КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.34.1352
UDC 004.8

Oleksandr Marchenko
DSc (Phys. & Math.), Professor of the Department of Mathematical Informatics
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
ORCID: 0000-0002-5408-5279
omarchenko@univ.kiev.ua

Maksym Shevchenko
PhD student of the Department of Mathematical Informatics
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
ORCID: 0009-0008-5104-9767
maksym_shevchenko@knu.ua

RAG-BASED DIALOG-STATE CONVERSATIONAL RECOMMENDER FOR UKRAINIAN BOOKS

Abstract. Finding books in digital catalogs through natural-language interaction requires a system to combine verifiable catalog constraints, less formal thematic preferences, and context accumulated across multiple conversational turns. The aim of this study is to develop and experimentally evaluate a retrieval-augmented conversational recommender with an explicit dialog state for Ukrainian-language book requests. The proposed system operates over a normalized research catalog of approximately 5,500 Ukrainian books and separates natural-language understanding (NLU), deterministic dialog policy, structured and vector retrieval, user-profile ranking, and grounded response generation. A strict intermediate frame distinguishes catalog entities from semantic preferences and preserves unresolved mentions, while the dialog manager maintains active constraints, focus, pending actions, feedback, and recommendation history. Depending on the request, the system selects structured, semantic, hybrid, similar-book, author, or profile-based retrieval. Evaluation was conducted at three complementary levels: 90 deterministic regression tests, standalone NLU assessment on 1,000 Ukrainian inputs, and 70 end-to-end (E2E) scenarios comprising 170 turns. All regression tests passed. The large language model (LLM)-based parser achieved 98.3% intent accuracy, 88.7% constraint F1, and 88.0% relaxed full-frame accuracy. In E2E evaluation, the two tested NLU configurations reached 90.59–92.94% relaxed frame accuracy, 92.94–96.47% dialog-manager subset accuracy, and 97.44% recommendation-assertion accuracy. Manual inspection of 185 recommendation-bearing outputs found no book-level mismatch among the lists actually returned. Neither configuration produced duplicate works within a recommendation list or repeated a work across turns of the same scenario. The results demonstrate that explicit state, deterministic retrieval control, and grounded generation can provide reproducible, context-aware book recommendations without delegating catalog membership or item selection to free-form generation. The architecture and evaluation protocol can be reused for other catalog-grounded conversational recommendation tasks.

Keywords: conversational recommender system; retrieval-augmented generation; dialog state; natural-language understanding; hybrid retrieval; vector search; recommendation evaluation.

INTRODUCTION

Problem statement. Digital catalogs reduce the physical cost of accessing literature, but they do not eliminate the cognitive cost of choosing among many items. Conventional search interfaces are effective when a reader already knows a title, author, or catalog category. They are less suitable when the reader expresses an incomplete preference, combines hard and soft conditions, or changes the request during interaction. A request may contain an exact constraint such as an author, publication period, language, or maximum number of pages, while also containing a thematic preference such as "a reflective story about memory and loss" («рефлексивна історія про пам’ять і втрату»). The two kinds of information should not be processed identically: catalog constraints can be verified and filtered symbolically, whereas thematic preferences are more naturally handled through semantic representations.

Recommender-system research commonly distinguishes content-based, collaborative, and hybrid approaches [1], [2]. In a book domain, content descriptions and catalog metadata are particularly valuable because

they make it possible to recommend an item even when extensive interaction statistics are unavailable. Conversely, user feedback can personalize ranking by representing the reader's accumulated preferences. A conversational recommender adds another source of evidence: the current dialog. Instead of requiring the user to specify all preferences in one form, it can request missing information, interpret follow-up utterances, record feedback, and refine an earlier search [3], [4].

This setting creates a state-management problem. Consider the sequence "recommend family sagas" («порадь сімейні саги»), followed by "only two, and shorter ones" («лише дві, і коротші»). The second utterance is not an independent query. It modifies the requested count and adds a page-related preference while retaining the active thematic preference. A later utterance such as "now show books by Lina Kostenko" («тепер покажи книги Ліни Костенко») should instead start a new search and clear the earlier theme and page constraints. The system must therefore distinguish continuation from replacement, and it must do so without silently carrying obsolete filters into a new task.

The same problem occurs with contextual references. Expressions such as "this book" («ця книга»), "its author" («її автор»), or "similar to the first one" («схожа на першу») are meaningful only relative to a focus established by previous turns. Dialog-state tracking research treats the maintained state as a compact representation of the user's current goal and of evidence accumulated across turns [5]. Explicit state is especially useful in a recommender because it can be inspected, validated, and applied to retrieval. It also avoids making the language model responsible for reconstructing every operational detail from an unstructured transcript.

Natural-language understanding for Ukrainian introduces additional ambiguity. Ukrainian is morphologically rich, and the same person, title, or category can appear in several inflected forms. Recent Ukrainian named entity recognition research likewise identifies limited annotated resources and linguistic complexity as practical difficulties [6]. In a book catalog, ambiguity is not limited to morphology. A phrase can resemble more than one metadata field, and a general concept can coincide with the name of a catalog category or series. For example, "Chinese manga" («китайська манга») may denote the category Manga plus a language or country-related property, or it may be a single semantic description. An implementation must preserve the user's meaning without forcing every phrase into a catalog slot.

Large language models provide flexible interpretation and response generation, but a fully generative pipeline is difficult to control. A model can produce plausible titles or metadata that are not present in the local collection. Retrieval-augmented generation (RAG) addresses this general problem by combining parametric language processing with explicit non-parametric knowledge [7]. For a recommender, however, retrieval must do more than supply text passages. It must execute structured filters, select a retrieval route, rank candidate books, enforce recommendation-history rules, and return stable identifiers. Generation should verbalize those decisions instead of replacing them.

Analysis of recent studies and publications. Classical recommender systems estimate which items may be useful to a user from item characteristics, interaction histories, or a combination of both. Content-based methods compare item representations with a user profile, while collaborative methods infer preference from relationships among users, items, and ratings. Hybrid systems combine complementary evidence or switch between methods under different conditions [1], [2]. These distinctions remain relevant in a conversational system, but the input is no longer limited to a historical interaction matrix. The current utterance and dialog state become first-class recommendation signals.

Similarity measures are central to both item retrieval and profile matching. The proposed system uses cosine-based comparison because book descriptions, semantic queries, and preference profiles are represented as high-dimensional embedding vectors. This comparison ranks candidates according to vector orientation, so vector magnitude does not dominate the distance used by the retrieval layer [8].

Conversational recommender systems integrate preference elicitation, recommendation, explanation, feedback, and dialog management. Previous surveys [3], [4] show that evaluation is difficult precisely because a conversational recommender contains multiple interdependent tasks. A relevant book may still be delivered through an awkward response, while a fluent response may be based on an incorrectly understood constraint. Component-level and E2E evaluations therefore answer different questions and should not be collapsed into one score.

Several studies illustrate alternative approaches to conversational recommendation. ReDial and an E2E neural architecture for movie recommendation dialogs were introduced in [9]. That line of work demonstrates the value of real conversational data and E2E neural modeling. For the present study, the explicit separation of NLU, dialog management, retrieval, and natural-language generation (NLG) also makes component-level error attribution more direct. The configurable ConveRSE framework was proposed in [10], where disambiguation was identified as an important part of natural-language recommendation. Their results also support combining natural language with explicit interface controls. The temporal contribution of dialog context was modeled in [11], with external knowledge and retrieval used to improve both recommendation and response generation. Together, these

works confirm the importance of context, but they do not remove the need for domain-grounded state and deterministic retrieval rules.

Dialog-state tracking provides a useful conceptual foundation. A dialog manager maintains a representation of the user's goal and uses it to select the next system action [5]. The proposed architecture adopts this separation and uses a transparent application-specific state instead of a probability distribution over latent goals. This choice makes the state directly actionable: author and category slots become SQL predicates, semantic text becomes a vector query, focus entities resolve elliptical references, and pending fields define clarification behavior.

RAG provides the second foundation. It was shown in [7] that generation can be conditioned on retrieved non-parametric knowledge. In the present system, the idea is extended from passage retrieval to recommendation orchestration. The non-parametric layer consists of a relational catalog, a vector index, persistent user feedback, and a structured dialog state. The generated response is downstream of these resources. It does not independently decide which books exist or which identifiers are returned.

Evaluation literature further cautions against treating recommendation accuracy as the only objective. Conversational recommenders should be evaluated at the level of individual components, recommendation task support, dialog behavior, and user-facing interaction [12]. Established recommender-evaluation frameworks also distinguish relevance-oriented measures from properties such as diversity, novelty, serendipity, and coverage [13], [14], [15]. The present work therefore reports NLU scores, dialog manager behavior, formal recommendation assertions, and output-diversity indicators separately.

Existing studies provide separate foundations for conversational recommendation, dialog-context modeling, retrieval-augmented generation, and recommender evaluation. This work addresses the system-level integration of these methods for a Ukrainian book catalog. The resulting system combines an explicit NLU frame, deterministic state transitions, relational and semantic retrieval, user feedback, work-level novelty, and reproducible multi-level testing. The contribution concerns this system-level combination and its experimental evaluation, not the individual methods in isolation.

The aim of this article is to develop and experimentally evaluate an architecture that converts Ukrainian requests into catalog-grounded and context-aware recommendations by combining structured filtering, semantic retrieval, profile ranking, and constrained NLG. The object of this study is the process of generating book recommendations in a multi-turn natural-language dialog. The subject is the architecture, algorithms, data structures, and evaluation methods of a dialog-state retrieval-augmented recommender for Ukrainian books.

The study addresses four research questions. First, how accurately does LLM-based NLU recover intents, catalog constraints, referenced entities, and semantic preferences compared with deterministic baselines? Second, how reliably does the complete system maintain dialog state, select a retrieval route, and satisfy formal recommendation requirements in multi-turn scenarios? Third, can the lower-cost gpt-5.6-luna model provide results comparable to gpt-5.6-terra? Fourth, do work-level deduplication and novelty rules prevent repeated recommendations within a list and within the same dialog?

The contributions are as follows. A modular architecture is proposed for Ukrainian conversational book recommendation. Its NLU interface uses a strict structured frame, catalog grounding, and explicit unresolved-entity fields. Its dialog manager separates new searches, refinements, feedback, pending actions, and contextual references. Its retrieval policy unifies structured, semantic, hybrid, similar-book, author, and profile-based routes. The system applies edition-level normalization, rated-item exclusion, and recommendation-history filtering before producing a result. Finally, the study provides a normalized research dataset of approximately 5,500 books and a multi-level evaluation protocol. The code, prepared data, scenarios, and evaluation materials are available through a public repository [16].

RESEARCH METHODS

The experimental data layer contains a normalized local catalog of approximately 5,500 Ukrainian book records. A record can contain a title, description, publication metadata, language, type, page count, series, publisher, period, and links to one or more authors and categories. Separate entity tables reduce duplicated strings and support canonical identifiers.

The system uses three complementary storage forms. SQLite stores books, authors, categories, publishers, series, periods, users, ratings, and interactions. Many-to-many tables represent book-author and book-category relationships. This structure supports exact filters, joins, feedback persistence, hydration of recommendation identifiers, and verification of returned metadata.

Chroma stores vector representations of book text and user profiles. A book vector is generated from the indexed title-and-description representation. The catalog does not require a separate manually assigned tag for every mood or theme. A request such as "a book about war, memory, and loss" («книга про війну, пам'ять і втрату») can therefore be matched to descriptions through semantic retrieval. Chroma also stores a user vector

derived from rated books. The two vector uses are distinct: a semantic query represents the current request, whereas a profile vector represents accumulated preference.

StateRepository stores a per-user DialogState during an active session. It contains active constraints, semantic query text, current focus books and authors, recently mentioned and recommended books, selected and liked books, pending actions, pending slots, pending candidates, and the recommendation count. This state is operational, not documentary. It retains only information needed for future decisions.

Relational and vector storage are not interchangeable. SQL filtering is appropriate for a request such as "Ukrainian-language books published after 2015 with fewer than 300 pages" («україномовні книги після 2015 року до 300 сторінок»), because each condition has a verifiable catalog interpretation. Vector retrieval is appropriate for "a philosophical story about solitude" («філософська історія про самотність»), because the request describes meaning that cannot be expressed as a simple database predicate. A hybrid request uses both.

Each turn follows the staged architecture shown in Figure 1. The upper panel shows how a user utterance becomes a catalog-grounded ParsedUtterance. The lower panel shows how this frame updates the dialog state, selects a retrieval route, and becomes a grounded response. The separation is deliberate. LLM components interpret and verbalize language, whereas deterministic components validate catalog values, control state, choose retrieval operations, and enforce recommendation rules.

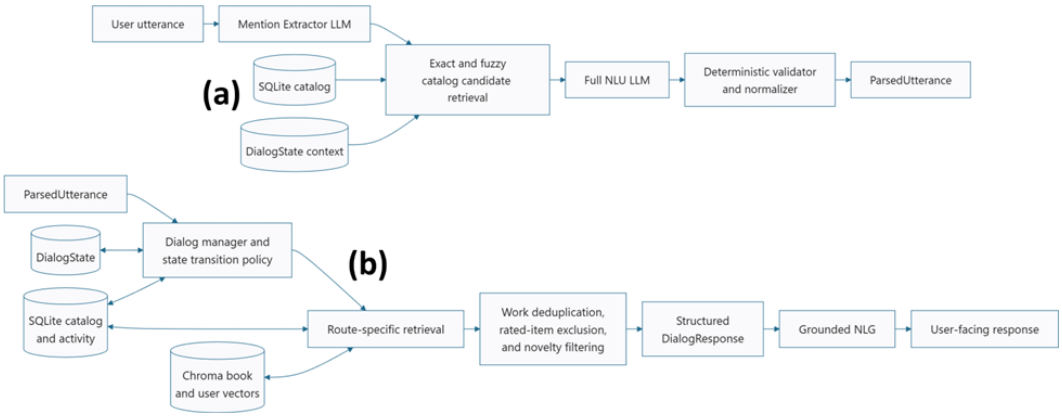

Fig. 1. Detailed architecture of the dialog-state RAG recommender: (a) catalog-grounded NLU pipeline; (b) dialog management, retrieval, and response generation

The orchestrating service implements this staged flow for each turn. It first constructs compact context from DialogState rather than relying on an unrestricted raw transcript as the sole source of memory. The current utterance and dialog context are processed by the NLU pipeline to produce a catalog-grounded ParsedUtterance. The dialog manager then updates the state and selects an action and retrieval route. Depending on the request, retrieval may use SQL, vector, hybrid, author, similar-book, or profile-based search. The resulting candidates undergo work-level deduplication, rated-item exclusion, and novelty filtering. Finally, the NLG component receives the structured DialogResponse together with hydrated catalog records and verbalizes the resulting decision.

The main NLU output is a ParsedUtterance. As shown in Figure 2, the ParsedUtterance encapsulates both the semantic query and structured constraints.

```
ParsedUtterance = {
  intent,
  structured_constraints: {
    authors, categories, series, publishers,
    languages, periods, types,
    year_from, year_to, pages_min, pages_max
  },
  semantic_query: { text },
  books,
  requested_count,
  unresolved_entities,
  raw_utterance
}
```

Fig. 2. ParsedUtterance structure with key fields

The supported intents include recommendation, similar-book search, book and author queries, positive and negative feedback, clarification answers, recommendation history, explanation, acknowledgment, greeting, and unknown requests. The schema deliberately separates referenced books from constraints because a title can be the target of feedback, an information query, or a similarity request without becoming a filter over new recommendations.

The NLU algorithm has six stages. First, the MentionExtractor proposes surface mentions. Second, exact category recovery scans for catalog phrases that the extractor may have omitted. Third, deterministic alias and fuzzy matching retrieve canonical candidates. Fourth, candidates from the current turn are ordered before candidates derived from dialog context. Fifth, the full NLU model generates a JSON frame constrained by the schema. Sixth, the validator checks allowed values, normalizes numeric bounds, removes duplicate concepts across fields, promotes exact catalog categories when justified, preserves unknown explicit mentions, and applies deterministic corrections for high-impact intent patterns.

This design addresses a recurrent failure mode of purely generative parsing: an explicit unknown entity must remain unknown. If the current focus author is Panteleimon Kulish and the user asks "what did Omar Khayyam write?" («що написав Омар Хаям?»), the system must not answer about Kulish merely because he is present in context. If Omar Khayyam is absent from the catalog, the author string is stored under unresolved_entities.authors and the downstream action is a grounded no-results response.

Structured and semantic meanings are also separated. "Classical prose" («класична проза») can map to a canonical category if that value exists. "Atmospheric and sad" («атмосферне і сумне») remains semantic text. The validator removes cross-field duplicates when a language, type, period, year, or page expression has already been resolved. This prevents one phrase from simultaneously acting as a hard filter and an unresolved error.

The dialog manager receives the user identifier and parsed frame. It returns a DialogResponse with an action (recommend, message, clarification, or no_results), recommendation identifiers, retrieval type, reason, result counts, payload, and pending-state fields.

Context resolution follows an explicit priority (Figure 3). A resolved entity in the current utterance has the highest priority. A morphological or alias match against compact context is considered next. An explicit unresolved entity is preserved instead of being overwritten. Pending candidates from a previous clarification can then be used, followed by the current focus for a genuinely implicit reference. If none is sufficient, the system asks a clarification question.

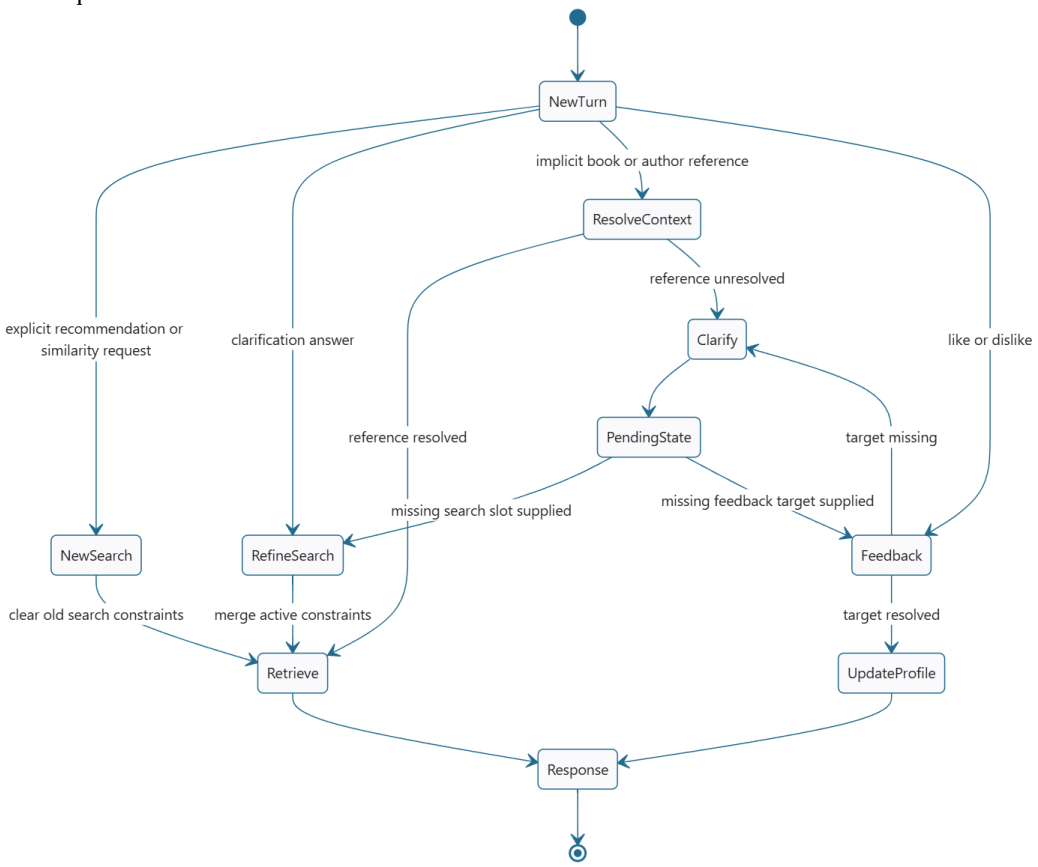

Fig. 3. Dialog-state lifecycle for new searches, refinements, contextual references, and pending actions

The state lifecycle distinguishes a new search from a refinement. An explicit recommend_book or similar_book request starts a new search and clears active search constraints, semantic text, and stale clarification state. It preserves user feedback and conversational focus where they remain useful. A clarification_answer instead merges new information into the active search. Thus, after "recommend contemporary books" («порадь сучасні книги»), the reply "preferably fantasy" («краще фентезі») retains the period and adds the category. By contrast, "now recommend four books for me" («тепер порадь чотири книги для мене») starts a profile request without inheriting the earlier fantasy filter.

Pending actions handle incomplete operations. If the user says "I liked it" («мені сподобалося») without an identifiable book, the manager sets pending_action=feedback_like, records the missing books slot, and asks for a title. A subsequent title is interpreted as the missing argument, the rating is stored, the user profile is rebuilt, and the interrupted action is completed.

The default output count is five. An explicit requested count is clamped to the inclusive range from 1 to 20. When more candidates are available than can be displayed, the manager still returns a ranked subset and records both total_found and requested_count. This lets NLG explain how many books matched and how many are shown.

Route selection is determined from the parsed request and state (Figure 4). Structured constraints invoke SQL candidate generation. A semantic query without structured constraints invokes vector search. Their combination produces a hybrid route: SQL restricts the candidate set and vector similarity ranks the remaining books. A named-book similarity request retrieves neighbors of the source book vector. An author-only similarity formulation returns books by the named author in the current implementation. It does not attempt to discover stylistically similar authors. An unconstrained recommendation uses the user profile if one exists, otherwise the system enters a cold-start clarification.

The user profile is an unnormalized weighted sum of rated-book embeddings:

$$u = \sum_{i=1}^{N} r_i \, e_i, \tag{1}$$

where e_i is the embedding of the i-th rated book and r_i is its feedback weight scaled to $[-2,2]$. Positive and negative feedback normally contribute $+2e_i$ and $-2e_i$, respectively. If the vector is missing, unusable, or is the zero vector, it is treated as absent.

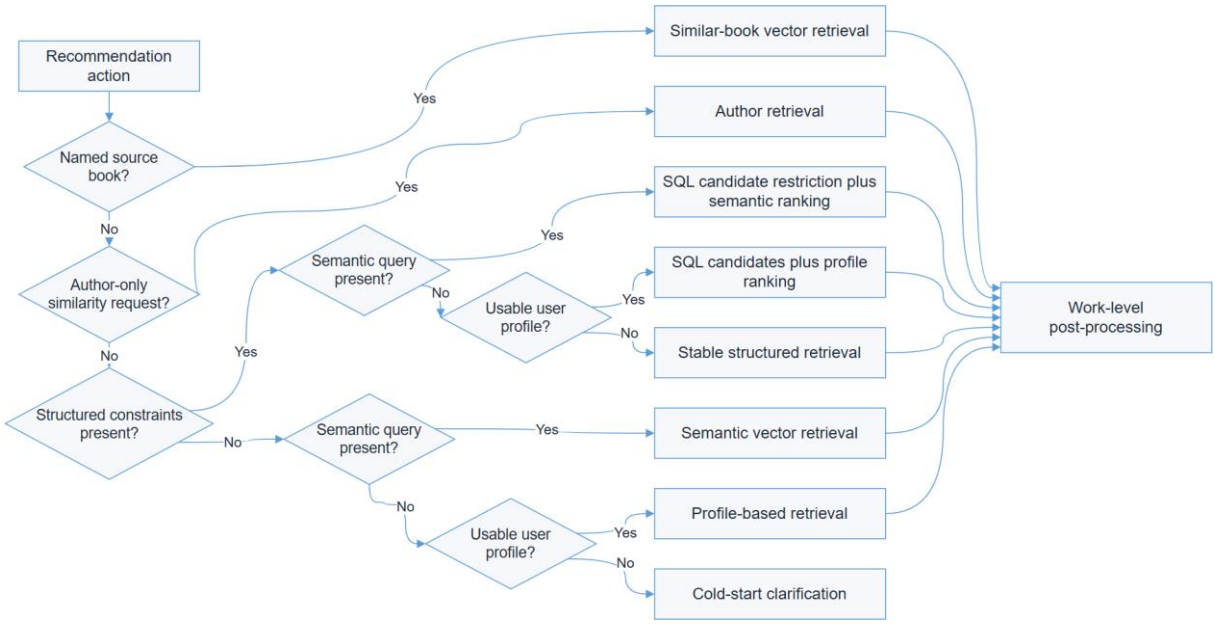

Fig. 4. Retrieval-route decision tree

In Chroma collections where proximity is measured using cosine distance, vector similarity is expressed as the corresponding distance metric:

$$d = 1.0 - \frac{\sum(A_i \times B_i)}{\sqrt{\sum(A_i^2)} \cdot \sqrt{\sum(B_i^2)}}, \tag{2}$$

where A and B are the compared vectors, and a smaller distance indicates greater vector proximity [8]. The corresponding cosine similarity is $s = 1 - d$. Ranking by ascending Chroma distance is therefore equivalent to ranking by descending cosine similarity. Cosine-based comparison is a common option in recommender vector spaces, although its behavior depends on the input representation [17].

All routes share a final post-processing stage. Alternative editions of the same work are collapsed into one work key. Rated works are excluded. Previously unseen works are prioritized over works already recommended to the same user, and previously recommended works are reused only if necessary to fill the requested count. The final payload therefore reflects not only similarity or metadata fit but also interaction history.

NLG receives the structured DialogResponse, the current utterance, and hydrated catalog records. For recommendation actions, it formats a readable list, explains the connection to the request using available fields, reports when only a subset of matches is shown, and may ask a relevant follow-up question. For clarification and no-results actions, it verbalizes the manager's pending fields or unresolved entities. Some high-impact messages, including metadata-resolution prompts and impossible active-constraint responses, bypass the generative model and remain deterministic.

The grounding boundary is important. Recommendation identifiers are fixed before NLG, and the component is instructed to verbalize only the books present in the structured payload. Titles, authors, and other factual metadata are supplied by the catalog. The generated text can vary linguistically, but the recommended set used by the application remains the output of the retrieval and post-processing pipeline.

The evaluation contains three complementary levels. First, the deterministic regression suite contains 90 tests. It covers NLU candidate handling and validation, Ukrainian aliases, contextual resolution, pending actions, search reset and refinement, cold start, repositories, profile vectors, retrieval, work deduplication, novelty, recommendation history, and NLG data contracts. All 90 tests passed. The suite exercises deterministic component contracts with mocks, temporary databases, in-memory repositories, and ephemeral vector storage where appropriate.

Second, standalone NLU was evaluated on 1,000 Ukrainian input utterances with expected frames. The set covers 11 intents and 22 scored constraint paths, including resolved and unresolved entities, numeric bounds, semantic text, referenced books, and requested counts. Each utterance is independent and contains no multi-turn context. The experiment compares rule/keyword, fuzzy n-gram, combined deterministic, and LLM-based parsers. The LLM configuration uses Luna for mention extraction and full-frame parsing. The archived inputs, outputs, and scorer support offline recomputation of the reported metrics.

Precision, recall, and F1 are calculated over extracted slot values. These measures are reported separately from intent accuracy and full-frame exact accuracy because they represent different error structures. Strict exact accuracy requires the entire normalized frame to match. Relaxed exact accuracy accepts spelling, inflection, and semantically equivalent free-text variants, while still requiring the correct intent, complete structured information, requested count, and unresolved-entity behavior.

Third, E2E evaluation contains 70 scenarios and 170 turns. Thirty-two scenarios are single-turn and 38 are multi-turn. They cover structured, semantic, hybrid, similar-book, author, and profile routes; requested-count boundaries; search refinement; cold start; feedback; recommendation history; novelty; contextual references; metadata resolution; and deliberately unsupported-feature probes. Each scenario receives isolated user, state, rating, interaction, and profile storage. The source vector store is copied to temporary storage, preventing evaluation from changing persistent development data.

Two E2E configurations were run with the same mention extractor, NLG component, scenarios, and storage setup. They differ in the full NLU model: gpt-5.6-terra and gpt-5.6-luna. The official model comparison positions Terra as a balance between intelligence and cost and Luna as optimized for cost-sensitive, high-volume workloads [18]. NLU is evaluated by strict and relaxed frame accuracy and by slot-level precision, recall, and F1. Dialog manager subset accuracy compares only the expected DialogResponse fields specified for a turn. Recommendation assertions are defined on 39 turns and can verify counts, unique works, exclusion rules, and metadata constraints.

The automated correctness measure for recommendation-bearing turns is recommendation-assertion accuracy:

$$A_{rec} = \frac{N_{passed\ recommendation\ turns}}{N_{turns\ with\ recommendation\ assertions}}, \tag{3}$$

which reports the share of designated turns that satisfy all scenario-specific recommendation conditions. It captures verifiable final-payload properties, including output count, work uniqueness, exclusion rules, and metadata constraints. This turn-level measure complements NLU and dialog manager scores by evaluating the recommendation delivered at the end of the pipeline.

Output diversity is characterized by the number of unique works, authors, and categories, the unique-work-to-placement ratio:

$$D_{work} = \frac{|W_{unique}|}{N_{placements}}, \quad (4)$$

and whether works are repeated within a list or across turns of the same scenario. The counts and ratio are computed directly from stored catalog identifiers and recommendation exports. Together, these study-specific descriptive indicators characterize output variety and repetition at the level of stored catalog identifiers and archived recommendation placements.

Finally, we manually inspected every recommendation-bearing turn in the archived exports: 94 for Terra and 91 for Luna. Each returned list was checked against the interpreted request, active constraints, scenario context, and catalog metadata. This manual inspection complements the formal assertions because a semantically appropriate request can admit more than one valid set of books.

RESEARCH RESULTS

Table 1 shows a large difference between deterministic parsing and the LLM-based pipeline. Luna achieved 98.3% intent accuracy, 88.7% constraint F1, 74.2% strict full-frame accuracy, and 88.0% relaxed full-frame accuracy. The combined deterministic parser improved precision over either simple baseline but recovered only 28.4% of expected constraint values.

The 13.8 percentage-point (pp) difference between strict and relaxed frame accuracy is informative. Many strict mismatches reflect alternative representations while the intended meaning remains recoverable. Ukrainian inflection and synonymy can produce equivalent surface forms. Similar or overlapping names among categories, series, periods, languages, and general thematic descriptions can also cause one phrase to be assigned to different slots.

Table 1

Standalone NLU parser comparison on 1,000 Ukrainian inputs

Parser type	Intent accuracy	Precision	Recall	F1-score	Strict exact	Relaxed exact
Rule/keyword	45.3%	39.2%	11.7%	18.0%	3.9%	4.8%
Fuzzy n-gram	45.9%	34.8%	19.7%	25.2%	7.8%	8.1%
Combined	52.1%	70.1%	28.4%	40.4%	11.6%	13.2%
LLM parser	98.3%	87.1%	90.3%	88.7%	74.2%	88.0%

Several examples illustrate the problem. "Modern novels" («сучасні романи») may be represented as the period Contemporary literature, the category Novels, or a combination of period and semantic text, depending on the exact catalog vocabulary. "Friendship and growing up" («дружба і дорослішання») is a semantic concept, but a parser can incorrectly promote it to a similarly named category and thereby change semantic retrieval into hybrid retrieval. Similar names shared by series and categories can produce the same type of cross-field mismatch.

Unknown entities create the inverse problem. If a requested title such as "Rubaiyat" («Рубаї») is absent from the local catalog, an approximate string match can be more harmful than no match. The correct behavior is to preserve the explicit title under unresolved entities and produce a no-results or clarification flow. This is why the evaluation includes unresolved values and why high field accuracy on empty slots alone would be misleading.

The error distribution also confirms the value of a staged NLU design. Errors can originate in mention extraction, candidate retrieval, full-frame interpretation, or local validation. Exact category recovery and deterministic cleanup correct some model omissions, but aggressive promotion can create false structured constraints. The relaxed metric therefore accepts linguistic equivalence without excusing a changed intent, a missing hard constraint, or an incorrectly grounded entity.

Table 2 compares Terra and Luna in the same 170-turn pipeline. The two configurations are close across all reported measures. Luna reached 87.65% strict and 92.94% relaxed NLU exact accuracy, compared with 84.71% and 90.59% for Terra. Luna also produced higher dialog manager subset accuracy. Terra achieved higher relaxed slot-level precision, recall, and F1. Both configurations passed 38 of 39 recommendation-assertion turns.

The results do not establish a universally superior model. Turn-level exact accuracy and micro-averaged slot F1 respond differently to clustered errors. A small number of frames containing several false-positive slots can reduce micro-F1 substantially while adding only a few failed turns. Conversely, one wrong high-impact intent can fail the entire turn even if most slots are empty and correct.

Table 2

End-to-end comparison of Terra and Luna full NLU configurations

Metric	Terra	Luna	Luna − Terra
Strict NLU exact	84.71%	87.65%	+2.94 pp
Relaxed NLU exact	90.59%	92.94%	+2.35 pp
Dialog manager accuracy	92.94%	96.47%	+3.53 pp
Recommendation assertions	97.44%	97.44%	0.00 pp
Relaxed constraint precision	94.85%	92.44%	−2.41 pp
Relaxed constraint recall	97.36%	96.92%	−0.44 pp
Relaxed constraint F1	96.09%	94.62%	−1.46 pp

The two models achieved comparable overall quality on this benchmark. Luna was adopted for subsequent use because the official model comparison positions it for cost-sensitive workloads and lists lower token pricing than Terra [18]. This choice lowers operating cost without an observed reduction in overall benchmark quality.

Dialog manager subset accuracy requires careful interpretation. The manager's state-transition and action-selection rules are deterministic for a given normalized input frame and current state. However, E2E scoring compares the response produced after real NLU. A wrong NLU intent, inherited false constraint, or unresolved contextual reference can therefore produce a downstream DialogResponse different from gold even though the manager applied its defined rule to the state it received. The 90 passing regression tests separately verify deterministic transition and retrieval contracts under controlled inputs.

The most frequent E2E errors involved category-versus-semantic interpretation, contextual author resolution, and inherited constraints in long refinements. For example, interpreting "cozy winter reading" («затишне зимове читання») as a catalog category changed the route from semantic to hybrid. In another long scenario, a short "show two" («покажи дві») reply inherited additional language, period, and type constraints. The resulting failure propagated because later turns used the modified state.

The recommendation audit is summarized in Table 3. Terra produced 430 placements across 94 turns and Luna produced 423 placements across 91 turns. Alternate editions were normalized into works. Neither run contained duplicate works inside a list, and neither repeated a work across turns of the same scenario.

Table 3

Recommendation output and diversity in the end-to-end runs

Metric	Terra	Luna
Turns containing recommendations	94	91
Book placements	430	423
Unique book IDs	232	219
Unique works	227	214
Unique-work / placement ratio	52.79%	50.59%
Unique authors	207	188
Unique categories	112	117
Duplicate works within a list	0	0
Repeated works across turns in one scenario	0	0

Recommendation-assertion accuracy was identical at 97.44%. The single failed assertion in each run did not identify an invented book inside a returned list. In the Terra run, a cold-start sequence did not reach the expected recommendation after two liked books. In the Luna run, inherited false constraints caused a clarification instead of the expected two-book list. Thus, the failed checks were availability-of-expected-output failures caused by upstream state or NLU deviations.

Our manual inspection covered all 185 recommendation-bearing turn outputs from the two archived runs. Each returned list was checked against the interpreted query, active constraints, scenario context, and available catalog metadata. The inspection identified no book-level mismatches among the lists actually returned. This result is consistent with the architecture: once an input frame and state select a route, the manager retrieves catalog identifiers and applies explicit filters. It does not generate titles.

The unique-work ratio is slightly higher for Terra, while Luna covers more categories despite fewer recommendation turns. These differences should not be attributed to the NLU model alone. A changed intent or route alters the candidate space, and the total number of recommendation-bearing turns affects the number of opportunities for global coverage. The strongest route-independent result is the absence of work-level repetition within lists and scenarios, which directly validates the implemented deduplication and novelty rules.

Several multi-turn scenarios were intentionally more complex than a typical short recommendation exchange. They combine refinement, focus changes, cold-start recovery, feedback, contextual pronouns, requested-count changes, and unsupported requests. Their purpose is to expose interactions among components under stress.

The long scenarios reveal a characteristic cascade. If one turn is misinterpreted, the next turn receives a state that is internally valid but differs from the user's intended state. Because the system cannot know that the previous interpretation was wrong, a short elliptical reply may reinforce the deviation. This explains why some later dialog manager mismatches are consequences of an earlier NLU decision and not independent policy failures.

In a live conversation, many such cascades can be interrupted by a corrective user utterance. A reader can restate an author, remove a constraint, clarify that a phrase is a theme and not a category, or begin an explicit new search. During development and scenario inspection, the state display served as a diagnostic aid by exposing misinterpreted constraints. This observation motivates a user-facing state preview alongside the conversation. The preview could show active filters, semantic preference, focused book or author, and any pending clarification. A user could then identify a mistaken language, period, category, or focus before it affects several later turns.

This implication is consistent with the broader finding that natural language and explicit interaction controls can complement one another in conversational recommendation [10]. In the proposed system, state visibility would not expose internal model reasoning. It would expose only the structured, operational interpretation already used by retrieval. Such transparency can be treated as an error-management mechanism and as a basis for future user-centered evaluation

CONCLUSIONS AND PROSPECTS FOR FURTHER RESEARCH

This paper presented a RAG-based dialog-state conversational recommender for Ukrainian books. The architecture assigns different responsibilities to language models and deterministic services: NLU converts an utterance into a grounded frame; validation checks the frame against catalog candidates; the dialog manager maintains explicit state and selects an action; retrieval returns book identifiers through structured, semantic, hybrid, author, similar-book, or profile routes; and NLG verbalizes the resulting payload. This separation supports contextual interaction without delegating catalog membership or recommendation selection to free-form generation.

The implementation is supported by a research catalog of approximately 5,500 books and a multi-level evaluation. All 90 deterministic regression tests passed. On 1,000 standalone NLU inputs, the LLM parser achieved 98.3% intent accuracy, 88.7% constraint F1, and 88.0% relaxed full-frame accuracy. In 170 E2E turns, the two NLU configurations achieved 90.59–92.94% relaxed frame accuracy, 92.94–96.47% dialog manager subset accuracy, and 97.44% recommendation-assertion accuracy. Both configurations avoided duplicate works within lists and repeated works across turns of a scenario.

Terra and Luna exhibited different error distributions but comparable overall system quality. Luna showed slightly higher turn-level exact and dialog manager subset accuracy, while Terra showed slightly higher slot-level F1. Luna was selected for subsequent use because its measured pipeline quality was comparable and the official model comparison positions it as the lower-cost option.

The main scientific contribution is not the use of RAG by itself. It is the integration of a strict Ukrainian NLU frame, explicit unresolved entities, deterministic dialog-state transitions, route-specific retrieval, grounded NLG, and work-level novelty within one testable system. The evaluation also demonstrates why component and E2E metrics must be reported together: the deterministic manager can follow its policy correctly for a state that was made incorrect by an earlier interpretation.

Further work will implement a user-visible state preview and controls for correcting active constraints, focus, and semantic preference. Later studies can then evaluate how state visibility affects perceived relevance, effort, trust, and satisfaction. Further retrieval experiments will compare embedding and reranking methods on the same catalog. Context reset, contextual entity validation, and negative-preference handling are the principal algorithmic extensions.

REFERENCES

1. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99
2. Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331–370. https://doi.org/10.1023/A:1021240730564

3. Gao, C., Lei, W., He, X., de Rijke, M., & Chua, T.-S. (2021). Advances and challenges in conversational recommender systems: A survey. *AI Open, 2*, 100–126. https://doi.org/10.1016/j.aiopen.2021.06.002
4. Jannach, D., Manzoor, A., Cai, W., & Chen, L. (2021). A survey on conversational recommender systems. *ACM Computing Surveys, 54*(5), 1–36. https://doi.org/10.1145/3453154
5. Williams, J. D., Raux, A., Ramachandran, D., & Black, A. (2013). The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference* (pp. 404–413). https://aclanthology.org/W13-4065/
6. Radchenko, V., & Drushchak, N. (2025). Improving named entity recognition for low-resource languages using large language models: A Ukrainian case study. In *Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP 2025)* (pp. 27–35). https://doi.org/10.18653/v1/2025.unlp-1.3
7. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474. https://doi.org/10.48550/arXiv.2005.11401
8. Chroma. (n.d.). *Configure collections*. Chroma Docs. Retrieved August 4, 2026, from https://docs.trychroma.com/docs/collections/configure
9. Li, R., Ebrahimi Kahou, S., Schulz, H., Michalski, V., Charlin, L., & Pal, C. (2018). Towards deep conversational recommendations. *Advances in Neural Information Processing Systems, 31*, 9748–9758. https://doi.org/10.48550/arXiv.1812.07617
10. Iovine, A., Narducci, F., & Semeraro, G. (2020). Conversational recommender systems and natural language: A study through the ConveRSE framework. *Decision Support Systems, 131*, Article 113250. https://doi.org/10.1016/j.dss.2020.113250
11. Wang, X., Liu, J., & Duan, J. (2024). Improved conversational recommender system based on dialog context. *Natural Language Engineering, 30*(6), 1210–1228. https://doi.org/10.1017/S1351324923000451
12. Jannach, D. (2023). Evaluating conversational recommender systems: A landscape of research. *Artificial Intelligence Review, 56*, 2365–2400. https://doi.org/10.1007/s10462-022-10229-x
13. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5–53. https://doi.org/10.1145/963770.963772
14. Kaminskas, M., & Bridge, D. (2017). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems, 7*(1), 1–42. https://doi.org/10.1145/2926720
15. Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (pp. 109–116). https://doi.org/10.1145/2043932.2043955
16. Gurdel. (2026). *RAG-based dialog-state conversational recommender for Ukrainian books* [Computer software]. GitHub. Retrieved August 17, 2026, from https://github.com/Gurdel/RAG-based-dialog-state-conversational-recommender-for-Ukrainian-books
17. Fkih, F. (2022). Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison. *Journal of King Saud University – Computer and Information Sciences, 34*(9), 7645–7669. https://doi.org/10.1016/j.jksuci.2021.09.014
18. OpenAI. (n.d.). *Compare models.* OpenAI API. Retrieved August 15, 2026, from https://developers.openai.com/api/docs/models/compare

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023

Марченко Олександр Олександрович
д.ф.-м.н., професор, професор кафедри математичної інформатики
Київський національний університет імені Тараса Шевченка, Київ, Україна
ORCID: 0000-0002-5408-5279
omarchenko@univ.kiev.ua

Шевченко Максим Олексійович
аспірант кафедри математичної інформатики
Київський національний університет імені Тараса Шевченка, Київ, Україна
ORCID: 0009-0008-5104-9767
maksym_shevchenko@knu.ua

ДІАЛОГОВА РЕКОМЕНДАЦІЙНА СИСТЕМА ДЛЯ УКРАЇНСЬКИХ КНИГ НА ОСНОВІ ГЕНЕРАЦІЇ З ДОПОВНЕНИМ ПОШУКОМ ТА МОДЕЛІ СТАНУ ДІАЛОГУ

Анотація. Пошук книг у цифрових каталогах за допомогою запитів природною мовою потребує поєднання точних обмежень каталогу, менш формалізованих тематичних уподобань і контексту, накопиченого протягом кількох реплік діалогу. Метою цього дослідження є розроблення та експериментальне оцінювання діалогової рекомендаційної системи для україномовних запитів щодо книг на основі генерації, доповненої пошуком, та з явним станом діалогу. Запропонована система працює з нормалізованим дослідницьким каталогом приблизно з 5 500 українських книг і розділяє розуміння природної мови, детерміновану політику діалогу, структурований і векторний пошук, ранжування на основі профілю користувача та обґрунтовану даними генерацію відповідей. Структурований проміжний фрейм відокремлює сутності каталогу від семантичних уподобань і зберігає нерозв'язані згадки, а менеджер діалогу зберігає активні обмеження, фокус, очікувані дії, зворотний зв'язок та історію рекомендацій. Залежно від запиту система обирає структурований, семантичний, гібридний, авторський, профільний пошук або пошук схожих книг. Оцінювання проводилося на трьох рівнях: 90 детермінованих регресійних тестів, окремому оцінюванні розуміння природної мови на 1 000 україномовних вхідних висловлюваннях та 70 наскрізних сценаріях із 170 реплік. Усі регресійні тести було успішно пройдено. Парсер на основі великої мовної моделі досяг 98,3% точності визначення наміру, 88,7% F1-міри для обмежень і 88,0% точності повного фрейму за пом'якшеним критерієм. Під час наскрізного оцінювання дві протестовані конфігурації модуля розуміння природної мови досягли 90,59–92,94% точності фрейму за пом'якшеним критерієм, 92,94–96,47% точності підмножини полів менеджера діалогу та 97,44% точності перевірок рекомендацій. Ручна перевірка 185 вихідних відповідей, що містили рекомендації, не виявила жодної невідповідності на рівні книг у фактично повернутих списках. Жодна з конфігурацій не створювала дублікатів творів у межах одного списку рекомендацій і не повторювала той самий твір у різних репліках одного сценарію. Результати показують, що явний стан діалогу, детерміноване керування пошуком та обґрунтована даними генерація дають змогу формувати відтворювані контекстно-залежні рекомендації. Вибір книг і перевірка їх наявності в каталозі при цьому не покладаються на вільну генерацію. Архітектура та протокол оцінювання можуть бути повторно використані для інших задач діалогових рекомендацій, обґрунтованих даними каталогу.

Ключові слова: діалогова рекомендаційна система; генерація з доповненням пошуком; стан діалогу; розуміння природної мови; гібридний пошук; векторний пошук; оцінювання рекомендацій.

REFERENCES

1. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99
2. Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331–370. https://doi.org/10.1023/A:1021240730564

3. Gao, C., Lei, W., He, X., de Rijke, M., & Chua, T.-S. (2021). Advances and challenges in conversational recommender systems: A survey. *AI Open, 2*, 100–126. https://doi.org/10.1016/j.aiopen.2021.06.002
4. Jannach, D., Manzoor, A., Cai, W., & Chen, L. (2021). A survey on conversational recommender systems. *ACM Computing Surveys, 54*(5), 1–36. https://doi.org/10.1145/3453154
5. Williams, J. D., Raux, A., Ramachandran, D., & Black, A. (2013). The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference* (pp. 404–413). https://aclanthology.org/W13-4065/
6. Radchenko, V., & Drushchak, N. (2025). Improving named entity recognition for low-resource languages using large language models: A Ukrainian case study. In *Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP 2025)* (pp. 27–35). https://doi.org/10.18653/v1/2025.unlp-1.3
7. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474. https://doi.org/10.48550/arXiv.2005.11401
8. Chroma. (n.d.). *Configure collections*. Chroma Docs. Retrieved August 4, 2026, from https://docs.trychroma.com/docs/collections/configure
9. Li, R., Ebrahimi Kahou, S., Schulz, H., Michalski, V., Charlin, L., & Pal, C. (2018). Towards deep conversational recommendations. *Advances in Neural Information Processing Systems, 31*, 9748–9758. https://doi.org/10.48550/arXiv.1812.07617
10. Iovine, A., Narducci, F., & Semeraro, G. (2020). Conversational recommender systems and natural language: A study through the ConveRSE framework. *Decision Support Systems, 131*, Article 113250. https://doi.org/10.1016/j.dss.2020.113250
11. Wang, X., Liu, J., & Duan, J. (2024). Improved conversational recommender system based on dialog context. *Natural Language Engineering, 30*(6), 1210–1228. https://doi.org/10.1017/S1351324923000451
12. Jannach, D. (2023). Evaluating conversational recommender systems: A landscape of research. *Artificial Intelligence Review, 56*, 2365–2400. https://doi.org/10.1007/s10462-022-10229-x
13. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5–53. https://doi.org/10.1145/963770.963772
14. Kaminskas, M., & Bridge, D. (2017). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems, 7*(1), 1–42. https://doi.org/10.1145/2926720
15. Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (pp. 109–116). https://doi.org/10.1145/2043932.2043955
16. Gurdel. (2026). *RAG-based dialog-state conversational recommender for Ukrainian books* [Computer software]. GitHub. Retrieved August 17, 2026, from https://github.com/Gurdel/RAG-based-dialog-state-conversational-recommender-for-Ukrainian-books
17. Fkih, F. (2022). Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison. *Journal of King Saud University – Computer and Information Sciences, 34*(9), 7645–7669. https://doi.org/10.1016/j.jksuci.2021.09.014
18. OpenAI. (n.d.). *Compare models.* OpenAI API. Retrieved August 15, 2026, from https://developers.openai.com/api/docs/models/compare

Отримано редакцією журналу / Received: 11.02.26
Прорецензовано / Revised: 26.02.26
Схвалено до друку / Accepted: 24.09.26