Optimizing Reading Comprehension and Understanding in Learning by Reading Systems through Natural Language Processing and User Feedback

Liang Wenhao[1] and Zhao Meiling[2]

[1]Guizhou Minzu University, School of Computer Science, Minzu Road, Huaxi District, Guiyang, Guizhou, China
[2]Jiangxi Agricultural University, College of Information and Electrical Engineering, Yefu Road, Nanchang County, Jiangxi, China

2024

Abstract

Reading comprehension is a multifaceted challenge that involves syntactic parsing, semantic interpretation, and the alignment of textual content with stored knowledge. In environments where automated agents process textual data continuously, the optimization of these processes becomes crucial for robust learning. Recent developments in natural language processing have provided techniques for accurate lexicon-based extraction of information, improved representation of implicit contextual cues, and the facilitation of more interpretable embeddings. Integrating user feedback into this pipeline further enhances comprehension by refining system parameters and solidifying inference strategies in light of observed human insights. Such collaborative approaches rely on iterative updates to distributed vector spaces that capture linguistic phenomena and enhance the precision of automated annotations. Future systems may leverage dynamic knowledge updating, whereby newly acquired insights from readers and real-time user interactions are seamlessly incorporated into existing models to mitigate confusion from ambiguous or polysemous terms. This work explores the foundations of representation and computation that drive current advancements in reading comprehension and understanding, while highlighting design principles that scale to increasingly sophisticated domains and heterogeneous text collections. By demonstrating how structured logic statements and linear algebraic formalizations unify multiple dimensions of textual data, it becomes possible to converge on a framework that reliably supports knowledge-intensive tasks and evolves in tandem with user-driven feedback loops.

1 Introduction

Reading comprehension systems aim to extract salient information from written texts and integrate these data into coherent internal representations [1]. A fundamental requirement is the fusion of lexical, syntactic, and semantic cues in a manner that yields a faithful representation of underlying meanings. In many computational frameworks, the primary focus has been on building pipelines that first parse written material for surface-level features, such as tokens and part-of-speech tags, and subsequently apply higher-level transformations, such as dependency or constituency analysis [2]. The development of refined pipeline components that can handle the subtleties of language is an ongoing research endeavor. Various approaches, including transition-based parsing and graph-based methods, seek to increase accuracy and coverage in linguistic representations. When these structural levels are properly defined, it becomes possible to link syntactic categories with semantic interpretations in a way that systematically reduces ambiguity. [3]

Another dimension of significance is the alignment between textual elements and extralinguistic knowledge sources. In many scenarios, a text might reference specific facts, events, or entities that belong to an established domain. Traditional reading comprehension systems might rely on explicit knowledge graphs or other databases to extract relevant relationships [4]. The key challenge lies in building or updating these knowledge representations so that they reflect newly acquired information. This can be achieved by assigning unique symbolic identifiers to domain-relevant entities and, if necessary, using logical statements to define relationships between them [5]. For instance, one can use statements of the form $\forall x \, (\text{Mention}(x) \wedge \text{EntityType}(x) \rightarrow \text{Relevance}(x))$ to indicate how certain mentions in a text lead to relevant semantic content when combined with recognized entity types.

By leveraging logical constructs, one can validate whether the comprehended text is consistent with preexisting domain knowledge and detect areas of conflict or missing links that may require clarification.

In parallel, progress in representation learning has introduced methods that capture underlying semantic properties within continuous vector spaces. One may define a text fragment t as a vector $\mathbf{v}_t \in \mathbb{R}^n$, where dimension n is chosen to encapsulate different layers of linguistic information. Various mapping functions $f : t \mapsto \mathbf{v}_t$ exist, including neural encoders that process tokens sequentially or graph-based architectures that incorporate syntactic parse trees. By projecting textual units into these vector spaces, one can measure similarity or relatedness using standard norms or inner products, such as $\mathbf{v}_t^\top \mathbf{v}_{t'}$. Such similarity measures not only determine whether two fragments address similar concepts but also guide the subsequent stages of reasoning [6]. The interplay between symbolic logic representations and these continuous embeddings remains an area of active research. System designers must determine how best to reconcile rule-based constraints with the flexibility and scalability of distributed representations.

Moreover, the emergence of large-scale textual resources has propelled the need for automated reading comprehension methods that adapt to domain-specific nuances [7]. These methods must be capable of handling domain idiosyncrasies in vocabulary and concept distributions. Traditional approaches may incorporate rule sets, while modern paradigms rely more on data-driven strategies. There remain challenges such as coping with polysemous words, idiomatic expressions, and the evolving nature of specialized terminology [8], [9]. Consequently, a robust reading comprehension engine might incorporate dynamic lexical adaptation, ensuring that references to newly introduced terms integrate properly with previously established linguistic categories.

An equally crucial element is user feedback [10]. When a system interacts with end users, it receives critical hints about its performance gaps. For instance, a user might notice that the system erroneously maps a given text segment to the wrong concept in its knowledge base. The immediate question becomes how to incorporate this feedback into the system's existing models [11]. One approach is to incrementally update the parameters of the text-embedding mechanisms or the structural logic statements whenever user corrections are supplied. By introducing a feedback function $g(u, \mathbf{v}_t)$ that quantifies user feedback as a function of user input u and the text embedding $\mathbf{v}_t$, systems can modulate learned representations in real time. This process might involve adjusting attention weights in neural-based encoders or refining constraints in symbolic logic repositories. Over repeated interactions, the system transitions toward a state in which user-endorsed interpretations become dominant. [12]

The significance of a robust reading comprehension platform becomes even more evident in tasks that entail complex reasoning, such as question answering, summarization, or argument mining. The ability to disambiguate textual references, identify contradictions, and leverage prior knowledge is indispensable for generating concise and accurate system outputs. For instance, question answering may demand that the system identify relevant textual spans in a lengthy document, reconcile them with external knowledge, and produce answers in natural language [13]. Summarization systems, in turn, must detect core themes in a text and condense them without distorting the meaning. In both cases, advanced reading comprehension capabilities serve as the foundation for reliable performance. [14]

Furthermore, there is a need to distinguish between literal and implied meanings. Language can be figurative, and writers often assume significant background knowledge on the part of the reader. A sophisticated system must parse lexical units while also recognizing contexts that may alter their interpretation [15]. Symbolic logic statements can be enriched with contextual predicates. For example, an implication might hold true only under conditions α, β, γ, as stated in $(\alpha \wedge \beta \wedge \gamma) \to \delta$. Recognizing the contexts in which an entity or concept is valid can prevent erroneous generalizations [16], [17]. While continuous word embeddings can capture broad usage patterns, the addition of symbolic constraints helps restrict inferences to valid contexts.

To address these challenges holistically, researchers often employ iterative or staged processing pipelines. First, raw text data is parsed into tokens [18]. Next, structural analyses produce partial representations. Subsequent steps refine these partial representations by referencing external repositories or knowledge bases [19]. Finally, the reading comprehension engine inspects the consistency of these merged representations. The overarching goal is to move beyond shallow keyword matching toward a deeper appreciation of authorial intent, argumentation structures, and how text segments interrelate. The next sections delve into the theoretical and practical intricacies that underpin such systems, emphasizing logical formalisms, strategies for robust syntactic modeling, user feedback mechanisms, cognitive aspects of textual engagement, and techniques to integrate these processes within deep learning paradigms. [20]

2 Formalisms of Semantic Extraction

The task of capturing textual semantics can be approached through various formalizations, each highlighting different facets of meaning. One can rely on higher-order logic to capture property inheritance, compositional operations, and quantifier scope. In this framework, let S denote a discourse space, where each element $s \in S$ is a symbolic representation of an entity or concept [21]. Assertions about the discourse can then be modeled through

predicates, such as $P(s)$, or through higher-arity relations, such as $R(s_1, s_2, s_3)$. By encoding textual statements into logical formulas, one can verify the internal coherence of these statements and compare them to known facts. If a text asserts $R(a, b, c)$ and the knowledge base has a known contradiction $\neg R(a, b, c)$, the system flags potential conflicts that must be resolved. [22]

In computational scenarios, semantic extraction can also be interpreted through a graph-theoretic lens. Each node in a graph might correspond to an entity or a concept, while directed or undirected edges represent relationships [23]. This representation aligns with many advanced parsing systems that construct dependency trees or semantic graphs [24]. The text is then a collection of subgraphs whose interconnected structure reflects the underlying linguistic content. Although these formal graphs can be large, especially for extensive documents, the advantage lies in more transparent local connectivity [25]. Given a node n, one can quickly enumerate its neighbors $N(n)$ to identify lexical or semantic relations. If the semantic extraction process has assigned an edge labeled causes between nodes corresponding to textual fragments f_1 and f_2, the system can incorporate domain knowledge to determine whether such a causal link is consistent or requires refinement.

Beyond these symbolic structures, attention has turned to distributional semantics. One might represent each token, phrase, or sentence as a point in a vector space $\mathbb{R}^k$. A critical advantage of this approach is the ability to measure similarity or relatedness between terms by vector operations [26]. For instance, the normed dot product $\frac{\mathbf{v}_x \cdot \mathbf{v}_y}{\|\mathbf{v}_x\| \|\mathbf{v}_y\|}$ can serve as a proxy for semantic similarity between textual elements x and y. In large language models, this technique is extended to produce contextual embeddings, where the meaning of a word or phrase shifts depending on its surrounding text. Consequently, the representation $\mathbf{v}_w$ in a sentence $w_1, w_2, \ldots, w_n$ is influenced by all other tokens w_j. This capacity for capturing context-sensitive meaning is a notable breakthrough, enabling advanced reading comprehension systems to detect subtle distinctions in usage.

Although distributional representations can model certain aspects of meaning effectively, they may be less transparent when it comes to capturing explicit logic-based constraints or enumerating all possible inference paths [27], [28]. Symbolic representations, conversely, excel at clarity but may struggle with capturing gradient or uncertain phenomena. Hybrid systems thus aim to unify these paradigms by linking vector-based semantics with symbolic structures [29]. One approach involves augmenting a knowledge base with embedding vectors for each entity or relation label, allowing a system to approximate the likelihood of a particular inference by computing distances in the embedding space. If the system has a relation R with embedding $\mathbf{r}$ and entities e_1, e_2 with embeddings $\mathbf{e}_1, \mathbf{e}_2$, then a potential measure of how likely $R(e_1, e_2)$ is to be valid can be related to a scoring function $\phi(\mathbf{e}_1, \mathbf{r}, \mathbf{e}_2)$. Logic-based constraints can still be used to prune or guide the search space of potential relations.

Mathematical approaches to semantic extraction also rely on linear algebraic constructs that enable incremental updates to knowledge representations [30]. Suppose a text introduces a new concept c. The system must integrate $\mathbf{v}_c$ into an existing matrix of embeddings. If there is an adjacency or correlation matrix $A \in \mathbb{R}^{m \times m}$ describing relations among m known entities, then upon encountering c, the system extends A to $A' \in \mathbb{R}^{(m+1)\times(m+1)}$ by appending a new row and column. The newly introduced elements $A'_{m+1,j}$ and $A'_{j,m+1}$ are assigned initial values based on similarity measures or user feedback. Over time, if the text includes statements of the form $R(c, e)$ or indicates that c belongs to certain categories, these assignments are revised. Such updates incorporate techniques from matrix factorization, spectral methods, or gradient-based optimization, ensuring that the final representation is consistent with existing knowledge and logically compatible with user-corrected input. [31]

Formal structures for semantic extraction can also incorporate temporal or modal logic. Text often describes events that occur at specific points or intervals, and reading comprehension systems need to handle expressions of time or possibility. One might define an operator $\Box$ to denote necessity and an operator $\Diamond$ for possibility, allowing the representation of statements such as $\Box R(a, b)$ to indicate that $R(a, b)$ holds across all relevant states [32]. In temporal contexts, an operator like G could imply that a given proposition remains true globally (in all future states). The interplay of such modal and temporal operators complicates the semantics but enhances the expressiveness needed for accurate reading comprehension [33]. By fusing these operators with distributional approaches, systems gain the ability to capture not only who or what is described but also under what circumstances or conditions those descriptions hold.

The semantic extraction process thus can be seen as a pipeline in which textual inputs are successively transformed into formal representations capturing logical, relational, and contextual dimensions. Specialized algorithms then map these representations to comprehensive knowledge repositories [34]. Following this stage, user feedback or additional text can further refine the extracted information. This cyclical process serves to align computational models with ever-evolving language use and domain-specific contexts, ensuring that reading comprehension systems maintain up-to-date and accurate internal states over time.

3 Modeling Syntactic Structures

Syntactic analysis stands as a crucial step in bridging raw text and deeper semantic interpretations [35]. Various computational methods have been engineered to parse a given sentence into its syntactic skeleton, delineating how words relate hierarchically. Parsing might produce a tree structure where each node corresponds to a phrase, such as a noun phrase (NP) or verb phrase (VP), and edges denote dependency relations like subject, object, or modifier. One can represent this structure formally as a directed graph $G = (V, E)$ where each vertex $v \in V$ is a word, and edges $(v_1, v_2) \in E$ define the syntactic link from v_1 to v_2 [36]. Traditional grammar formalisms like context-free grammars (CFGs) or lexicalized probabilistic CFGs provide a foundation for these analyses, but more advanced methods may incorporate neural sequence models or transformer-based architectures that capture long-range dependencies efficiently.

In pursuit of higher accuracy, researchers have developed algorithms to manage syntactic ambiguities [37]. A sentence like "Students see teachers with microscopes" can be interpreted in multiple ways. One approach involves generating multiple parse trees with different attachment points for modifiers. Another approach uses weighted parse forests, where each parse has a probability assigned to it, and the parser selects the most likely tree based on the training data [38]. Despite these complexities, successful syntactic modeling helps disambiguate part-of-speech assignments and structural relationships crucial for advanced reasoning. For example, a subsequent logic-based system might rely on the parse tree to derive statements like HasInstrument(Student, Microscope) or HasInstrument(Teacher, Microscope), depending on the correct interpretation.

Parsing models have also evolved to incorporate morphological features. In languages with rich morphology, each word may contain information about case, gender, number, and tense, which influences syntactic structure [39], [40]. Let $\mu(w)$ denote the morphological annotation of a word w. The syntactic parser then treats each annotated token $(w, \mu(w))$ as input, expanding possible states in the parse search space. Over time, inclusion of these features has improved the ability to parse structurally complex sentences accurately, an essential requirement for robust reading comprehension systems that must handle global contexts spanning multiple sentences or paragraphs. [41]

Beyond conventional parse trees, advanced approaches make use of graph-based encodings that capture non-tree edges, particularly for languages where certain constructions defy simple tree constraints. These graph-based syntactic representations might have re-entrancies or cross-serial dependencies that are not easily captured by a single rooted tree [42]. For instance, a control structure in a sentence might cause a single verb to appear as though it has two subjects at different layers of the parse. By allowing a node v to have multiple parents or children, the graph-based representation can faithfully reflect such linguistic phenomena. Symbolic constraints then ensure that these re-entrancies remain consistent with morphological or lexical features [43]. Alternatively, an adjacency matrix can be defined, where $M_{ij} = 1$ if token i is syntactically related to token j, and 0 otherwise. Such adjacency matrices, potentially extended by user feedback, can provide a flexible backbone for subsequent semantic interpretation.

Transformer-based models have introduced another perspective on syntactic structure, where attention mechanisms dynamically learn relationships between tokens. Instead of explicitly constructing parse trees, such models implicitly encode syntactic information in their internal representations [44]. Let T be a transformer encoder, and let $\mathbf{h}_i$ be the hidden representation of token i. The attention matrix A where A_{ij} measures the attention weight from token i to token j can be interpreted as capturing syntactic and semantic dependencies. Although not always isomorphic to a formal parse tree, these attention patterns can serve as a proxy for structural relationships. Subsequent tasks can thus read off relevant dependencies from A by thresholding or otherwise processing the attention weights. Some reading comprehension systems adapt this approach to build flexible pipeline models, where the text is first fed into a transformer, and then a specialized component translates attention patterns into symbolic constraints or logical forms. [45]

Crucially, syntactic structures must be linked to semantic representations. If a parse indicates that a particular token is the subject of a verb, the semantic layer must reflect this role [46]. This alignment can be carried out by mapping syntactic roles to argument positions in a predicate. For example, if a parse reveals that the word "students" is the subject of the verb "see," and "teachers" is its object, then the system can form a logical triple See(students, teachers). Conversely, if an adverb or prepositional phrase modifies a verb, the system can attach these modifiers to the appropriate semantic role. This consistent alignment ensures that reading comprehension processes interpret grammatical relationships as meaningful constraints on who is doing what to whom or what [47]. Without careful syntactic modeling, ambiguous statements may lead to incorrect semantic inferences.

Syntactic parsers are also integral to anaphora resolution, where pronouns or references must be linked to antecedent entities in preceding text. The parse may reveal that a certain pronoun is in the subject position of a clause, matching a nearby noun phrase in number and gender features [48]. By building a chain of references, reading comprehension systems can properly accumulate contextual information. This is especially pertinent in extended discourses, where skipping or misidentifying references can cascade into errors in the final interpretation. A logic statement like $\text{RefersTo}(p, e)$ can be established if the parser and morphological analysis jointly indicate that a pronoun p is coreferential with an entity e. Larger textual analyses often rest on these microscopic syntactic

cues. [49]

Finally, advanced usage requires handling elliptical or incomplete syntactic constructions. In dialogues or casual writing, certain expected words or phrases may be omitted [50]. The parser might produce incomplete trees or guess missing elements. Systems that aim for deep reading comprehension can attempt syntactic reconstruction, hypothesizing the elliptical items that are implied by context. For instance, if the text says, "Students see teachers, but not always," the second clause might omit the object, requiring an inference that the object remains "teachers." Such phenomena exemplify the intricacy of real-language usage, making explicit syntactic analysis a cornerstone for reliable interpretation [51]. By capturing these details, reading comprehension systems can better track context and maintain textual coherence throughout a wide range of linguistic constructions.

4 User Feedback Mechanisms

Although many reading comprehension systems operate autonomously, there is growing recognition that user feedback can significantly enhance performance and adaptability. The fundamental idea is to incorporate signals from readers or domain experts into the processes that govern syntactic parsing, semantic extraction, and final inference [52]. Formally, one can define a feedback signal δ that arises whenever a discrepancy is detected between a system's interpretation and a user's expectation. The system then uses δ to adjust either its learned parameters or symbolic rules.

Consider a scenario where the system incorrectly identifies the subject of a sentence due to unusual word order in the text [53]. A user might provide feedback that the subject is, in fact, a different entity. By integrating such a correction, the parser updates its internal model to reduce the probability of generating the incorrect parse in similar contexts [54]. If the parser is based on a parameterized function θ, the system might compute a gradient $\nabla_\theta \mathcal{L}(\theta)$ where $\mathcal{L}$ is a loss function reflecting misparsed structures and user corrections. Over time, repeated corrections drive the parameters toward more robust generalizations about syntax. Similarly, for semantic extractions, if the system mistakenly classifies a phrase as an attribute rather than an event, user feedback can correct the classification and propagate that information to improve future processing.

Beyond correcting specific errors, user feedback can signal domain-relevant distinctions that the system is not initially equipped to recognize [55]. In specialized fields, terminology may not align well with general language models. By highlighting nuances, the user effectively expands the system's vocabulary or modifies its embedding space. For instance, if a specialized term t appears in a technical document, the user could indicate that t is closely related to another established domain concept c [56]. The system can then reposition the vector $\mathbf{v}_t$ closer to $\mathbf{v}_c$ in embedding space or add a new logical statement $\text{CloseRelation}(t, c)$. Such alignment helps the system immediately improve its reading comprehension performance in that domain without needing vast amounts of retraining data. This idea is particularly valuable in fields like medicine, law, or engineering, where precise terminology usage is crucial.

An additional advantage arises when user feedback is aggregated from multiple observers [57]. If a reading comprehension engine serves a large user base, aggregated feedback can yield statistically robust signals about common pitfalls or ambiguous phrases. Suppose the system processes thousands of documents daily, and certain textual patterns consistently elicit user interventions [58]. One can weight these signals by frequency or confidence to adjust the system's internal models, effectively harnessing collective intelligence. Let $\Delta = \{\delta_1, \delta_2, \ldots, \delta_k\}$ be a set of feedback signals collected over time. The system can define an aggregate function $F(\Delta)$ that produces a single global update to its knowledge base or parameters, ensuring that prevalent feedback patterns become deeply integrated. One might apply a form of incremental gradient descent, taking each δ_i in turn, or apply a batch update using the entire set Δ [59]. The net effect is an evolving system that gradually converges to more accurate interpretations consistent with user-informed reality.

Mechanisms for user feedback can also be designed to solicit clarifications in uncertain or ambiguous cases. Instead of passively waiting for error corrections, the system can detect conditions under which multiple parses or semantic interpretations have nearly identical plausibility [60]. If the difference in probability or scoring between interpretation A and B is very small, the system might query the user for clarification. In formal terms, let $P(A|T)$ be the system's probability of interpretation A given text T, and let $P(B|T)$ be the probability for interpretation B. If $|P(A|T) - P(B|T)| < \epsilon$, the system triggers a feedback request [61]. This strategy not only ensures resolution of ambiguity but also functions as an opportunity for the system to learn about subtle linguistic cues, domain-specific knowledge, or user preferences that differentiate A from B. The system subsequently updates its underlying representations so that, in the future, the correct interpretation becomes more probable without user intervention. [62]

The integration of user feedback can extend beyond discrete corrections and queries. In some cases, the user's overall satisfaction or performance in reading tasks can be treated as a reward signal. Consider a scenario in which the system supports a learning-by-reading platform where learners engage with texts, answer questions, and receive feedback on their comprehension [63]. If the user is consistently confused by certain machine-generated

highlights or summaries, the system can infer that its reading comprehension outputs are suboptimal in those areas. By defining a reward function $r(\theta)$ that reflects average user satisfaction, the system can employ reinforcement learning methods to adjust parameters. For instance, an agent might explore different ways of generating textual inferences or highlights, receiving reward signals from user behavior [64]. Over repeated interactions, policy gradient or Q-learning approaches could converge to reading comprehension strategies that maximize user satisfaction and comprehension.

Finally, feedback mechanisms must be robust against spurious or malicious inputs. In large, open platforms, users may intentionally or unintentionally provide incorrect signals [65]. The system should treat feedback as a weighted influence, factoring in user credibility or consistency with established domain knowledge. If a piece of feedback contradicts a large body of prior data or validated rules, the system can require further confirmation or reference additional sources before accepting it [66], [67]. Thus, user feedback is integrated in a manner consistent with an overarching reliability framework. The synergy between user-driven corrections and automated reading comprehension fosters a virtuous cycle: as users guide the system toward correct interpretations, the system becomes increasingly adept, requiring less corrective intervention over time.

5 Cognitive Aspects of Reading

Reading is not merely a mechanical process of parsing and decoding text; it involves intricate cognitive processes that influence comprehension, retention, and inference [68]. Although computational reading comprehension systems may not replicate the entire suite of human cognitive functions, understanding these cognitive principles can inform more advanced architectures and feature engineering. One influential perspective stems from schema theory, where readers interpret new textual information against a backdrop of existing mental frameworks. If a system stores a knowledge graph or logical database as an analog to human schemata, it can match new statements to relevant nodes and edges, expediting comprehension and highlighting incongruities. [69]

Psycholinguistic research on working memory limitations also offers insights for system design. Humans can only hold a limited amount of information active at once, influencing how they interpret ambiguous sentences or clauses that refer back to earlier segments of text. Computational analogs to working memory might impose constraints on how many tokens or partial parses can be manipulated simultaneously [70]. These constraints might be realized through bounded buffer sizes in neural encoders or limitations on the depth of certain parsing recursions. By simulating the cost of re-activating tokens or semantic units, systems can make more informed choices about how to manage references in extended discourses [71]. Alternatively, they can employ caching strategies that store partial results for quick retrieval, mirroring how humans often revisit previous sentences to clarify meaning.

Another relevant aspect is inference in reading. Humans commonly use bridging inferences to fill gaps in text [72]. For instance, if the text states, "John picked up a hammer. The nail was driven into the wood," the reader infers that John used the hammer to drive the nail. Systems aiming for robust reading comprehension might replicate this process by searching for possible bridging events given the text's context [73]. Formally, if the text contains references to "hammer," "nail," and "drive," the system might look up or generate a bridging relation $\text{UseInstrument}(\text{hammer}, \text{drive})$. By combining these inferences with syntactic analysis, the system can produce coherent explanations that match typical human understanding.

Working from theories of situation models, reading comprehension can also be viewed as constructing a dynamic mental representation of described events, entities, and relations. Each sentence updates or extends this representation, possibly requiring partial rewrites if new information conflicts with earlier assumptions [74]. Computationally, one can define a dynamic knowledge state K_t at time t. When a new sentence S is read, the system transforms K_t into K_{t+1} by integrating the content of S. If the text asserts that a character moves from one location to another, the system updates the location property of that character [75]. This process may invoke a constraint solver if prior statements conflict with the new assertion. The final outcome is a consolidated model of the situation described across the entire text segment. Logic statements like $\text{At}(\text{Character}, \text{Location2})$ can override previous statements $\text{At}(\text{Character}, \text{Location1})$ if the text indicates a change.

Additionally, human readers use metacognitive strategies to monitor and regulate their comprehension [76]. They may realize that they did not understand a particular sentence and reread it. In computational systems, a parallel approach might involve self-evaluation modules that estimate the confidence of each semantic extraction or inference. If the confidence falls below a threshold, the system might re-invoke the parser or the logic engine with adjusted parameters [77]. Alternatively, the system might seek user clarification when feasible. This self-monitoring loop ensures that the model does not propagate misunderstandings unchecked into higher-level tasks. Probability-based frameworks can assign low confidence scores to ambiguous or contradictory interpretations, prompting reanalysis or feedback solicitation. [78]

Reading also invokes emotional and connotative layers, though many current computational systems do not delve deeply into these aspects. Still, some advanced systems include sentiment analysis or affective computing modules that annotate text with polarity or emotion labels [79]. For instance, a sentence about a scientific discovery

might carry a positive connotation. If the text later describes controversy or negative outcomes, the system can note a shift in sentiment, which might affect subsequent reading comprehension tasks such as summarization or question answering. Formally, if one defines a sentiment function $\sigma(S)$ on a text segment S, the system might track changes in $\sigma(S)$ across the narrative structure [80]. Although these emotional cues may not directly affect logical consistency, they can shape user perceptions of the text and highlight aspects of reading comprehension relevant to user-centered applications.

Finally, considering cognitive load is vital when designing interactive reading systems. If a system presents too many details or an overly complex representation, it may overwhelm readers [81]. Conversely, if it oversimplifies the text, comprehension may suffer. Balancing detail requires a measure of readability for both humans and the computational model. Just as humans gauge whether a text is too technical, systems can estimate textual complexity by analyzing syntactic depth, lexical variety, or the presence of specialized terms [82]. When user feedback indicates confusion, the system can adapt by providing clarifications or reformatting summaries into simpler structures. This approach mirrors how teachers or expert readers might paraphrase or annotate challenging passages [83]. Over time, the system refines its approach, targeting an optimal level of detail that supports robust comprehension without causing extraneous cognitive load.

6 Integration with Deep Neural Paradigms

Deep neural networks have revolutionized natural language processing, enabling the development of models that learn nuanced representations of text through end-to-end training. Central to these methods is the use of multi-layer architectures, which progressively transform input sequences into higher-level features [84]. Let $x = (x_1, x_2, \ldots, x_n)$ be a sequence of tokens from a text. A neural architecture might encode each token x_i into an embedding $\mathbf{e}_i$. Through a series of transformations, such as recurrent networks, convolutional filters, or transformer blocks, the system produces context-enriched representations $\mathbf{h}_i$. These representations can then feed into task-specific heads that perform classification, question answering, or summarization.

One challenge is reconciling these deep neural techniques with the more structured symbolic and logical approaches that have historically guided reading comprehension [85]. One solution is a hybrid pipeline in which neural components generate candidate semantic structures or fill knowledge bases with likely inferences, and then a symbolic module applies constraints or logical checks. For instance, a neural semantic parser might convert a sentence into a set of candidate logical forms $\{L_1, L_2, \ldots, L_k\}$ with associated probabilities $\{p_1, p_2, \ldots, p_k\}$. A subsequent symbolic reasoning layer can examine these candidate forms for internal consistency or alignment with existing knowledge. The highest-scoring logical form that remains consistent is selected as the final representation [86]. While this approach can be computationally expensive, it achieves a balance between the flexibility of neural models and the interpretability of symbolic logic.

Another avenue involves embedding symbolic knowledge into neural architectures [87], [88]. Techniques such as knowledge graph embedding train neural models to align entity and relation embeddings with known facts, making it easier to incorporate domain-specific information. If a reading comprehension system is tasked with answering questions about a text, it can augment the text encoder with an embedding matrix for known entities and relations. When a mention of an entity e appears, the system links it to the corresponding embedding $\mathbf{e} \in \mathbb{R}^d$ in the knowledge graph. By attending to these embeddings during text encoding, the model gains immediate access to relevant domain knowledge [89]. This synergy is especially powerful for specialized areas, such as biomedical text analysis, where correct interpretation depends heavily on domain-specific facts about proteins, genes, or diseases.

Deep neural models also enable transfer learning, wherein a model pre-trained on large-scale corpora is fine-tuned on specific reading comprehension tasks. Let M_{pre} be a base model trained on a massive textual dataset. For a given downstream task, one defines a task-specific objective $\mathcal{L}_{\text{task}}$. During fine-tuning, the base model parameters are updated to minimize $\mathcal{L}_{\text{task}}$. This procedure allows the system to leverage general linguistic knowledge acquired during pre-training [90]. The result is often superior performance with relatively little labeled data. For reading comprehension, fine-tuning might involve a labeled dataset of questions and answers for specific texts, or annotated semantic roles. The system thus adapts to the subtleties of question interpretation and context retrieval without requiring manual engineering of domain rules. [91]

In addition to supervised fine-tuning, unsupervised or self-supervised learning paradigms can refine reading comprehension models. Techniques like masked language modeling or next-sentence prediction are used to capture contextual dependencies and text coherence [92]. These tasks can be extended to more advanced objectives designed specifically to foster deeper comprehension. For instance, a model might learn to predict whether two sentences logically follow each other or if a bridging inference is needed. By training on large text corpora, the model acquires latent representations that can expedite the process of building textual coherence [93]. When integrated with symbolic checks or user feedback, such a model becomes more robust in handling edge cases or domain-specific phenomena.

Another dimension of deep neural integration is the use of attention mechanisms to capture cross-sentence

and cross-paragraph relationships. Some reading comprehension tasks require fusing information from multiple parts of a document [94]. If the text is split into segments $\{T_1, T_2, \ldots, T_m\}$, each segment can be encoded using a transformer. A higher-level cross-attention mechanism can then integrate these segment embeddings into a coherent representation. This hierarchical approach mirrors human reading strategies, where individuals read multiple paragraphs and form a global mental model. Formally, let $\mathbf{H}^{(j)}$ be the set of hidden states for segment T_j. A cross-attention module can produce an integrated representation $\mathbf{u}$ by weighting and combining the states across all j. This integrated representation can then be used for tasks such as summarization, question answering, or contradiction detection. [95]

Yet, while deep neural networks exhibit impressive performance, they often behave as black boxes. Users or researchers may struggle to interpret why a model arrives at a particular conclusion [96]. This opacity can be partially mitigated by explainability techniques, such as attention visualizations or saliency maps. However, a more direct path to interpretability emerges when neural systems work alongside symbolic representations. By explicitly constructing logic-based statements for each inference, the system can provide a trace of its reasoning [97]. If a neural module is responsible for generating these statements, user feedback can identify flaws in the generation process. This fosters a loop in which user corrections gradually align the neural model's learned representations with user expectations for transparency and correctness.

Finally, deploying deep neural paradigms for reading comprehension requires careful attention to computational resource management and latency [98]. Large-scale models with billions of parameters can be expensive to run in real-time applications. Techniques such as model distillation, parameter pruning, or quantization can reduce resource usage while retaining the performance needed for advanced text interpretation. If the reading comprehension system interacts with users continuously, it must serve responses in a timely manner [99]. One approach is to maintain a smaller, distilled model for inference and a larger model for batch re-training or offline updates. Over time, the offline model can integrate new texts, incorporate user feedback, and refine the knowledge base, while the online model provides quick responses [100]. This architecture mirrors the separation of fast, reactive processes from slower, reflective processes in human cognition, reinforcing the synergy between user demands for immediate comprehension and the system's longer-term drive for accuracy and comprehensiveness.

7 Conclusion

As reading by automated systems shifts from basic text processing to more profound comprehension, it becomes evident that a multifaceted approach is indispensable. Traditional parsing and symbolic logic provide a degree of interpretability and consistency checking, yet they can falter under the overwhelming variety, ambiguity, and evolving nature of real-world language [101], [102]. Deep neural paradigms, on the other hand, exhibit remarkable adaptability and performance but often sacrifice interpretability and explicit reasoning. The path to a balanced framework combines these elements, leveraging rigorous symbolic representations, distributional embeddings, and user feedback to achieve a more holistic understanding of text.

In such a combined system, syntax is modeled through formal grammars or attention-based architectures that capture long-range dependencies and morphological nuances [103]. Semantics emerges from logic-based schemas and distributional vector spaces that track context-sensitive meanings. User feedback mechanisms operate at every level, correcting errors in structural analyses or enriching the knowledge base with domain-specific details. These incremental updates align the system's models with the practical realities of language use, ensuring that oversights and incorrect assumptions are progressively diminished [104]. The incorporation of cognitive principles, such as working memory constraints and bridging inferences, adds robustness, enabling the system to emulate certain aspects of human reading strategies and reduce interpretive gaps in ambiguous passages.

High-level tasks, such as question answering, summarization, or domain-specific reasoning, benefit from these improved representations [105]. By cross-referencing internal symbolic structures with neural embeddings, the system pinpoints not only what is being stated but also how it connects to preexisting knowledge. Over time, repeated user interactions serve as a valuable feedback loop, refining embeddings, grammar rules, or logic statements to handle specialized terminology or rare linguistic constructions. This capacity for continuous learning underscores the long-term viability of reading comprehension frameworks that adapt in tandem with the texts and users they serve. [106]

Ultimately, the journey toward optimizing reading comprehension and understanding in automated systems demands not a single solution but a tapestry of interwoven techniques. Logical formalism underpins consistency checks and explicit reasoning. Distributional semantics and deep neural networks empower the system with flexibility and generalization [107]. User feedback knits these components into a living architecture that reflects real-world language dynamics. By evolving in response to domain shifts and iterative corrections, such systems offer the promise of robust, transparent, and contextually aware reading comprehension. This promise becomes increasingly vital as textual data grows in volume and complexity, and as readers and practitioners demand more intelligent and adaptive tools for engaging with information. [108]

References

[1] Y. Zhong, J.-H. Feng, X. Cui, and X.-L. Cui, "Machine learning aided key-guessing attack paradigm against logic block encryption," *Journal of Computer Science and Technology*, vol. 36, no. 5, pp. 1102–1117, Sep. 30, 2021. DOI: 10.1007/s11390-021-0846-6.

[2] Z. Gu, J. Yi, H. Yao, and Y. Wang, "Intelligent online guiding network regional planning based on software-driven autonomous communication system," *Automated Software Engineering*, vol. 29, no. 1, Jan. 3, 2022. DOI: 10.1007/s10515-021-00309-7.

[3] M. Kearney, P. F. Burke, and S. Schuck, "The ipac scale: A survey to measure distinctive mobile pedagogies," *TechTrends*, vol. 63, no. 6, pp. 751–764, Aug. 6, 2019. DOI: 10.1007/s11528-019-00414-1.

[4] M. Facchin, "Are generative models structural representations," *Minds and Machines*, vol. 31, no. 2, pp. 277–303, Mar. 30, 2021. DOI: 10.1007/s11023-021-09559-6.

[5] N. D. Agham and U. M. Chaskar, "Learning and non-learning algorithms for cuffless blood pressure measurement: A review.," *Medical & biological engineering & computing*, vol. 59, no. 6, pp. 1201–1222, Jun. 3, 2021. DOI: 10.1007/s11517-021-02362-6.

[6] C. M. Martínez, J. R. Piorno, J. J. E. Otero, and M. G. Mata-García, "Responsive inclusive design (rid): A new model for inclusive software development," *Universal Access in the Information Society*, vol. 22, no. 3, pp. 893–902, Jul. 1, 2022. DOI: 10.1007/s10209-022-00893-9.

[7] C.-H. Cheng, Y.-S. Chen, A. K. Sangaiah, and Y.-H. Su, "Evidence-based personal applications of medical computing models in risk factors of cardiovascular disease for the middle-aged and elderly," *Personal and Ubiquitous Computing*, vol. 22, no. 5, pp. 921–936, Jun. 19, 2018. DOI: 10.1007/s00779-018-1172-z.

[8] B. M. Bhatti, S. Mubarak, and S. V. Nagalingam, "Information security implications of using nlp in it outsourcing: A diffusion of innovation theory perspective," *Automated software engineering*, vol. 28, no. 2, pp. 1–29, Jul. 16, 2021. DOI: 10.1007/s10515-021-00286-x.

[9] M. Kansara, "A framework for automation of cloud migrations for efficiency, scalability, and robust security across diverse infrastructures," *Quarterly Journal of Emerging Technologies and Innovations*, vol. 8, no. 2, pp. 173–189, 2023. DOI: 10.17613/bnw08-t6s36.

[10] P. V. de Campos Souza and E. Lughofer, "Evolving fuzzy neural classifier that integrates uncertainty from human-expert feedback.," *Evolving systems*, vol. 14, no. 2, pp. 319–341, Aug. 15, 2022. DOI: 10.1007/s12530-022-09455-z.

[11] D. Basile, M. H. ter Beek, A. Ferrari, and A. Legay, "Exploring the ertms/etcs full moving block specification: An experience with formal methods," *International Journal on Software Tools for Technology Transfer*, vol. 24, no. 3, pp. 351–370, Apr. 10, 2022. DOI: 10.1007/s10009-022-00653-3.

[12] null Sakshi and V. Kukreja, "A dive in white and grey shades of ml and non-ml literature: A multivocal analysis of mathematical expressions," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 7047–7135, Dec. 6, 2022. DOI: 10.1007/s10462-022-10330-1.

[13] S. Kaur, S. Bawa, and R. Kumar, "A survey of mono- and multi-lingual character recognition using deep and shallow architectures: Indic and non-indic scripts," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1813–1872, May 25, 2019. DOI: 10.1007/s10462-019-09720-9.

[14] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi, and J. S. Suri, "Human activity recognition in artificial intelligence framework: A narrative review.," *Artificial intelligence review*, vol. 55, no. 6, pp. 4755–4808, Jan. 18, 2022. DOI: 10.1007/s10462-021-10116-x.

[15] Z. Feng, "Hot news mining and public opinion guidance analysis based on sentiment computing in network social media," *Personal and Ubiquitous Computing*, vol. 23, no. 3, pp. 373–381, Dec. 18, 2018. DOI: 10.1007/s00779-018-01192-y.

[16] F. Badosa, A. Espinosa, C. Acevedo, G. Vera, and A. Ripoll, "A history-based resource manager for genome analysis workflows applications on clusters with heterogeneous nodes," *International Journal of Parallel Programming*, vol. 47, no. 2, pp. 317–342, Sep. 22, 2018. DOI: 10.1007/s10766-018-0600-z.

[17] A. Abhishek and A. Basu, "A framework for disambiguation in ambiguous iconic environments," in *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*, Springer, 2005, pp. 1135–1140.

[18] R. K. Tesser and E. Borin, "Containers in hpc: A survey," *The Journal of Supercomputing*, vol. 79, no. 5, pp. 5759–5827, Oct. 27, 2022. DOI: 10.1007/s11227-022-04848-y.

[19] J. Zheng, L. Shen, X. Peng, H. Zeng, and W. Zhao, "Mashredroid: Enabling end-user creation of android mashups based on record and replay," *Science China Information Sciences*, vol. 63, no. 10, pp. 1–20, Sep. 16, 2020. DOI: 10.1007/s11432-019-2646-2.

[20] V. G. Costa and C. E. Pedreira, "Recent advances in decision trees: An updated survey," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4765–4800, Oct. 10, 2022. DOI: 10.1007/s10462-022-10275-5.

[21] Z. Chen, X. Cheng, S. Dong, *et al.*, "Information retrieval: A view from the chinese ir community," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 151 601–, Sep. 29, 2020. DOI: 10.1007/s11704-020-9159-0.

[22] J. Borrego-Díaz and J. Galán-Páez, "Explainable artificial intelligence in data science," *Minds and Machines*, vol. 32, no. 3, pp. 485–531, May 12, 2022. DOI: 10.1007/s11023-022-09603-z.

[23] X. Chen and Q. Li, "Event modeling and mining: A long journey toward explainable events," *The VLDB Journal*, vol. 29, no. 1, pp. 459–482, Jul. 1, 2019. DOI: 10.1007/s00778-019-00545-0.

[24] M. Yuan and A. Vlachos, "Zero-shot fact-checking with semantic triples and knowledge graphs," in *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, 2024, pp. 105–115.

[25] J. P. Singh, S. Jain, S. Arora, and U. P. Singh, "A survey of behavioral biometric gait recognition: Current success and future perspectives," *Archives of Computational Methods in Engineering*, vol. 28, no. 1, pp. 107–148, Nov. 20, 2019. DOI: 10.1007/s11831-019-09375-3.

[26] *A closer look onto breast density with weakly supervised dense-tissue masks*, vol. 14, Suppl 1, Germany: Springer Science and Business Media LLC, May 18, 2019. DOI: 10.1007/s11548-019-01969-3.

[27] T. A. Soomro, L. Zheng, A. J. Afifi, A. H. H. Ali, M. Yin, and J. Gao, "Artificial intelligence (ai) for medical imaging to combat coronavirus disease (covid-19): A detailed review with direction for future research.," *Artificial intelligence review*, vol. 55, no. 2, pp. 1–31, Apr. 15, 2021. DOI: 10.1007/s10462-021-09985-z.

[28] A. Basu *et al.*, "Iconic interfaces for assistive communication," in *Encyclopedia of Human Computer Interaction*, IGI Global, 2006, pp. 295–302.

[29] F. L. da Silva, B. K. Slodkowski, K. K. A. da Silva, and S. C. Cazella, "A systematic literature review on educational recommender systems for teaching and learning: Research trends, limitations and opportunities.," *Education and information technologies*, vol. 28, no. 3, pp. 3289–3328, Sep. 14, 2022. DOI: 10.1007/s10639-022-11341-9.

[30] C. Casalnuovo, K. Sagae, and P. Devanbu, "Studying the difference between natural and programming language corpora," *Empirical Software Engineering*, vol. 24, no. 4, pp. 1823–1868, Jan. 11, 2019. DOI: 10.1007/s10664-018-9669-7.

[31] C.-M. Rim, Y.-C. Sin, and K.-H. Paek, "A mobile robot localization method based on polar scan matching and adaptive niching chaos optimization algorithm," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 1, Sep. 6, 2022. DOI: 10.1007/s10846-022-01724-y.

[32] M. Y. Mhawish and M. Gupta, "Predicting code smells and analysis of predictions: Using machine learning techniques and software metrics," *Journal of Computer Science and Technology*, vol. 35, no. 6, pp. 1428–1445, Nov. 30, 2020. DOI: 10.1007/s11390-020-0323-7.

[33] J. L. Sánchez-Cervantes, L. O. Colombo-Mendoza, G. Alor-Hernández, J. L. García-Alcaraz, J. M. Alvarez-Rodríguez, and A. Rodríguez-González, "Lindasearch: A faceted search system for linked open datasets," *Wireless Networks*, vol. 26, no. 8, pp. 5645–5663, May 22, 2019. DOI: 10.1007/s11276-019-02029-z.

[34] M. Ishaq, A. Abid, M. S. Farooq, *et al.*, "Advances in database systems education: Methods, tools, curricula, and way forward.," *Education and information technologies*, vol. 28, no. 3, pp. 2681–2725, Aug. 31, 2022. DOI: 10.1007/s10639-022-11293-0.

[35] K. Lu and H. Liao, "A survey of group decision making methods in healthcare industry 4.0: Bibliometrics, applications, and directions.," *Applied intelligence (Dordrecht, Netherlands)*, vol. 52, no. 12, pp. 13 689–13 713, Jan. 5, 2022. DOI: 10.1007/s10489-021-02909-y.

[36] F. Puentes, M. Pérez-Godoy, P. González, and M. J. del Jesus, "An analysis of technological frameworks for data streams," *Progress in Artificial Intelligence*, vol. 9, no. 3, pp. 239–261, Jun. 28, 2020. DOI: 10.1007/s13748-020-00210-6.

[37] S. Cheng, J. Wang, X. Shen, Y. Chen, and A. K. Dey, "Collaborative eye tracking based code review through real-time shared gaze visualization," *Frontiers of Computer Science*, vol. 16, no. 3, pp. 1–11, Nov. 18, 2021. DOI: 10.1007/s11704-020-0422-1.

[38] R. Santana, L. Martí, and M. Zhang, "Gp-based methods for domain adaptation: Using brain decoding across subjects as a test-case," *Genetic Programming and Evolvable Machines*, vol. 20, no. 3, pp. 385–411, May 10, 2019. DOI: 10.1007/s10710-019-09352-6.

[39] A. Maiti, M. Sultana, and S. Bhattacharya, "Detection of skin cancer through hybrid color features and soft voting ensemble classifier," *Innovations in Systems and Software Engineering*, Nov. 16, 2022. DOI: 10.1007/s11334-022-00498-8.

[40] A. Sharma, M. Witbrock, and K. Goolsbey, "Controlling search in very large commonsense knowledge bases: A machine learning approach," *arXiv preprint arXiv:1603.04402*, 2016.

[41] A. Aslam and E. Curry, "Investigating response time and accuracy in online classifier learning for multimedia publish-subscribe systems," *Multimedia tools and applications*, vol. 80, no. 9, pp. 13 021–13 057, Jan. 9, 2021. DOI: 10.1007/s11042-020-10277-x.

[42] G. Taranto-Vera, P. Galindo-Villardón, J. Merchán-Sánchez-Jara, J. Salazar-Pozo, A. Moreno-Salazar, and V. Salazar-Villalva, "Algorithms and software for data mining and machine learning: A critical comparative view from a systematic review of the literature," *The Journal of Supercomputing*, vol. 77, no. 10, pp. 11 481–11 513, Mar. 25, 2021. DOI: 10.1007/s11227-021-03708-5.

[43] P. A. Kirschner, J. Sweller, F. Kirschner, and R. J. Zambrano, "From cognitive load theory to collaborative cognitive load theory," *International journal of computer-supported collaborative learning*, vol. 13, no. 2, pp. 213–233, Apr. 25, 2018. DOI: 10.1007/s11412-018-9277-y.

[44] V. N. M. Aradhya, H. T. Basavaraju, and D. S. Guru, "Decade research on text detection in images/videos: A review," *Evolutionary Intelligence*, vol. 14, no. 2, pp. 405–431, Jun. 6, 2019. DOI: 10.1007/s12065-019-00248-z.

[45] P. U. B. Albuquerque, D. K. de A. Ohi, N. S. Pereira, B. de A. Prata, and G. C. Barroso, "Proposed architecture for energy efficiency and comfort optimization in smart homes," *Journal of Control, Automation and Electrical Systems*, vol. 29, no. 6, pp. 718–730, Aug. 30, 2018. DOI: 10.1007/s40313-018-0410-y.

[46] A. Trifa, A. Hedhili, and W. L. Chaari, "Knowledge tracing with an intelligent agent, in an e-learning platform," *Education and Information Technologies*, vol. 24, no. 1, pp. 711–741, Aug. 25, 2018. DOI: 10.1007/s10639-018-9792-5.

[47] S. Piantadosi, "The computational origin of representation," *Minds and machines*, vol. 31, no. 1, pp. 1–58, Nov. 3, 2020. DOI: 10.1007/s11023-020-09540-9.

[48] S. Ferilli, D. Redavid, and D. D. Pierro, "Holistic graph-based document representation and management for open science," *International Journal on Digital Libraries*, vol. 24, no. 4, pp. 205–227, Jun. 29, 2022. DOI: 10.1007/s00799-022-00328-z.

[49] L. A. Leria, P. Benitez, and F. J. Fraga, "Assistive technology in large-scale assessments for students with visual impairments: A systematic review and recommendations based on the brazilian reality," *Education and Information Technologies*, vol. 26, no. 3, pp. 3543–3573, Jan. 14, 2021. DOI: 10.1007/s10639-020-10419-6.

[50] C. M. Keet, "The african wildlife ontology tutorial ontologies.," *Journal of biomedical semantics*, vol. 11, no. 1, pp. 1–11, Jun. 23, 2020. DOI: 10.1186/s13326-020-00224-y.

[51] H. Lyre, "The state space of artificial intelligence," *Minds and Machines*, vol. 30, no. 3, pp. 325–347, Sep. 7, 2020. DOI: 10.1007/s11023-020-09538-3.

[52] S. Jamshidi, R. Rejaie, and J. Li, "Characterizing the dynamics and evolution of incentivized online reviews on amazon," *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 1–15, May 10, 2019. DOI: 10.1007/s13278-019-0563-0.

[53] A. Zaffar and S. M. A. Hussain, "Modeling and prediction of kse - 100 index closing based on news sentiments: An applications of machine learning model and arma (p, q) model.," *Multimedia tools and applications*, vol. 81, no. 23, pp. 33 311–33 333, Apr. 18, 2022. DOI: 10.1007/s11042-022-13052-2.

[54] V. M. Niño-Martínez, J. O. Ocharán-Hernández, X. Limón, and J. C. Pérez-Arriaga, "A microservice deployment guide," *Programming and Computer Software*, vol. 48, no. 8, pp. 632–645, Dec. 21, 2022. DOI: 10.1134/s0361768822080151.

[55] Y. Guo, "Design of artistic graphic symbols based on intelligent guidance marking system," *Neural Computing and Applications*, vol. 35, no. 6, pp. 4255–4266, Mar. 13, 2022. DOI: 10.1007/s00521-022-07088-6.

[56] M. Talal, A. A. Zaidan, B. B. Zaidan, *et al.*, "Smart home-based iot for real-time and secure remote health monitoring of triage and priority system using body sensors: Multi-driven systematic review," *Journal of medical systems*, vol. 43, no. 3, pp. 42–42, Jan. 15, 2019. DOI: 10.1007/s10916-019-1158-z.

[57] S. Ali, Y. Hafeez, M. Humayun, N. Jhanjhi, and D.-N. Le, "Towards aspect based requirements mining for trace retrieval of component-based software management process in globally distributed environment," *Information Technology and Management*, vol. 23, no. 3, pp. 151–165, Nov. 5, 2021. DOI: 10.1007/s10799-021-00343-7.

[58] D.-R. Beddiar, M. Oussalah, and T. Seppänen, "Automatic captioning for medical imaging (mic): A rapid review of literature.," *Artificial intelligence review*, vol. 56, no. 5, pp. 4019–4076, Sep. 17, 2022. DOI: 10.1007/s10462-022-10270-w.

[59] D. Doloreux, J. G. de la Puerta, I. Pastor-López, I. P. Gómez, B. Sanz, and J. M. Zabala-Iturriagagoitia, "Territorial innovation models: To be or not to be, that's the question," *Scientometrics*, vol. 120, no. 3, pp. 1163–1191, Jul. 13, 2019. DOI: 10.1007/s11192-019-03181-1.

[60] J. Wang, A. Jatowt, M. Färber, and M. Yoshikawa, "Improving question answering for event-focused questions in temporal collections of news articles," *Information Retrieval Journal*, vol. 24, no. 1, pp. 29–54, Jan. 2, 2021. DOI: 10.1007/s10791-020-09387-9.

[61] J. Stenseke, "Artificial virtuous agents: From theory to machine implementation," *AI & SOCIETY*, vol. 38, no. 4, pp. 1301–1320, Dec. 15, 2021. DOI: 10.1007/s00146-021-01325-7.

[62] H. Azevedo, J. P. R. Belo, and R. A. F. Romero, "Using ontology as a strategy for modeling the interface between the cognitive and robotic systems," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 3, pp. 431–449, Aug. 29, 2019. DOI: 10.1007/s10846-019-01076-0.

[63] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based lstm encoder-decoder network in cloud environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–18, Dec. 17, 2019. DOI: 10.1186/s13638-019-1605-z.

[64] C. Bozsahin, "Computers aren't syntax all the way down or content all the way up," *Minds and Machines*, vol. 28, no. 3, pp. 543–567, Jul. 18, 2018. DOI: 10.1007/s11023-018-9469-2.

[65] G. Boella, L. Di, and V. Leone, "Semi-automatic knowledge population in a legal document management system," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 227–251, Dec. 13, 2018. DOI: 10.1007/s10506-018-9239-8.

[66] L. Foschini, G. Martuscelli, R. Montanari, and M. Solimando, "Edge-enabled mobile crowdsensing to support effective rewarding for data collection in pandemic events," *Journal of grid computing*, vol. 19, no. 3, pp. 1–17, Jul. 8, 2021. DOI: 10.1007/s10723-021-09569-9.

[67] A. Sharma and K. M. Goolsbey, "Learning search policies in large commonsense knowledge bases by randomized exploration," 2018.

[68] C. Ragkhitwetsagul and J. Krinke, "Siamese: Scalable and incremental code clone search via multiple code representations," *Empirical Software Engineering*, vol. 24, no. 4, pp. 2236–2284, Mar. 5, 2019. DOI: 10.1007/s10664-019-09697-7.

[69] C. Xie, Y. Wang, J. MacIntyre, M. Sheikh, and M. Elkady, "Using sensors data and emissions information to diagnose engine's faults.," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 1142–1152, 2018. DOI: 10.2991/ijcis.11.1.86.

[70] F. Khalil and G. Pipa, "Transforming the generative pretrained transformer into augmented business text writer," *Journal of Big Data*, vol. 9, no. 1, Nov. 22, 2022. DOI: 10.1186/s40537-022-00663-7.

[71] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Building direct citation networks," *Scientometrics*, vol. 115, no. 2, pp. 817–832, Feb. 20, 2018. DOI: 10.1007/s11192-018-2676-z.

[72] E. Loginova, S. Varanasi, and G. Neumann, "Towards end-to-end multilingual question answering," *Information Systems Frontiers*, vol. 23, no. 1, pp. 227–241, Feb. 28, 2020. DOI: 10.1007/s10796-020-09996-1.

[73] J. Lin, G. Sun, T. Cui, *et al.*, "From ideal to reality: Segmentation, annotation, and recommendation, the vital trajectory of intelligent micro learning," *World Wide Web*, vol. 23, no. 3, pp. 1747–1767, Oct. 23, 2019. DOI: 10.1007/s11280-019-00730-9.

[74] M. E. Zorrilla and M. de Lima Silva, "Sociograms: An effective tool for decision making in social learning," *Technology, Knowledge and Learning*, vol. 24, no. 4, pp. 659–681, May 30, 2019. DOI: 10.1007/s10758-019-09416-7.

[75] C. Zhou, "Integration of modern technologies in higher education on the example of artificial intelligence use," *Education and Information Technologies*, vol. 28, no. 4, pp. 3893–3910, Oct. 4, 2022. DOI: 10.1007/s10639-022-11309-9.

[76] S. Dhahbi, M. Berrima, and F. A. M. Al-Yarimi, "Load balancing in cloud computing using worst-fit bin-stretching," *Cluster Computing*, vol. 24, no. 4, pp. 2867–2881, May 15, 2021. DOI: 10.1007/s10586-021-03302-7.

[77] P. C. Mondal, "Mental structures as biosemiotic constraints on the functions of non-human (neuro)cognitive systems," *Biosemiotics*, vol. 13, no. 3, pp. 385–410, Aug. 25, 2020. DOI: 10.1007/s12304-020-09390-z.

[78] K. Bouchard, J. Lapalu, B. Bouchard, and A. Bouzouane, "Clustering of human activities from emerging movements," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3505–3517, Oct. 3, 2018. DOI: 10.1007/s12652-018-1070-2.

[79] F. Ouyang, L. Zheng, and P. Jiao, "Artificial intelligence in online higher education: A systematic review of empirical research from 2011 to 2020," *Education and Information Technologies*, vol. 27, no. 6, pp. 7893–7925, Feb. 26, 2022. DOI: 10.1007/s10639-022-10925-9.

[80] G. Latif, A. Shankar, J. Alghazo, V. Kalyanasundaram, C. S. Boopathi, and M. A. Jaffar, "I-cares: Advancing health diagnosis and medication through iot," *Wireless Networks*, vol. 26, no. 4, pp. 2375–2389, Oct. 18, 2019. DOI: 10.1007/s11276-019-02165-6.

[81] F. Folino and L. Pontieri, "Ai-empowered process mining for complex application scenarios: Survey and discussion," *Journal on Data Semantics*, vol. 10, no. 1, pp. 77–106, Mar. 9, 2021. DOI: 10.1007/s13740-021-00121-2.

[82] M. Majdalawieh, N. Nizamuddin, M. Ala'raj, S. N. Khan, and A. Bani-Hani, "Blockchain-based solution for secure and transparent food supply chain network," *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, pp. 3831–3850, Jul. 20, 2021. DOI: 10.1007/s12083-021-01196-1.

[83] M. Autili, I. Malavolta, A. Perucci, G. L. Scoccia, and R. Verdecchia, "Software engineering techniques for statically analyzing mobile apps : Research trends, characteristics, and potential for industrial adoption," *Journal of Internet Services and Applications*, vol. 12, no. 1, pp. 1–60, Jul. 23, 2021. DOI: 10.1186/s13174-021-00134-x.

[84] S. Klikovits and D. Buchs, "Pragmatic reuse for dsml development," *Software and Systems Modeling*, vol. 20, no. 3, pp. 837–866, Oct. 14, 2020. DOI: 10.1007/s10270-020-00831-4.

[85] A. P. Piotrowski and A. E. Piotrowska, "Differential evolution and particle swarm optimization against covid-19.," *Artificial intelligence review*, vol. 55, no. 3, pp. 1–71, Aug. 19, 2021. DOI: 10.1007/s10462-021-10052-w.

[86] J. Monteiro, J. Barata, M. Veloso, L. Veloso, and J. Nunes, "A scalable digital twin for vertical farming," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 10, pp. 13 981–13 996, Aug. 1, 2022. DOI: 10.1007/s12652-022-04106-2.

[87] H. Yin, F. S. Melo, A. Paiva, and A. Billard, "An ensemble inverse optimal control approach for robotic task learning and adaptation," *Autonomous Robots*, vol. 43, no. 4, pp. 875–896, Apr. 30, 2018. DOI: 10.1007/s10514-018-9757-y.

[88] A. Sharma, K. M. Goolsbey, and D. Schneider, "Disambiguation for semi-supervised extraction of complex relations in large commonsense knowledge bases," in *7th Annual Conference on Advances in Cognitive Systems*, 2019.

[89] P. Juárez-Smith, L. Trujillo, M. García-Valdez, F. Vega, and F. Chávez, "Local search in speciation-based bloat control for genetic programming," *Genetic Programming and Evolvable Machines*, vol. 20, no. 3, pp. 351–384, Mar. 23, 2019. DOI: 10.1007/s10710-019-09351-7.

[90] Z. Nasar, S. W. Jaffry, and M. K. Malik, "Information extraction from scientific articles: A survey," *Scientometrics*, vol. 117, no. 3, pp. 1931–1990, Sep. 29, 2018. DOI: 10.1007/s11192-018-2921-5.

[91] O. Saha, P. Dasgupta, and B. Woosley, "Real-time robot path planning from simple to complex obstacle patterns via transfer learning of options," *Autonomous Robots*, vol. 43, no. 8, pp. 2071–2093, May 8, 2019. DOI: 10.1007/s10514-019-09852-5.

[92] A. Bakki, L. Oubahssi, S. George, and C. Cherkaoui, "A model and tool to support pedagogical scenario building for connectivist mooc," *Technology, Knowledge and Learning*, vol. 25, no. 4, pp. 899–927, Apr. 16, 2020. DOI: 10.1007/s10758-020-09444-8.

[93] H. Yarahmadi and S. M. H. Hasheminejad, "Design pattern detection approaches: A systematic review of the literature," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5789–5846, Apr. 20, 2020. DOI: 10.1007/s10462-020-09834-5.

[94] F. Niederman and E. W. Baker, "Ethics and ai issues: Old container with new wine?" *Information Systems Frontiers*, vol. 25, no. 1, pp. 9–28, Jun. 30, 2022. DOI: 10.1007/s10796-022-10305-1.

[95] S. D. Riccio, D. Dyankov, G. Jansen, G. D. Fatta, and G. Nicosia, "Icann (2) - pareto multi-task deep learning," in Germany: Springer International Publishing, Oct. 14, 2020, pp. 132–141. DOI: 10.1007/978-3-030-61616-8_11.

[96] J. Wen, L. Yeshu, Y. Shi, H. Huang, D. Bing, and X. Xinping, "A classification model for lncrna and mrna based on k-mers and a convolutional neural network," *BMC bioinformatics*, vol. 20, no. 1, pp. 1–14, Sep. 13, 2019. DOI: 10.1186/s12859-019-3039-3.

[97] A. S. Albahri, A. A. Zaidan, O. S. Albahri, B. B. Zaidan, and M. A. Alsalem, "Real-time fault-tolerant mhealth system: Comprehensive review of healthcare services, opens issues, challenges and methodological aspects," *Journal of medical systems*, vol. 42, no. 8, pp. 1–56, Jun. 23, 2018. DOI: 10.1007/s10916-018-0983-9.

[98] P. C. A. Brepohl and H. Leite, "Virtual reality applied to physiotherapy: A review of current knowledge," *Virtual Reality*, vol. 27, no. 1, pp. 71–95, Jul. 22, 2022. DOI: 10.1007/s10055-022-00654-2.

[99] B. Athira, J. Jones, S. M. Idicula, A. Kulanthaivel, and E. Zhang, "Annotating and detecting topics in social media forum and modelling the annotation to derive directions-a case study," *Journal of Big Data*, vol. 8, no. 1, pp. 1–23, Feb. 27, 2021. DOI: 10.1186/s40537-021-00429-7.

[100] A. S. Patel, R. Vyas, O. P. Vyas, and M. Ojha, "A study on video semantics; overview, challenges, and applications," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6849–6897, Jan. 19, 2022. DOI: 10.1007/s11042-021-11722-1.

[101] S. Nadal, P. Jovanovic, B. Bilalli, and O. Romero, "Operationalizing and automating data governance.," *Journal of big data*, vol. 9, no. 1, pp. 117–, Dec. 10, 2022. DOI: 10.1186/s40537-022-00673-5.

[102] K. D. Forbus, C. Riesbeck, L. Birnbaum, K. Livingston, A. Sharma, and L. Ureel, "Integrating natural language, knowledge representation and reasoning, and analogical processing to learn by reading," in *Proceedings of the National Conference on Artificial Intelligence*, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, vol. 22, 2007, p. 1542.

[103] A. Diveev and S. V. Konstantinov, "Study of the practical convergence of evolutionary algorithms for the optimal program control of a wheeled robot," *Journal of Computer and Systems Sciences International*, vol. 57, no. 4, pp. 561–580, Aug. 25, 2018. DOI: 10.1134/s106423071804007x.

[104] G. Cortellessa, F. Fracasso, A. Umbrico, *et al.*, "Co-design of a tv-based home support for early stage of dementia," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 4, pp. 4541–4558, Mar. 2, 2020. DOI: 10.1007/s12652-020-01823-4.

[105] R. Espejo, "Cybersyn, big data, variety engineering and governance.," *AI & society*, vol. 37, no. 3, pp. 1163–1177, Jan. 4, 2022. DOI: 10.1007/s00146-021-01348-0.

[106] G. Polhill, J. Ge, M. Hare, *et al.*, "Crossing the chasm: A 'tube-map' for agent-based social simulation of policy scenarios in spatially-distributed systems," *GeoInformatica*, vol. 23, no. 2, pp. 169–199, Jan. 7, 2019. DOI: 10.1007/s10707-018-00340-z.

[107] J. B. Lee, T. Lee, and H. P. In, "Topic modeling based warning prioritization from change sets of software repository," *Journal of Computer Science and Technology*, vol. 35, no. 6, pp. 1461–1479, Nov. 30, 2020. DOI: 10.1007/s11390-020-0047-8.

[108] C. Castiello and C. Mencar, "Fine-tuning the fuzziness of strong fuzzy partitions through pso," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 1415–1428, 2020. DOI: 10.2991/ijcis.d.200904.002.