www.ijarp.com ISSN 2456-9992 Page: 01-12

International Journal Advanced Research Publications

Review Article

Volume: 02
Issue: 04

E-LIB VOICE: A PROTOTYPE VOICE-CENTRIC AI E-LIBRARY FRAMEWORK FOR SECURE ACCESS, GROUNDED QUESTION ANSWERING, AND ADAPTIVE RECOMMENDATION

[*1]Nishad Sutar, [2]Aditi Gotawade, [3]Anuja Ghadge, [4]Taqi Khan, [5]Dr. Ramesh Shahabade

[1]Student, Department of Computer Engineering Terna Engineering College University of Mumbai, Navi Mumbai, India.

[2]Student, Department of Computer Engineering Terna Engineering College University of Mumbai, Navi Mumbai, India.

[3]Student, Department of Computer Engineering Terna Engineering College University of Mumbai, Navi Mumbai, India.

[4]Student, Department of Computer Engineering Terna Engineering College University of Mumbai, Navi Mumbai, India.

[5]Professor, Department of Computer Engineering Terna Engineering College, University of Mumbai, Navi Mumbai, India.

Article Received: 15 March 2026, Article Revised: 04 April 2026, Published on: 24 April 2026

***Corresponding Author: Nishad Sutar**

Student, Department of Computer Engineering Terna Engineering College University of Mumbai, Navi Mumbai, India.

DOI: https://doi-doi.org/101555/ijarp.2139

ABSTRACT

Digital libraries are widely adopted in higher edu- cation, yet most production systems still assume keyboard-driven navigation, manual browsing, and visually intensive interaction. This paper presents *E-lib Voice*, a prototype voice-centric e- library framework that integrates speaker-aware access control, local speech processing, retrieval-augmented question answering, automated PDF ingestion, and adaptive recommendation within a Django-based architecture. The system combines browser-side voice activity detection, Resemblyzer-based speaker verification, faster-whisper transcription, TF-IDF with Logistic Regression for intent detection, a rule-guided smart router, Pinecone-backed retrieval, and large language model generation through Open- Router. Rather than claiming benchmark superiority, the paper frames the system as an engineering prototype for studying how voice-first interaction can be layered onto digital library workflows while preserving factual grounding and modular main- tainability. Mathematical formulations are provided for biometric similarity, intent

estimation, retrieval scoring, and recommen- dation inference. Because evidence remains prototype-oriented, evaluation is framed through scenario validation, module-level testing, and a future quantitative assessment framework.

***INDEX TERMS*:** Voice interfaces, digital libraries, speaker ver- ification, retrieval-augmented generation, intent classification, recommender systems, accessible computing.

I. INTRODUCTION

Digital libraries have become essential infrastructure for contemporary learning, but their interaction models have evolved more slowly than their storage and retrieval ca- pabilities. Most systems provide searchable catalogs, docu- ment viewers, and metadata-based navigation; however, the dominant user workflow still depends on text input, click-driven traversal, and sustained visual attention. This reduces accessibility for users with limited motor mobility and impedes multitasking learners who must move repeatedly between search, reading, and question answering.

The problem is not merely one of convenience. Existing e- library interfaces typically separate navigation, reading, secu- rity, and information retrieval into different interaction layers. A user may authenticate using conventional credentials, search using keyword interfaces, open a document manually, and then rely on external tools or generic assistants for summaries or explanations. Compared with conventional catalog-driven portals, such systems provide weak hands-free interaction; compared with generic voice assistants, they lack direct access to institution-specific book collections; and compared with standalone large language model chat interfaces, they provide limited factual grounding to local holdings. This fragmentation becomes especially problematic for long-form educational documents, where users may want to issue voice commands, receive audio playback, ask context-specific questions, and obtain recommendations within a single session.

E-lib Voice is proposed as a prototype framework to inves- tigate a unified alternative. The system combines voice-driven navigation, speaker-based authentication, automatic speech recognition, intent classification, retrieval-augmented conver- sational assistance, automated PDF structuring, and hybrid recommendation under one modular backend. The project is not positioned as a finished production platform or benchmark- winning model stack; instead, it is presented as a pilot ar- chitecture for studying how multiple AI components can be orchestrated in a practical digital library setting with explicit scope boundaries.

This paper makes four contributions. First, it formulates a research-oriented design for a

voice-first e-library that unifies command execution and conversational retrieval in one inter- action loop. Second, it maps limitations observed in prior e- library and conversational systems to concrete design require- ments, including grounded answering, lightweight intent han- dling, and adaptive personalization. Third, it provides math- ematical formulations for the main inference stages, thereby connecting the engineering implementation to a reproducible methodological description. Fourth, it proposes an evaluation framework suitable for prototype-stage systems where scenario validation is available but large-scale benchmark evidence is not yet established.

II. Literature Review and Design Requirements

Research relevant to E-lib Voice spans classical information retrieval, automatic speech recognition, speaker verification, retrieval-augmented generation, and recommender systems. Traditional digital library search is strongly informed by doc- ument weighting and ranking methods such as TF-IDF, which remain robust for corpus indexing and query matching [1], [2]. These foundations explain why keyword search remains effective for catalogs, yet they do not resolve the interaction burden imposed by menu-based or text-only interfaces.

Recent speech systems have shown substantial gains in recognition quality through large- scale weak supervision and neural sequence modeling [3]. Similarly, embedding-based speaker verification methods demonstrate that a short speech signal can be transformed into a compact representation suit- able for identity matching [4]. These advances suggest that voice can be used not only as an input modality, but also as a security and personalization signal. However, translating these models into a library application introduces system- level questions that the literature often leaves open, such as when to bypass costly verification, how to distinguish between navigational commands and informational queries, and how to preserve privacy when biometric data is involved.

Grounded conversational systems have increasingly relied be unnecessarily expensive when low-latency deterministic routing is more important than nuanced semantic modeling.

Recommendation research similarly highlights a tradeoff be- tween collaborative and content-based methods. Collaborative filtering captures peer behavior but suffers when user history is sparse, whereas content-based filtering supports personaliza- tion from item descriptors but depends on informative feature representations [6], [7]. In educational and library settings, this implies that a practical system must change recommendation logic as user profiles mature rather than relying on a single static method.

The literature therefore does not point to a single mono- lithic model choice; rather, it maps

operational gaps to de- sign requirements. The limited accessibility of conventional e-library interfaces motivates a voice-first interaction layer. The lack of grounded, corpus-specific assistance motivates a retrieval-augmented chatbot rather than a purely generative assistant. The mismatch between low-latency command rout- ing and transformer-heavy conversational models motivates a lightweight intent classifier coupled with a rule-aware dis- patcher. Finally, the sparse-history problem in personalization motivates a hybrid recommendation strategy that changes behavior as user profiles mature. The present work adopts these requirements directly, treating prior research not only as background, but also as a requirements source for the proposed prototype.

III. METHODOLOGY

The methodology models E-lib Voice as a staged inference pipeline in which each module transforms user input into a more structured representation. The system receives speech or document input, performs authentication and transcrip- tion where needed, predicts the interaction mode, retrieves contextual evidence when the query is informational, and generates either a deterministic command action or a grounded conversational response. Figure 1 summarizes the online path followed by a spoken user query.

A. Voice and Authentication Pipeline

When a user provides speech input, the browser captures an audio segment and ends recording after approximately 1.2 seconds of silence. Let $\mathbf{e}_u \in \mathrm{R}^d$ denote the enrolled voiceprint for user u and $\mathbf{e}_x \in \mathrm{R}^d$ denote the embedding extracted from a candidate utterance, where the implementation uses a 256-dimensional speaker representation. Identity confidence is measured through cosine similarity, on retrieval-augmented generation to reduce unsupported re- sponses by conditioning the language model on retrieved external evidence [5]. Dense embedding models further im- prove semantic matching between user queries and candidate text passages [9], while transformer-based language models provide strong contextual representations [8], [10]. Yet these approaches are not always a good fit for every component of an interactive system. In particular, applying transformer- heavy inference to short, repetitive interface commands may Access is accepted when $S_{\text{voice}}(u, x) \geq \tau$, where τ is an application-level threshold chosen to balance false acceptance.

and false rejection risk. This formulation is appropriate for a prototype because it exposes the verification mechanism without overstating empirical security guarantees.

In the implemented workflow, strict verification is retained for voice login and other protected interactions, whereas some.

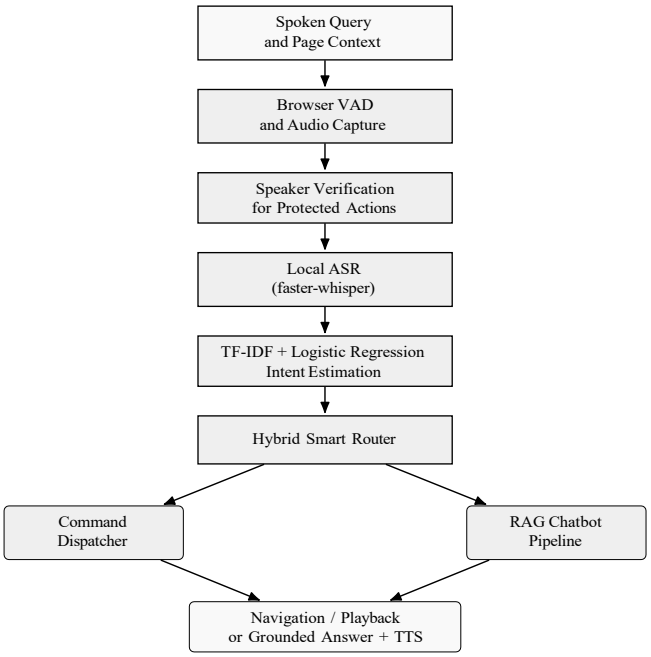

Fig. 1. End-to-end online interaction pipeline for spoken user queries in E-lib Voice.

low-risk public-page interactions may bypass the expensive biometric check to control latency. After authentication or low-risk bypass, the audio is transcribed using a local speech recognizer. Local transcription is a deliberate methodological choice because it reduces dependence on external API latency and limits transmission of raw speech data. The resulting transcript is then forwarded to the intent and routing stage.

B. Intent Estimation and Smart Routing

Intent detection is formulated as a multiclass classification problem over a limited set of site commands and interaction intents. A transcript is normalized through lowercasing, punctuation removal, and tokenization, and is then transformed into a TF-IDF feature vector. For token t in utterance d, the weight is defined as

$$w_{t,d} = \mathrm{tf}(t, d) \log \frac{N}{1 + \mathrm{df}(t)}, \qquad (2)$$

where p denotes page context, C is the set of command intents, Q is the set of conversational intents, and g is an ordered rule set over lexical patterns, page state, and utterance length.

In the implemented prototype, very high-confidence command predictions trigger direct command dispatch, while ambiguous cases are resolved through these fallback rules. This hybrid design reflects an engineering observation: the system is not required to solve

unrestricted semantic interpretation for every short utterance, but it must fail safely when the distinction between control and conversation is uncertain.

$$P(y = k \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{x} + b_k)}{\sum_j \exp(\mathbf{w}_j^\top \mathbf{x} + b_j)}. \quad (3)$$

$$\hat{r}(\mathbf{x}, p) = \begin{cases} \text{command}, & \max_{k \in \mathcal{C}} P(y = k \mid \mathbf{x}) \geq \gamma_c \\ \text{chatbot}, & \max_{k \in \mathcal{C}} P(y = k \mid \mathbf{x}) \geq \gamma_u \\ g(\mathbf{x}, p), & \text{otherwise}, \end{cases} \quad (4)$$

C. *Retrieval-Augmented Question Answering*

If the router classifies an utterance as informational, the query is sent to the retrieval-augmented branch. Uploaded textbooks are first segmented into overlapping chunks to preserve local context across page boundaries. Let q be a user query, d_i a candidate chunk, and $\phi(\cdot)$ an embedding function. Retrieval relies on nearest-neighbor similarity,

$$\text{score}(q, d_i) = \cos\ \phi(q), \phi(d_i)\ , \quad (5)$$

with the top-k chunks selected as supporting context. In the present implementation, Pinecone indexes chunk embeddings and returns the top five candidates for prompt construction.

The generation stage is intentionally constrained. Rather than allowing unconstrained response synthesis, the system assembles a prompt that instructs the large language model to answer using only the retrieved context. This does not eliminate hallucination in a formal sense, but it operationally reduces unsupported claims by binding generation to retrieved evidence. Retrieval therefore serves not only a performance role, but also a governance role by narrowing the answer space to the ingested corpus.

D. *Adaptive Recommendation Model*

The recommendation subsystem is designed around profile maturity. Let n_u denote the number of logged interactions for user u. If $n_u < 5$, the system applies a cold-start strategy based on collaborative signals from users with overlapping genre and profile attributes. Recommendation scores in this phase are derived from peer frequency counts over candidate books after excluding items already consumed by the target user. where N is the number of training utterances and df(t) is the number of utterances containing t.

The feature vector $\mathbf{x}$ is passed to a Logistic Regression classifier. For class k, the posterior estimate is exp $(\mathbf{w}^\top \mathbf{x} + b_k)$ If $n_u \geq 5$, the system transitions to content-based recom-

mendation. Each book b is represented by a TF-IDF vector $\mathbf{v}_b$ built from concatenated metadata such as title, author, genres, description, and year. The implementation uses unigram and bigram features with a vocabulary cap of 5000 dimensions.

The user taste profile is represented by the centroid

$$\mathbf{c}_u = \frac{1}{|H_u|} \sum_{b \in H_u} \mathbf{v}_b \qquad (6)$$

where H_u is the set of books associated with the user's reading history. The recommendation score for candidate book b is then

$$R(u, b) = \cos(\mathbf{c}_u, \mathbf{v}_b). \qquad (7)$$

This formulation captures the intuition that recommendation quality should depend on the angular proximity between a user's aggregate content preferences and the metadata profile of unseen books.

IV. System Framework

The system is implemented as a modular Django-based framework in which authentication, catalog management, voice processing, chatbot orchestration, recommendations, and administrative operations are separated into dedicated applica- tion components. Concretely, the codebase is partitioned into user, book-management, chatbot, voice-bot, recommendation, and administration services that expose distinct API surfaces while sharing a common persistence layer. Figure 2 shows the layered relationship between the client interface, application modules, and model/data backends. This modularity is sig- nificant for research positioning because it allows individual modules to be substituted or evaluated independently without redesigning the entire platform.

At the interaction layer, the framework begins with client- side microphone capture and silence-based segmentation. The backend then performs conditional speaker verification, fol- lowed by speech-to-text conversion. The transcript is passed to a smart routing stage that decides whether to invoke a deter- ministic command handler or the retrieval-augmented chatbot. If the utterance is a command, the system executes actions such as navigation, search, playback control, or save-state modification. If the utterance is conversational, the system encodes the query, retrieves relevant book chunks, constructs a grounded prompt, sends the prompt through OpenRouter, and returns the answer as text and optionally as synthesized speech. When the user is already in a reading session, the conversational

branch can operate inline so that question an- swering does not require abandoning the audiobook workflow. The framework also includes a document ingestion pathway. Uploaded PDFs are parsed with PyMuPDF, cleaned using regular-expression filters, and scanned for chapter-like bound- aries. The resulting text units are stored in structured form for page-level reading, audio rendering, embedding generation, and retrieval. Batched database writes are executed using bulk_create inside transaction.atomic() blocks, which is a key engineering choice because it reduces insertion overhead and lowers the risk of partially committed document states.

From a workflow perspective, the framework connects four objectives often separated in existing systems: access control, interaction, comprehension support, and personalization. This makes the prototype useful as a systems research artifact even without large-scale deployment.

V. IMPLEMENTATION

The implementation combines conventional web infrastruc- ture with lightweight AI components chosen for operational fit. Django provides routing, model management, REST endpoint organization, and authentication support. MySQL is used for structured persistence, while Pinecone stores vector represen- tations for semantic retrieval. JWT-based session management supports authenticated client interactions across the platform. The project uses Resemblyzer for voiceprint generation, faster-whisper for local transcription, scikit-learn for TF-IDF and Logistic Regression, PyMuPDF for document extraction, and gTTS for speech synthesis. The intent model and vec- torizer are serialized with joblib and loaded lazily on server startup to avoid retraining costs during normal operation. Speaker encoders are handled in singleton-like fashion to reduce repeated memory allocation, synthesized speech is cached by content hash to avoid redundant text-to-speech requests, and recommendation caches are invalidated when the library corpus changes. Recent implementation refinements described in the technical documentation also include batching optimizations for PDF ingestion and safeguards around rec- ommendation queries to avoid problematic MySQL subquery
patterns.

Several design decisions are notable from a research- engineering standpoint. TF-IDF with Logistic Regression was selected over transformer classifiers because the command space is limited and the platform requires fast CPU-side inference. Speech recognition and speaker verification are executed locally where possible to reduce privacy exposure and network

dependence, while retrieval augmentation is pre- ferred over end-to-end generative answering because the target domain is an uploaded book collection rather than unrestricted web knowledge. Operationally, the framework also exposes separate maintenance procedures for document reprocessing, embedding rebuilds, and intent retraining, which improves reproducibility by allowing model and corpus updates to be conducted independently of user-facing traffic.

VI. Evaluation and Findings

The current evidence base is prototype-oriented rather than benchmark-oriented. Accordingly, evaluation is framed in terms of scenario-based validation, module-level testing, and future metric design rather than claimed state-of-the-art performance. The project documentation reports representative testing over four critical workflows: a voice command case, a conversational query case, a speaker verification case, and a PDF upload case. The implementation also supports isolated module testing under a lightweight development database con- figuration, allowing orchestration logic to be checked without the full production stack. In the documented prototype tests, the observed outputs matched the expected workflow behavior, indicating that the end-to-end orchestration is functional under controlled development conditions.

For a research paper, however, prototype success is not suf- ficient without a formal evaluation framework. The appropriate future assessment of the intent module should report accuracy, macro-precision, macro-recall, and confusion patterns across command classes. The voice subsystem should be assessed us- ing authentication acceptance and rejection statistics, plus end- to-end latency from speech termination to action execution. The retrieval-augmented chatbot should be evaluated through

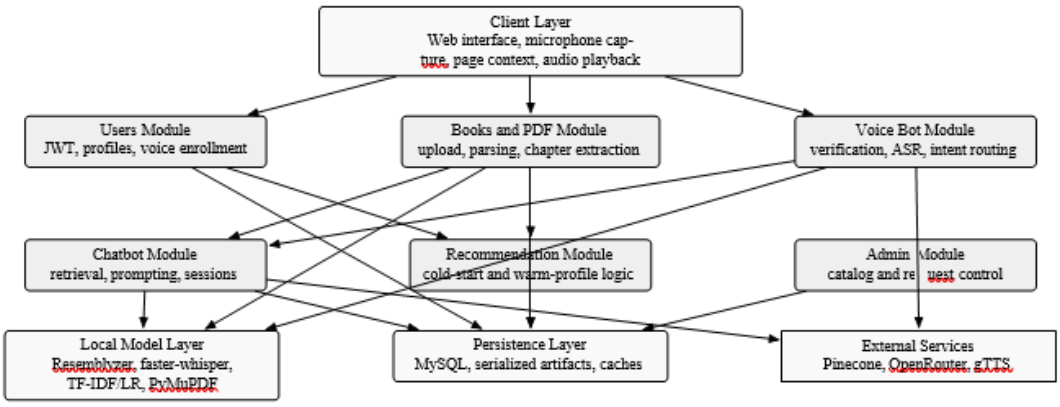

Fig. 2. Layered system architecture of E-lib Voice, connecting the browser interface

to modular Django services and the underlying model, storage, and external retrieval layers.

TABLE I
PROTOTYPE VALIDATION SCENARIOS

Case	Input	Prototype finding
TC1	"Open dashboard"	Command intent was recognized and routed to the navigation handler.
TC2	"Who is the author of Harry Potter book?"	Query was routed to the chatbot pipeline and answered through retrieval-backed processing.
TC3	Valid enrolled voice sample	Voice verification stage accepted the sample and enabled protected interaction.
TC4	PDF upload	Document was parsed and structured for storage, playback, and retrieval use.

response latency, evidence relevance, and human judgment of factual support from retrieved passages. The recommendation engine should be assessed using ranking-oriented metrics such as Precision@*k*, Recall@*k*, and normalized discounted cumulative gain, together with cold-start versus warm-profile ablations.

A useful system-level evaluation protocol would therefore combine three layers. The first layer is component evalua- tion, measuring each model in isolation. The second layer is workflow evaluation, measuring completion success for repre- sentative tasks such as logging in by voice, opening a book, asking for a summary, and receiving recommendations. The third layer is user-centered evaluation, measuring usability, perceived accessibility, and trust in grounded responses. This layered framework is especially important because the value of E-lib Voice depends on orchestration quality across modules rather than on a single standalone classifier.

The current findings support several cautious conclusions. The architecture appears viable for integrating command exe- cution, grounded conversational retrieval, and recommendation into a single interface. At the same time, the absence of large-scale benchmark data means the present work should be interpreted as a proof-of-concept system study rather than text books, which may reduce retrieval quality for cross-page concepts. Recommendation quality is limited by metadata completeness and the sparsity of early user histories. In addition, the current implementation is effectively English- centric and has not yet been validated across multilingual or highly diverse deployment settings.

There are also governance and ethical considerations. Voiceprints are biometric data and therefore require careful storage, access control, and consent management. Retrieval-augmented answers are more constrained than fully generative ones, but they may still reflect extraction errors, retrieval misses, or model misinterpretation of context. Recommenda- tion

outputs may overemphasize frequently read genres and underexpose less common material. These issues suggest that future versions should incorporate clearer audit trails, stronger user controls, and more explicit data governance policies.

Future work should therefore proceed along both technical and evaluative dimensions. On the technical side, promising directions include transformer-based or distilled intent models for harder paraphrase cases, adaptive multi-sample speaker enrollment, cross-encoder re-ranking for retrieval, multilingual support, and improved chunking strategies for long documents. On the evaluation side, the system would benefit from curated command datasets, controlled latency studies, user trials with accessibility-focused participants, and ablation experiments isolating the contribution of routing logic, retrieval grounding, and adaptive recommendation.

VII. Conclusion

This paper presented E-lib Voice as a prototype voice- centric AI e-library framework that integrates speaker-aware access control, local speech processing, grounded question answering, automated PDF ingestion, and adaptive recommen- dation. The central research contribution is not a claim of superior benchmark performance, but a systems-level demon- stration that these components can be combined coherently to support more accessible and conversational interaction with digital library content.

By grounding design choices in literature-derived gaps, expressing the main inference stages mathematically, and framing evaluation in realistic prototype terms, the paper positions E-lib Voice as a credible pilot system for further study. The current results support feasibility, while the stated limitations clarify that substantial empirical evaluation remains necessary before broader deployment claims can be made. Even at this stage, the system offers a useful blueprint for how future educational platforms may combine voice interaction, retrieval grounding, and adaptive personalization.

References

1. G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
2. C. D. Manning, P. Raghavan, and H. Schu¨tze, *Introduction to Informa- tion Retrieval*. Cambridge, U.K.: Cambridge University Press, 2008.
3. Radford *et al.*, “Robust speech recognition via large-scale weak supervision,” arXiv:2212.04356, 2022.
4. L. Wan, Q. Wang, A. Papir, and I. Lopez-Moreno, “Generalized end-to- end loss for

speaker verification," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 4879–4883.

5. P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.
6. P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
7. Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
8. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapo- lis, MN, USA, 2019, pp. 4171–4186.
9. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods in Natural Language Processing and Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3982– 3992.
10. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.