International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Retrieval-Augmented, Multi-Agent AI for Academic Administration Plan Generation: A Survey in the Context of the Pedagogy Pathfinder Network (PPN)

Kirtana Singh, Pooja Divekar, Pradnya Khanse, Dr. Neha Kudu
Department of Information Technology,
Vidyalankar Institute of Technology Mumbai, Maharashtra, India
Corresponding author e-mail: neha.kudu@vit.edu.in

Abstract—Preparing an Academic Administration Plan (AAP) for each course in each semester is a heavy documentation task for engineering faculty working under Outcome-Based Education (OBE), NEP 2020 and National Board of Accreditation (NBA) requirements; faculty-reported estimates exceed 40 hours per course. This paper surveys 17 studies published between 2020 and 2026 that bear on automating such documentation with large language models (LLMs). The studies were drawn from arXiv, IEEE Xplore, the ACL Anthology, the ACM Digital Library and SpringerLink and grouped into four themes: retrieval-augmented generation (RAG), LLM-based multi-agent orchestration, AI-assisted Bloom's taxonomy and CO–PO mapping, and AI-assisted lesson planning, assessment and grading. Each study is compared by technique, dataset or domain, strengths and limitations, and related to the Pedagogy Pathfinder Network (PPN), a three-layer, 17-agent framework proposed by the authors. The comparison shows that the individual components are well studied, that outcome-mapping work has concentrated on classifying text that already exists rather than generating it, and that none of the reviewed studies combines document-grounded generation, agent orchestration and automated compliance checking for Indian accreditation documents. Open gaps concern integration and compliance validation, small single-institution datasets, error propagation and scalability, and the lack of measured review effort. The survey does not experimentally validate PPN, and figures reported for it are treated as unverified estimates. Future directions include shared multi-institution benchmarks and comparison with manually prepared plans.

Keywords: Retrieval-Augmented Generation, Multi-Agent Systems, Outcome-Based Education, NBA Accreditation, Bloom's Taxonomy, Academic Administration

I. INTRODUCTION

A. Background and Motivation

Outcome-Based Education (OBE) has become central to how engineering programmes in India are planned, documented and accredited. Under the National Education Policy (NEP) 2020 and the National Board of Accreditation (NBA) framework, each course is documented through an Academic Administration Plan (AAP) that records its objectives, its course outcomes (COs), the mapping of those outcomes to programme outcomes (POs) and programme-specific outcomes (PSOs), and the associated lesson, assignment and assessment plans. Faculty members report that preparing an AAP takes more than 40 hours per course per semester, which displaces time from teaching, research and student mentoring. This figure is a faculty-reported estimate carried over from the authors' earlier survey report; it has not been measured in a controlled study.

Two strands of recent AI research are relevant to this burden. Retrieval-augmented generation (RAG) conditions a language model on retrieved reference documents so that its output can be traced to a source [1]–[4]. LLM-based multi-agent systems divide a complex task among specialised agents that plan, produce and check one another's work [5]–[8]. Separately, language models and classical machine learning have been applied to educational outcomes: Bloom's-taxonomy alignment and CO–PO mapping [9]–[13], and lesson planning, assessment and grading [14]–[17].

The Pedagogy Pathfinder Network (PPN) is a framework proposed by the authors that sits at the intersection of these strands. It accepts a course syllabus and a marking scheme as input, drafts an NBA-oriented AAP through a 17-agent pipeline built on a RAG backbone, checks the draft with an automated evaluator, and requires faculty approval of every section before the final documents are assembled. This paper is not an evaluation of PPN. It surveys the literature on which PPN's design rests, with the aim of showing where published work supports that design and where it does not.

B. Research Problem and Need for the Survey

Published tools for CO–PO mapping are mostly built around a single NLP or machine-learning classifier [12], [13], whereas the best-known multi-

agent frameworks, AutoGen, MetaGPT and ChatDev, were developed and evaluated on coding, mathematical problem solving and general dialogue [5]–[7]. Among the studies examined here, none addresses the combined task of generating rule-bound, structured accreditation documents from institutional source material. Before such a pipeline can be assessed, it is necessary to understand what each of its component techniques has and has not been shown to do, which assumptions transfer from one setting to another, and which questions remain open. This survey was undertaken for that purpose.

C. Objectives and Research Questions

The objectives of the survey are:

- to review recent literature, drawn mainly from 2021–2026 with one earlier foundational paper, on retrieval-augmented generation, LLM-based multi-agent orchestration, and AI-supported outcome-based education and accreditation tasks;
- to classify and compare the reviewed studies by technique, dataset or domain, strengths and limitations;
- to identify technical, dataset, performance and deployment gaps relevant to automated AAP generation; and
- to place the PPN pipeline within this literature and outline directions for future work.

These objectives are examined through three research questions. *RQ1:* Which techniques have been reported for each of the four thematic areas, and on what data? *RQ2:* How do the reviewed studies differ in the way they are evaluated, and what do their results establish? *RQ3:* What gaps remain for generating compliance-aware accreditation documents in Indian engineering education, and how does PPN relate to them?

D. Contributions and Organization

The survey makes four contributions. It provides a structured comparison of 17 studies across four themes; it separates OBE-automation work that classifies or maps existing text from work that generates new content; it identifies four categories of research gap with the evidence behind each; and it proposes an evaluation agenda. The description of PPN in Section VI is of a system designed by the authors and is kept distinct from these survey contributions.

Section II describes the review method. Section III introduces the technical background. Section IV reviews the literature by theme, and Section V compares the studies. Section VI describes PPN and relates it to the literature. Sections VII and VIII discuss research gaps and future directions, Section IX states the limitations of the review, and Section X concludes.

II. REVIEW METHODOLOGY

A. Sources and Search Terms

The literature was gathered primarily from arXiv, which carries much of the fast-moving work on LLMs and agents, and from IEEE Xplore, supplemented by the ACL Anthology, the ACM Digital Library and SpringerLink. Google Scholar and ResearchGate were used to locate peer-reviewed versions of preprints. This selection followed the sources recommended in the academic programme's guidelines (IEEE, ACM, Springer, Elsevier, Scopus and Web of Science). The search terms were: retrieval-augmented generation, RAG survey, retrieval-augmented text generation, LLM multi-agent systems, multi-agent conversation framework, agent orchestration pipeline, course outcome mapping, outcome-based education automation, NBA accreditation AI, AI lesson-plan generation, AI rubric grading, and AI assignment grading in higher education.

B. Selection Criteria

A paper was considered if it was published between 2020 and 2026, with preference for the most recent five years; if it appeared in a peer-reviewed venue such as AAAI, ACL, IJCAI, ACM SIGKDD, ACM CHI/CSCW, an IEEE conference or the Springer AIED proceedings, or was a widely cited arXiv preprint representing foundational or state-of-the-art work; and if it related clearly to at least one of three technical pillars: retrieval-grounded generation, multi-agent task decomposition, or AI-based processing of educational outcomes (CO–PO mapping, Bloom's tagging, lesson planning, rubric-based grading). Author names, venues and years were cross-checked against arXiv abstract pages, IEEE Xplore, the ACL Anthology, SpringerLink and the ACM Digital Library so that each reference can be traced to its source. General LLM papers with no educational angle, and vendor material about NBA or OBE software with unclear authorship and no listing in a recognised research database, were excluded. General chatbot tutoring systems and plagiarism-detection tools were treated as out of scope.

C. Screening and Thematic Grouping

The search queries returned a large pool of candidates. Screening against the criteria above reduced it to 17 studies, which were then arranged into four clusters: RAG for document-grounded generation (four studies), LLM-based multi-agent orchestration (four), AI-assisted OBE, Bloom's taxonomy and CO–PO mapping (five), and AI-assisted lesson planning, assessment and grading (four). The clusters correspond to the layers of the PPN design discussed in Section VI. Fig. 1 summarises the procedure as documented.

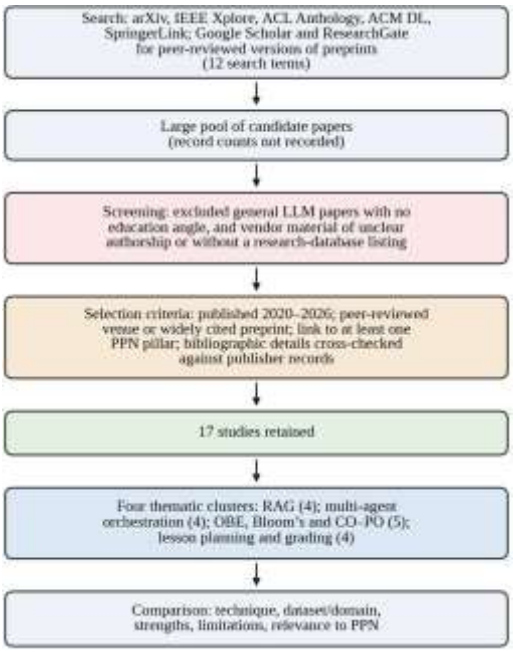


Fig. 1. Survey procedure as documented in the source report. Conceptual flowchart; record counts at each stage were not recorded.

D. Comparison Method and Procedural Limits

Each study was summarised by its technique, dataset or application domain, strengths and limitations; relevance to PPN was added for this paper. The type of evidence offered (survey, conceptual framework, or system with reported evaluation) was recorded so that findings of different strength are not treated alike. The procedure is a structured literature survey. The numbers of records retrieved and excluded at each stage, the dates of the searches and the number of screeners were not recorded, no formal quality-appraisal instrument was applied, and the paper does not claim PRISMA compliance or a fully reproducible systematic review. Section IX discusses the consequences.

III. BACKGROUND AND TECHNICAL FOUNDATIONS

A. Retrieval-Augmented Generation

A language model's knowledge is fixed in its parameters at training time and cannot be inspected. RAG addresses this by retrieving passages from an external corpus and supplying them as context for generation, so that statements can be tied to identifiable text [1]. A typical pipeline splits source documents into segments, converts them to vector embeddings, retrieves the segments closest to a query and passes them to the generator. Gao et al. [2] distinguish naive RAG (one retrieval pass), advanced RAG (additional pre- and post-retrieval processing) and modular RAG (composable retrieval and generation components). For document-generation tasks the practical question is whether retrieval works well on the structure of the source material, for example prose syllabi versus tabular marking schemes, a point taken up by work on non-text modalities [3]. Fig. 2 shows the generic workflow as it applies to PPN.

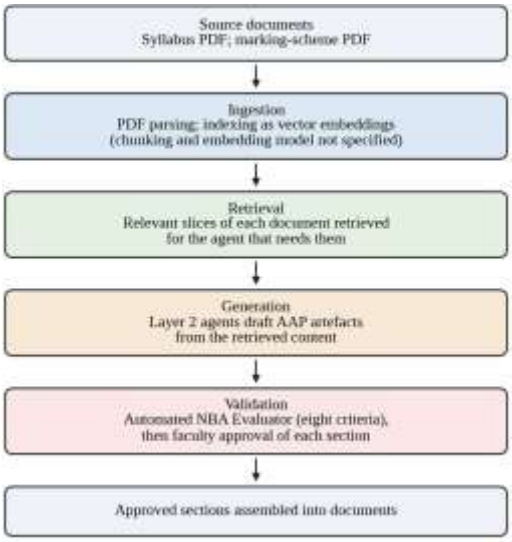


Fig. 2. Conceptual RAG workflow mapped to PPN's documented stages. Retrieval details (chunking, retriever, number of passages) are not specified in the source.

B. LLM-Based Multi-Agent Systems and Orchestration

In a multi-agent system an LLM is given a role, instructions and sometimes tools, and several such agents cooperate on a task. Frameworks differ in what they treat as the unit of coordination: a conversation between agents [5], a finished artefact handed from one role to the next [6], or a staged process [7]. Two recurring concerns are the growth of communication cost as agents are added [8] and the propagation of an early mistake through later stages [6]. Orchestration, the way agents are sequenced and connected, is therefore a design decision with consequences for reliability and cost.

C. Outcome-Based Education and Bloom's Taxonomy

In OBE a programme defines what graduates should be able to do (programme outcomes, with programme-specific outcomes for a given discipline) and each course states course outcomes that contribute to them. Course outcomes are normally written as measurable statements whose verbs correspond to levels of Bloom's taxonomy, which orders cognitive skills from recall through to creation. Tagging a statement with a Bloom's level, or generating assessment items at a chosen level, are therefore routine tasks in outcome-based course design [9], [10].

D. CO–PO–PSO Mapping and Accreditation Documentation

CO–PO–PSO mapping records which programme outcomes each course outcome supports, and underlies attainment calculations and NBA annexures. Automated approaches reviewed here treat it as a text-matching or classification problem over outcome statements that already exist [12], [13]. Generating the outcome statements themselves from a raw syllabus is a different problem, because the system must write the text before it can map it. This distinction between classifying and generating recurs throughout the survey.

E. AI-Assisted Lesson Planning and Assessment

Lesson plans, assignments, rubrics and graded feedback are the generative artefacts of an AAP. The reviewed work varies in how it controls them: through self-critique and refinement [14], expert curation [15], retrieval of rubric criteria and exemplar answers [16], or prompting alone [17].

IV. CRITICAL LITERATURE REVIEW

A. RAG for Document-Grounded Generation

Lewis et al. [1] combined a dense passage retriever with an encoder–decoder generator and evaluated it on knowledge-intensive tasks such as open-domain question answering over Wikipedia. Retrieval was a single pass with a fixed number of passages and no iterative refinement, a limitation that matters when a document must draw on several scattered sources. Three years later the volume of RAG variants had grown enough for Gao et al. [2] to organise them into naive, advanced and modular paradigms, which has become a common vocabulary for describing retrieval designs. Zhao et al. [3] extended the view from text to code, images and other modalities, noting that multi-modal retrieval remains computationally costly. Fan et al. [4], writing for the ACM SIGKDD audience, mapped the open questions in combining retrieval with LLMs and observed that few evaluations use real institutional document pipelines.

Read together, the four papers form a progression rather than four separate contributions: an architecture [1], a taxonomy of its variants [2], an extension beyond prose [3], and a call for testing in real domains [4]. Three of the four are surveys, so the group supplies design vocabulary and a list of open problems more than evidence about accreditation documents. The relevance to PPN lies in the shape of its inputs: a mostly prose syllabus and a mostly tabular marking scheme, from which different slices must be retrieved for different agents. The authors regard this as closer to the advanced or modular end of Gao et al.'s scale than to a single naive pass; none of the reviewed work tests that proposition.

B. LLM-Based Multi-Agent Orchestration

AutoGen [5] offers customisable, conversable agents that can use tools and execute code, with optional human participation. Its generality is also its cost: the reported coordination overhead grows with the number of agents. MetaGPT [6] takes a stricter line by encoding standardised operating procedures into agent prompts and having agents exchange structured outputs, which is intended to stop small errors from growing as they pass between agents. ChatDev [7] simulates a software company in which agents acting as designer, programmer and tester move work through a chat-based sequence of stages. The survey by Guo et al. [8] places these systems in a wider taxonomy and reports that fully connected communication between agents scales poorly.

The frameworks coordinate at different levels. AutoGen coordinates a conversation among fairly generic agents; MetaGPT coordinates the hand-off of defined artefacts; ChatDev coordinates stages. PPN's second layer is closest to the MetaGPT pattern, since each of its eight generation agents is responsible for one defined artefact rather than for open-ended negotiation, and its layered arrangement is consistent with the scaling concern raised by Guo et al. These are correspondences of design, not demonstrated advantages. All four studies were evaluated on software engineering, mathematics, question answering or general task settings, and the table of limitations in Section V shows that none was tested on structured educational documents.

C. AI-Assisted OBE, Bloom's Taxonomy and CO–PO Mapping

This cluster is the closest to the day-to-day operations of PPN's generation agents. Elkins et al. [9] had LLMs generate quizzes targeted at specific Bloom's levels and asked teachers to judge them; the teachers rated the generated quizzes slightly higher without a reported loss of quality. Waheed et al. [10] approached Bloom's levels from the opposite direction with BloomNet, a transformer-based classifier for learning-outcome statements that is reported to generalise well both in distribution and out of distribution; it classifies but does not generate. Luo et al. [11] rethink the taxonomy itself for the LLM era, extending it to information-literacy assessment; the work is a conceptual framework with limited large-scale validation, and its value to PPN is indirect, as an indication that the taxonomy is being treated as a design tool.

Two studies address CO–PO mapping directly in the Indian context. Sengupta and Das [12] combine NLP and machine learning to map course outcomes to programme outcomes in engineering curricula, and their results depend on how clearly the outcomes are phrased. Swamy et al. [13] present an SVM-based mapping framework set explicitly in the NEP 2020 context; the evaluation involves a small dataset and a

single algorithm. Both are directly comparable to the function of PPN's CO–PO–PSO Mapper.

Seen as a group, the cluster concentrates on classifying or mapping outcome statements that someone has already written [10], [12], [13]. Elkins et al. [9] generate assessment items, but from specified Bloom's levels rather than from a raw syllabus. What the cluster does not report is generation of the outcome statements themselves from an unstructured syllabus, which is the first task of PPN's CO Generator.

D. AI-Assisted Lesson Planning, Assessment and Grading

The fourth cluster is generative. Zheng et al. [14] produced each component of a lesson plan with RAG and then had the model critique and refine its own draft; against criteria set by human teachers, the self-critique step improved plan quality over single-pass generation across more than eighty mathematics topics in elementary grades. The suggestion for PPN is that a critique-and-refine step could be added to its Lesson Planner, although the evidence comes from school mathematics, not higher-education course structures.

Shiksha Copilot [15] is the largest deployment reviewed. It supports lesson planning in government schools in Karnataka through a human-in-the-loop pipeline in which curators review AI-drafted plans before teachers adapt them, and its study involved 1,043 teachers and 23 curators. Its relevance to PPN is the educator-review pattern and the Indian setting; its cost is a curator layer that may limit scale. Vatankhah Barenji et al. [16] ground LLM feedback in retrieved rubric criteria and exemplar answers and report 94–99% agreement with human graders on 701 student essays, which, if confirmed, would suggest that retrieval grounding can support human-level consistency in assessment tasks such as those handled by PPN's Rubric Builder and Assessment Planner. Yeung et al. [17] used the opposite strategy, zero-shot prompting without retrieval or fine-tuning, and found that students reported more positive motivation. Their result suggests that some parts of an AAP may not need full retrieval, but a zero-shot design may be less consistent than retrieval-grounded or fine-tuned graders.

These studies differ more in how output is controlled than in what is generated: self-critique [14], curator review [15], rubric retrieval [16], or none [17]. They also differ in the kind of evidence offered, from teacher-defined criteria and deployment studies to agreement statistics and student-reported motivation, so their results cannot be ranked on a common scale.

V. COMPARATIVE ANALYSIS

TABLE I summarises all 17 studies by technique, domain, contribution, strengths, limitations and relevance to PPN. Entries for strengths and limitations are as reported by the respective authors or as judged in the authors' earlier survey report; they have not been independently re-evaluated.

TABLE I. COMPARATIVE ANALYSIS OF THE 17 REVIEWED STUDIES

Study, year	Technique / approach	Dataset / domain	Main contribution	Strengths	Limitations	Relevance to PPN
Lewis et al. [1], 2020	Dense retriever + seq2seq generator	Open-domain QA (Wikipedia)	Original RAG architecture	Grounds output in retrieved text; reduces hallucination	Single-pass retrieval; fixed top-k; no iterative refinement	Basis of the retrieve-then-generate step
Gao et al. [2], 2023	Naive / advanced / modular RAG taxonomy	Cross-domain RAG survey	Taxonomy of RAG variants	Comprehensive; widely adopted framework	Descriptive; no new benchmark results	Vocabulary for judging PPN retrieval
Zhao et al. [3], 2024	RAG for AI-generated content (text, code, image)	Multi-modal generative tasks	Extends RAG beyond text-only pipelines	Broadens scope of retrieval design	Multi-modal retrieval is computationally costly	Relevant to tabular marking schemes
Fan et al. [4], 2024	RAG–LLM integration survey	General LLM applications	Maps open questions in RAG–LLM fusion	Identifies research gaps	Few domain-specific (education) case studies	Motivates tests on institutional documents
Wu et al. [5], 2023	Conversable multi-agent framework with human-in-the-loop	General tasks (math, coding, QA)	Customisable agents with tool and code use	Flexible roles; tool support	Coordination overhead grows with agent count	Analogue of specialised Layer 2 agents
Hong et al. [6], 2023	SOP-encoded role-based agents	Software engineering	Standard procedures embedded in prompts	Limits error cascades via structured outputs	Rigid SOPs limit adaptability to new domains	Artefact-centred hand-off resembles Layer 2
Qian et al. [7], 2024	Chat-chain multi-agent software company	Software development lifecycle	Simulated design–code–test pipeline	Covers a full development pipeline	Software-specific; not evaluated on document tasks	Staged flow parallels PPN stages
Guo et al. [8], 2024	Survey of LLM multi-agent architectures	Cross-domain multi-agent systems	Unifies terminology and communication strategies	Identifies open frontiers	Survey only; no empirical benchmarking	Supports layered over fully connected design
Study, year	**Technique / approach**	**Dataset / domain**	**Main contribution**	**Strengths**	**Limitations**	**Relevance to PPN**
Elkins et al. [9], 2024	LLM quiz generation aligned to Bloom's levels	Programming-course quizzes	Generation validated by teacher judgement	Teacher-preferred quiz quality	Quiz generation only; no full lesson plans	Assessment Planner; Rubric Builder
Waheed et al.	Transformer-based	Course	Bloom's-level	Strong in-	Classification only;	Bloom's tagging

[10], 2021	learning-outcome classifier (BloomNet)	learning-outcome statements	classification	distribution and OOD generalisation (as reported)	no generation	within CO generation
Luo et al. [11], 2025	LLM-oriented enhanced Bloom's taxonomy	Student information-literacy assessment	Extends taxonomy for LLM-era learning	Updates taxonomy for new context	Conceptual; limited large-scale validation	Taxonomy as a design tool
Sengupta & Das [12], 2023	NLP + ML CO–PO mapping	Engineering curricula (India)	Automates a core NBA/OBE task	Directly addresses accreditation workflow	Accuracy depends on CO phrasing quality	Comparable to CO–PO–PSO Mapper
Swamy et al. [13], 2025	SVM-based CO–PO mapping framework	Engineering COs (NEP 2020 context)	Intelligent CO–PO mapping	Targets Indian OBE/NBA workflow	Small dataset; single algorithm	Same policy context as PPN
Zheng et al. [14], 2024	3-stage RAG + self-critique + refinement	Elementary maths lesson plans	Lesson-plan generator	Self-critique improves over single pass	Maths only; not higher-education courses	Idea for Lesson Planner
Varuvel Dennison et al. [15], 2026	Human-in-the-loop RAG; curator review	Government-school lesson plans (Karnataka)	Teacher–AI collaboration at scale	1,043 teachers; reduced workload and stress	Curator layer adds a human bottleneck	Educator-review pattern; Indian context
Vatankhah Barenji et al. [16], 2026	Agentic RAG grounded in rubrics and exemplars	701 student essays, higher education	Retrieval-grounded feedback	94–99% agreement with human graders (reported)	Essay grading only; not full-course documents	Rubric Builder; Assessment Planner
Yeung et al. [17], 2025	Zero-shot LLM grading, no fine-tuning	Computational and explanatory assignments	Prompt-only grading	No training data; better student-reported motivation	May be less consistent than retrieval-grounded graders	Cheaper option for some AAP parts

A. Maturity and Types of Evidence

The 17 studies are not equivalent kinds of evidence. Four are surveys [2]–[4], [8] and one is a conceptual framework [11]; the remainder are methods or systems with reported evaluations, although their evaluations differ widely in scale and rigour. The foundational RAG and multi-agent papers ([1]–[4] and [5]–[8]) are general in scope and supply the shape of PPN's pipeline, while the education-specific papers ([9]–[13] and [14]–[17]) bear on the responsibilities of individual agents. Retrieval grounding, staged agent collaboration, Bloom's and CO–PO classification, and lesson-plan and rubric generation are each reasonably well studied in isolation.

B. Classification versus Generation

The authors' earlier report grouped [9]–[13] as classification-oriented and [14]–[17] as generation-oriented. On re-reading, this grouping needs refinement. The mapping and classification studies are [10], [12] and [13]; [9] generates quizzes; and [11] is conceptual. The qualitative point stands: the closest published relatives of the CO Generator and CO–PO–PSO Mapper start from outcome text that already exists, while generation of new documents is represented mainly by the lesson-planning and assessment studies, and none of those begins from a raw syllabus and marking scheme.

C. Differences in Evaluation

Evaluation methods range from teacher judgement of generated quizzes [9] and teacher-defined criteria for lesson plans [14], to a mixed-methods deployment study [15], agreement with human graders [16], and student-reported motivation [17]. The CO–PO studies depend on small datasets or on the clarity of outcome phrasing [12], [13]. The surveys report no new empirical measure. The consequence is that PPN cannot borrow a single established evaluation standard from this literature; a paper that reports 94–99% agreement [16] is not directly comparable with one that reports teacher preference [9].

D. Relevance to Indian Engineering Education

Three studies are set in India or in the NEP 2020 policy context [12], [13], [15]. They form a small and more directly comparable body of work than the multi-agent literature, which is centred largely on other contexts. Two address CO–PO mapping, which is a classification task, and one addresses school lesson planning rather than higher-education accreditation. None addresses end-to-end generation of NBA documentation.

VI. THE PPN FRAMEWORK AND ITS RELATIONSHIP TO THE LITERATURE

This section describes PPN as documented in the authors' earlier report. It is a system proposed by the

authors; nothing in this survey demonstrates its accuracy, reliability or superiority over other approaches. Fig. 3 summarises the workflow and agent structure.

A. Workflow

A faculty member signs in to a dashboard that shows the courses assigned to them and the generation status of each. The faculty member uploads two PDF documents, a course syllabus and a marking scheme, and supplies basic course information (course name, code, semester, credits, programme and prerequisites), much of which is filled in from the faculty profile. The system then moves through seven stages: upload; processing, in which the PDFs are parsed and indexed as vector embeddings for RAG; agent generation; automated evaluation; faculty review; assembly; and output of downloadable documents.

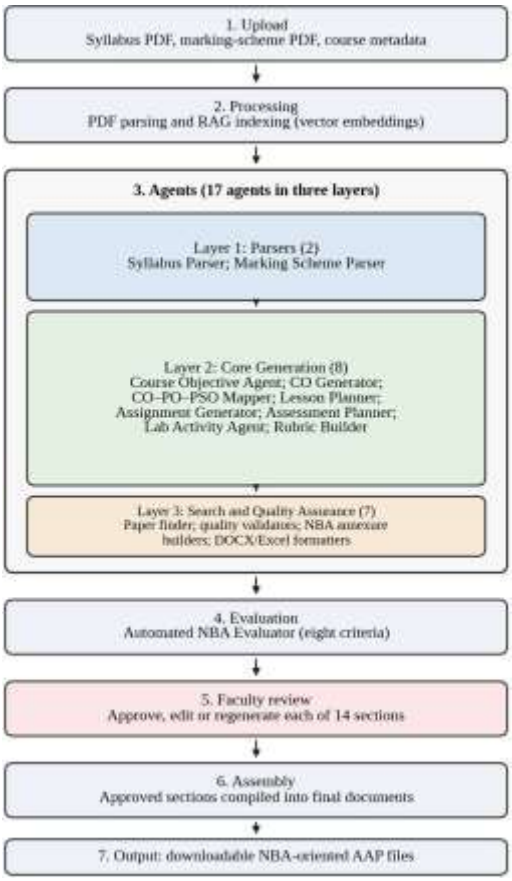


Fig. 3. PPN workflow and three-layer, 17-agent architecture, redrawn from the source report (reconstruction, not a system screenshot).

B. Three-Layer, 17-Agent Architecture

The 17 agents are arranged in three layers. Layer 1 (Parsers) contains the Syllabus Parser and the Marking Scheme Parser, which convert unstructured PDF content into structured fields. Layer 2 (Core Generation) has eight agents, each responsible for one artefact of the AAP: the Course Objective Agent, CO Generator, CO–PO–PSO Mapper, Lesson Planner, Assignment Generator, Assessment Planner, Lab Activity Agent and Rubric Builder. Layer 3 (Search and Quality Assurance) has seven agents, which include a research-paper finder, quality validators, NBA annexure builders and DOCX/Excel formatters. The text of the source report does not specify the internal algorithms, prompts or infrastructure of individual agents, and this survey does not attempt to reconstruct them. Fig. 4 reproduces the Agents stage of the deployed interface, captured mid-run, with 14 agents completed, one running, two pending and none failed. It shows how the agents are grouped and that the pipeline is implemented as an interface; it says nothing about the quality of what the agents produce. The screen also tags each agent with a model identifier, which the report does not discuss and which this survey does not evaluate.

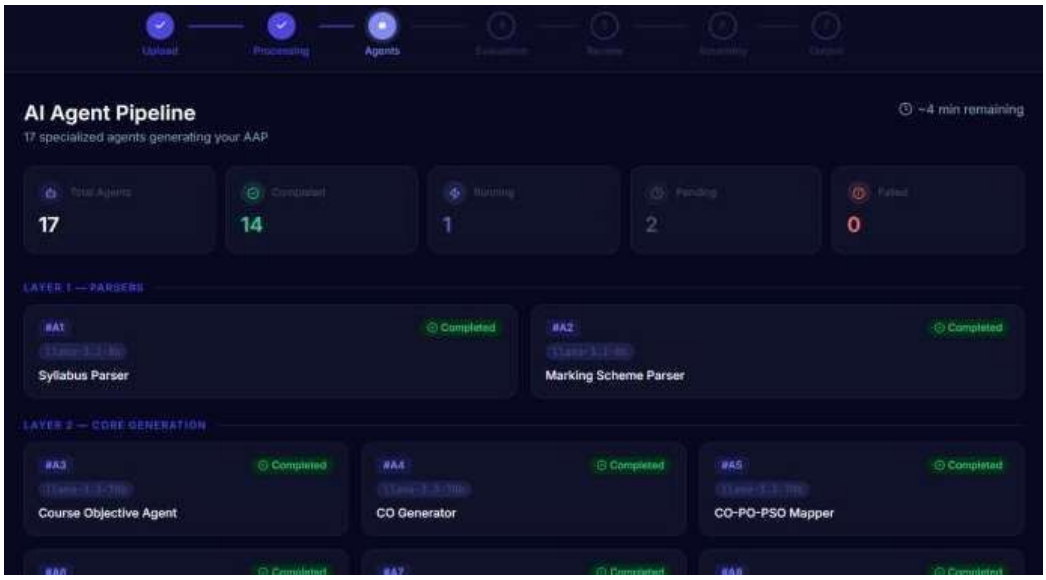


Fig. 4. Agents stage of the PPN interface during a run (original screenshot from the source report, cropped to the main panel; otherwise unaltered). It illustrates implementation and agent grouping, not output quality.

C. Evaluation, Faculty Review and Assembly

In the evaluation stage an automated NBA Evaluator tests the draft against eight accreditation criteria, scores each out of 100, and produces a composite score with sub-scores for accuracy, completeness and coherence. In a sample run described in the source, the evaluator passed seven of the eight criteria and left Bloom's-level coverage to the faculty member, which the authors take as an example of an automated evaluator deferring to human judgement rather than certifying its own output. In the review stage the faculty member can approve, edit or regenerate each of the 14 AAP sections, compare against the previous version, and give free-text feedback for the responsible agent to act on. All 14 sections must be approved before assembly begins. This pattern parallels the curator review of Shiksha Copilot [15] and the rubric-grounded instructor oversight of Vatankhah Barenji et al. [16].

Fig. 5 shows the evaluation screen from that sample run. The criteria listed include CO statement quality, CO–PO mapping completeness, Bloom's-level coverage, assessment alignment, lesson-plan completeness and lab-activity coverage. The screen reports a composite score of 94 out of 100 with sub-scores of 96% for accuracy, 92% for completeness and 94% for coherence, and a summary of seven of eight criteria passed, one needing review, CO coverage of 5/6 and Bloom's levels found of 4/6. These are values computed and displayed by the system. The source does not describe how the scores are calculated or checked against expert judgement, so they cannot be read as measures of correctness.

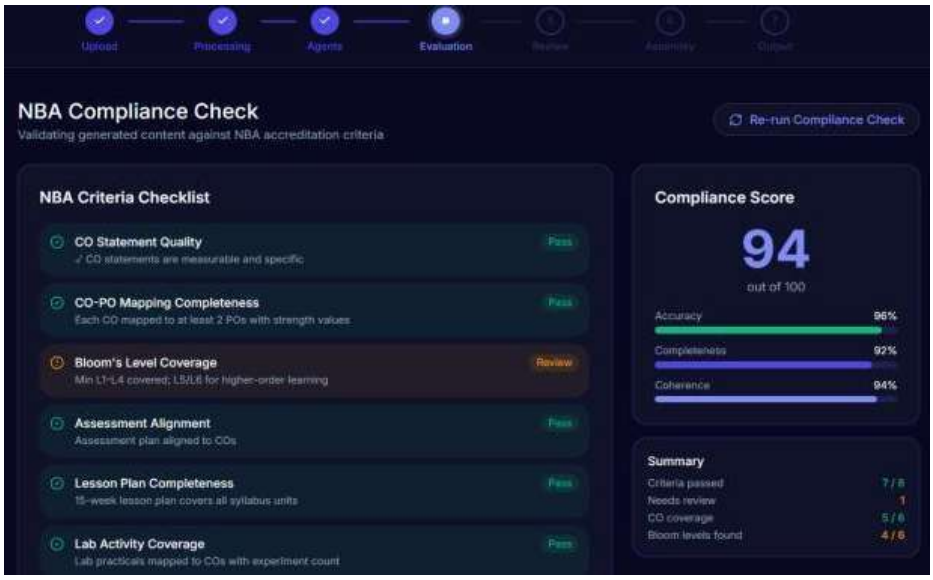


Fig. 5. NBA Compliance Check screen of the PPN interface (original screenshot from the source report, cropped to the main panel; otherwise unaltered). The scores are system-reported and have not been independently validated.

D. Relationship to the Reviewed Literature

Each stage of PPN corresponds to one of the four literature clusters. The processing stage draws on RAG [1], [2], and the two-document input makes the treatment of mixed prose and tables [3] a practical issue. The layered agent structure and artefact-level hand-offs reflect the multi-agent frameworks [6], [8]. The Layer 2 agents for outcomes and mapping have published relatives in Bloom's and CO–PO work [10], [12], [13], and those for lesson plans, assessments and rubrics in the generative cluster [14], [16]. The distinctive element is the combination: an automated compliance check placed inside the pipeline as a stage, applied to documents generated from institutional source files. The combination is the authors' design proposal and is not itself tested in the literature reviewed.

E. Reported System Figures and Their Status

TABLE II lists figures that the authors' earlier report attributes to PPN. They are either faculty-reported estimates or values shown by the PPN interface in a sample run. The number of runs, courses and faculty involved is not documented, no baseline was measured, and no independent verification was carried out. They are therefore reported here for context only and should not be read as experimental results.

Treating these figures as evidence would require a documented evaluation protocol: a stated set of courses, a manually prepared baseline AAP for each, a defined way of measuring faculty time and review effort, and an independent check of the compliance scores. None of these is available, so the figures are retained here as claims of the authors' system and are not graphed or compared.

TABLE II. FIGURES REPORTED FOR PPN (UNVERIFIED)

Item	Value	Basis
Generation time per AAP	≈ 8 minutes	Reported by authors
Inference cost per AAP	≈ ₹6.40	Reported by authors
Initial manual preparation	≈ 40 hours	Faculty-reported estimate
Faculty review time	≈ 8 hours	Reported estimate
Quality score	94/100	PPN interface, sample run
CO coverage	98%	PPN interface, sample run
Sections approved	14 of 14	PPN interface, sample run
Faculty time saved	32 hours	PPN interface estimate
Evaluator criteria passed	7 of 8	PPN interface, sample run

Fig. 6 shows the completed-run screen from which several of them are taken. Its "NBA COMPLIANT" label and its 98% CO coverage, 94/100 quality score and 32 hours saved are outputs displayed by the system. The source does not state whether this run is the same as the one in Fig. 5, which reports CO coverage as 5/6, so the two screens cannot be reconciled from the material available.



Fig. 6. Completed-run screen of the PPN interface (original screenshot from the source report, cropped to the main panel; otherwise unaltered). Displayed figures are system-reported, not independently validated results.

VII. RESEARCH GAPS AND OPEN CHALLENGES

A. Technical Integration and Compliance Validation

AutoGen, MetaGPT and ChatDev were designed and tested for tasks other than rule-bound document generation [5]–[7]. Among the multi-agent studies reviewed, none places an automated compliance checker inside the pipeline as a stage; compliance checking is left to a human after the fact. On the retrieval side, the general-purpose RAG literature [1]–[4] does not examine retrieval from two structurally mismatched institutional PDFs, one mostly prose and one mostly tables. This matters because a retrieval error in either source flows into every artefact generated from it. The claim is limited to the 17 studies reviewed and is not a statement that no such work exists.

B. Dataset Limitations and Cross-Institution Generalisation

The Bloom's-taxonomy and CO–PO studies reviewed here tend to rely on small samples drawn from a single institution [9]–[13], and neither CO–PO mapping paper [12], [13] nor BloomNet [10] uses a common, multi-institution collection of course outcomes. Without a shared benchmark, any component such as a CO Generator or CO–PO–PSO Mapper can only be compared with its own authors' data. There is also a real risk that a model tuned to the way one college phrases its outcomes will perform differently on another's.

C. Performance, Scalability and Error Propagation

Guo et al. [8] note that fully connected agent communication does not scale well as the number of agents grows, which is directly relevant to a 17-agent system unless communication is deliberately layered. Hong et al. [6] describe how an early error can be silently carried through later stages; in PPN an error in the Syllabus Parser or CO Generator could pass through the CO–PO–PSO Mapper into the annexures built on it until it is caught. Finally, none of the reviewed studies reports end-to-end cost or latency figures comparable to those reported for PPN (about ₹6.40 per document and about eight minutes), so whether those figures represent an advantage cannot be judged from the literature.

D. Deployment, Human Oversight and Evaluation

The two large deployment-oriented studies both keep a human in the loop, a curator for Shiksha Copilot [15] and an instructor sign-off in the rubric-grounded system [16]. Neither, as summarised in the authors' report, states how much generated text survived review unchanged. Review time is likely to grow with document length and agent count, yet it is not recorded in the reviewed literature, even though the roughly eight hours of faculty review claimed for PPN determines whether such a system saves time or only relocates the work. The figures that PPN's own output screen reports (14 of 14 sections approved, 98% CO coverage, 94/100 quality and 32 hours saved) are self-reported and are exactly the kind of number that needs independent verification. This is a measurement gap more than a technical one: the field has shown that human-in-the-loop systems can work, but has not

published enough to show how much work remains with the human.

VIII. DISCUSSION AND FUTURE RESEARCH DIRECTIONS

A. Synthesis

Taken together, the reviewed studies suggest that each building block of a pipeline like PPN has been examined, but mostly in separate settings and with different measures. What the literature establishes is that retrieval grounding is a mature pattern, that structured multi-agent hand-offs are a recognised way to limit error cascades, that outcome statements can be classified and mapped by NLP and machine-learning methods, and that LLMs can produce lesson plans and support grading under suitable control. What it does not establish is that these pieces combine reliably to produce accreditation-grade documents from raw syllabi, how such a combination performs across institutions, or how much human effort it saves.

B. Future Research Directions

The following directions follow from the gaps above. They are recommendations of the authors and are not results established by the reviewed studies.

1) *A shared, anonymised, multi-institution dataset* of course outcomes, syllabi and marking schemes, so that CO–PO mapping and Bloom's classification components can be compared on common data.

2) *Standardised benchmarks* for CO–PO mapping and Bloom's tagging, with agreed metrics and expert-labelled references.

3) *Layered versus fully connected agent communication*, tested on pipelines of comparable size to PPN, to examine the scaling argument made in general terms in the multi-agent literature [8].

4) *Measurement of generation accuracy, latency, inference cost and review effort*, including the proportion of generated text that reviewers change.

5) *Comparison with manually prepared AAPs*, using faculty time-to-approval, NBA-criterion pass rates and cost per document against a manually authored baseline.

6) *Extension to other accreditation frameworks* such as ABET or other Washington Accord signatories, to test whether an NBA-style evaluator transfers beyond the Indian context.

C. Human Oversight, Privacy and Traceability

Generated accreditation documents carry institutional and professional responsibility. Human approval of each section, traceability from generated text to the source passages that supported it, and careful handling of institutional documents are therefore design requirements and not optional extras. Datasets assembled for benchmarking should be anonymised, and an automated evaluator should be positioned as an aid to faculty judgement and not as a certifying authority, a stance consistent with the behaviour described for PPN's evaluator in Section VI.

IX. LIMITATIONS OF THE REVIEW

Several limitations apply. First, the review covers only 17 studies, so absence of a topic from this set does not show that no related work exists. Second, the search relied on a defined set of databases and on the authors' own screening, which introduces possible database and selection bias; search dates, record counts and screening stages were not documented, and no second reviewer or agreement measure was used. Third, the set includes preprints [2], [3], [5], [6], [11], [16], [17] alongside peer-reviewed publications, and preprint findings may change after review. Fourth, the studies use different tasks, datasets and metrics, so the comparison in Section V is qualitative, and the reviewed systems cannot be compared directly. Fifth, strengths and limitations of the individual studies were taken from their published descriptions and the authors' earlier report and were not reproduced. Finally, PPN itself has not been independently validated in this survey, and the figures in TABLE II remain unverified.

X. CONCLUSION

This survey examined 17 studies in four areas: retrieval-augmented generation, LLM-based multi-agent orchestration, Bloom's-taxonomy and CO–PO automation, and AI-assisted lesson planning and assessment. The components on which a system for generating NBA-oriented Academic Administration Plans would depend, namely retrieval grounding, staged agent collaboration, outcome classification and mapping, and lesson-plan and rubric generation, have each been studied, but under different conditions and with different evaluation methods. Within the reviewed set, outcome-mapping work concentrates on text that already exists, and none of the studies combines document-grounded generation, agent orchestration and automated compliance checking for Indian accreditation documents.

Four gap areas follow: integration and compliance validation, small single-institution datasets, error propagation and scalability, and the unmeasured cost of human review. Addressing them calls for shared benchmarks, tests of layered agent communication, and comparison of generated plans with manually prepared ones. Integrating RAG, multi-agent orchestration and compliance validation may reduce the documentation burden on faculty, but whether it does so reliably, and at what review cost, has still to be shown by independent evaluation.

ACKNOWLEDGMENT

The authors thank the Department of Information Technology, Vidyalankar Institute of Technology, for the resources used in preparing this survey, which was

carried out under the Honors Minor Specialization Programme.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459–9474.

[2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023 (preprint).

[3] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-augmented generation for AI-generated content: A survey," arXiv:2402.19473, 2024 (preprint).

[4] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A survey on RAG meeting LLMs: Towards retrieval-augmented large language models," in Proc. 30th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD), 2024, pp. 6491–6501.

[5] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework," arXiv:2308.08155, 2023 (preprint).

[6] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, and C. Wu, "MetaGPT: Meta programming for a multi-agent collaborative framework," arXiv:2308.00352, 2023 (preprint).

[7] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong, J. Xu, D. Li, Z. Liu, and M. Sun, "ChatDev: Communicative agents for software development," in Proc. 62nd Annu. Meeting Assoc. Computational Linguistics (ACL), vol. 1, 2024, pp. 15174–15186.

[8] T. Guo et al., "Large language model based multi-agents: A survey of progress and challenges," in Proc. 33rd Int. Joint Conf. Artificial Intelligence (IJCAI), 2024, pp. 8048–8057.

[9] S. Elkins, E. Kochmar, J. C. K. Cheung, and I. Serban, "How teachers can use large language models and Bloom's taxonomy to create educational quizzes," in Proc. AAAI Conf. Artificial Intelligence, vol. 38, no. 21, 2024, pp. 23378–23386.

[10] A. Waheed, M. Goyal, N. Mittal, D. Gupta, A. Khanna, and M. Sharma, "BloomNet: A robust transformer based model for Bloom's learning outcome classification," in Proc. 4th Int. Conf. Natural Language and Speech Processing (ICNLSP), 2021, pp. 209–218.

[11] Y. Luo, T. Liu, P. C.-I. Pang, D. McKay, Z. Chen, G. Buchanan, and S. Chang, "Enhanced Bloom's educational taxonomy for fostering information literacy in the era of large language models," arXiv:2503.19434, 2025 (preprint).

[12] S. Sengupta and A. K. Das, "Automated mapping of course outcomes to program outcomes using natural language processing and machine learning," in Proc. 2023 IEEE 3rd Applied Signal Processing Conf. (ASPCON), 2023, pp. 44–48, doi: 10.1109/ASPCON59071.2023.10396272.

[13] K. V. Swamy, S. Thirumala Devi, and S. Aouthu, "Mapping engineering course outcomes to program outcomes using an intelligent framework," J. Engineering Education Transformations, vol. 38, Special Issue 2, Jan. 2025.

[14] Y. Zheng, X. Li, Y. Huang, Q. Liang, T. Guo, M. Hou, B. Gao, M. Tian, Z. Liu, and W. Luo, "Automatic lesson plan generation via large language models with self-critique prompting," in Proc. Int. Conf. Artificial Intelligence in Education (AIED), Commun. Comput. Inf. Sci., vol. 2150. Springer, 2024, pp. 163–178.

[15] D. Varuvel Dennison, B. Ahtisham, K. Chourasia, N. Arora, R. Singh, R. F. Kizilcec, A. Nambi, T. Ganu, and A. Vashistha, "Shiksha Copilot: Teacher–AI collaboration for curating and customizing lesson plans in low-resource schools," Proc. ACM Hum.-Comput. Interact., vol. 10, no. 2, Art. CSCW038, Apr. 2026, doi: 10.1145/3788074.

[16] R. Vatankhah Barenji, N. Salimi, and S. Khoshgoftar, "An LLM-powered assessment retrieval-augmented generation (RAG) for higher education," arXiv:2601.06141, 2026 (preprint).

[17] C. Yeung, J. Yu, K. C. Cheung, T. W. Wong, C. M. Chan, K. C. Wong, and K. Fujii, "A zero-shot LLM framework for automatic assignment grading in higher education," arXiv:2501.14305, 2025 (preprint).