Based on Fuzzy Logic and a 3D Virtual Learning Environment for Disabled Student Academic Performance Assessment. *Appl. Sci.* **2023**, *13*, 4865. [CrossRef]
39. Salgado Granda, B.; Inzhivotkina, Y.; Ibáñez Apolo, M.F.; Ugarte Fajardo, J.G. Educational innovation: Exploring the Potential of Generative Artificial Intelligence in cognitive schema building. *Edutec Rev. Electron. Tecnol. Educ.* **2024**, *89*, 44–63. [CrossRef]
40. Misra, V.P.; Mishra, P.K.; Sharma, A. Artificial Intelligence in Education—Emerging Trends, Thematic Analysis & Application in Lifelong Learning. In Proceedings of the 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Nadi, Fiji, 4–6 December 2023; pp. 1–6. [CrossRef]
41. Vallayil, M.; Nand, P.; Yan, W.Q. Explainable AI through Thematic Clustering and Contextual Visualization: Advancing Macro-Level Explainability in AFV Systems. In Proceedings of the ACIS 2024, Canberra, Australia, 5–8 December 2024; p. 101. Available online: https://aisel.aisnet.org/acis2024/101 (accessed on 20 January 2025).
42. Jung, E.; Lim, R.; Kim, D. A Schema-Based Instructional Design Model for Self-Paced Learning Environments. *Educ. Sci.* **2022**, *12*, 271. [CrossRef]
43. Kaban, A. Artificial intelligence in education: A science mapping approach. *Int. J. Educ. Math. Sci. Technol. (IJEMST)* **2023**, *11*, 844–861. [CrossRef]
44. Delen, I.; Sen, N.; Ozudogru, F.; Biasutti, M. Understanding the Growth of Artificial Intelligence in Educational Research through Bibliometric Analysis. *Sustainability* **2024**, *16*, 6724. [CrossRef]
45. Hou, J.; Wan, J. Theme Mining and Evolutionary Analysis of Artificial Intelligence Integration in Higher Education Research. In Proceedings of the E-Business. New Challenges and Opportunities for Digital-Enabled Intelligent Future. WHICEB 2024, Wuhan, China, 24–26 May 2024; Tu, Y.P., Chi, M., Eds.; Lecture Notes in Business Information Processing; Springer: Cham, Switzerland, 2024; Volume 516, pp. 108–119. [CrossRef]
46. Ishaq, K.; Alvi, A.; Haq, M.I.; Rosdi, F.; Choudhry, A.N.; Anjum, A.; Khan, F.A. Level up your coding: A systematic review of personalized, cognitive, and gamified learning in programming education. *PeerJ Comput. Sci.* **2024**, *10*, e2310. [CrossRef] [PubMed]
47. Varshney, A.K.; Torra, V. Literature Review of the Recent Trends and Applications in Various Fuzzy Rule-Based Systems. *Int. J. Fuzzy Syst.* **2023**, *25*, 2163–2186. [CrossRef]
48. Murre, J.M.J.; Dros, J. Replication and Analysis of Ebbinghaus' Forgetting Curve. *PLoS ONE* **2015**, *10*, e0120644. [CrossRef]
49. Lewis, F.L.; Campos, J.; Selmic, R. *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*; Society of Industrial and Applied Mathematics Press: Philadelphia, PA, USA, 2002.
50. Apallius de Vos, I.M.; van den Boogerd, G.L.; Fennema, M.D.; Correia, A. Comparing in context: Improving cosine similarity measures with a metric tensor. In Proceedings of the 18th International Conference on Natural Language Processing (ICON), National Institute of Technology Silchar, Silchar, India, 16–19 December 2021; NLP Association of India (NLPAI): Goa, India, 2021; pp. 128–138.
51. Hedtke, J.; Petrov, S. Semantic Similarity Search. 2019. Available online: https://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26635521.pdf?utm_source=chatgpt.com (accessed on 20 January 2025).
52. Craswell, N. Precision at n. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: Boston, MA, USA, 2009. [CrossRef]

53. Craswell, N. Mean Reciprocal Rank. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: Boston, MA, USA, 2009. [CrossRef]
54. Weston, J.; Chopra, S.; Bordes, A. Memory networks. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
55. Xu, J.; Szlam, A.; Weston, J. Beyond Goldfish Memory: Long-Term Open-Domain Conversation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Kerrville, TX, USA, 2022; pp. 5180–5197.
56. Luan, Y.; Eisenstein, J.; Toutanova, K.; Collins, M. Sparse, Dense, and Attentional Representations for Text Retrieval. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 329–345. [CrossRef]
57. Gao, J.; Galley, M.; Li, L. Neural Approaches to Conversational AI. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Kerrville, TX, USA, 2018; pp. 2–7.
58. Ennouamani, S.; Mahani, Z. Towards Adaptive Learning Systems Based on Fuzzy-Logic. In *Intelligent Computing. CompCom 2019*; Arai, K., Bhatia, R., Kapoor, S., Eds.; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2019; Volume 997, pp. 625–640. [CrossRef]
59. Sukhbaatar, S.; Szlam, A.; Weston, J.; Fergus, R. End-to-end memory networks. In Proceedings of the 29th International Conference on Neural Information Processing Systems—Volume 2 (NIPS'15), Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; Volume 2, pp. 2440–2448.
60. Pleshakova, E.; Osipov, A.; Gataullin, S.; Gataullin, T.; Vasilakos, A. Next gen cybersecurity paradigm towards artificial general intelligence: Russian market challenges and future global technological trends. *J. Comput. Virol. Hack. Tech.* **2024**, *20*, 429–440. [CrossRef]