Percorso dottorale sviluppato con il sostegno finanziario di NextGenerationEU:
Missione X, Componente 0, Investimento 0.0, CUP B63D22000xxxxxxx
Borsa MUR ex DM 352/2022

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

Dottorato in Ingegneria e Scienze dell'Informazione

38° Ciclo

Coordinatore/Coordinatrice: Prof. Mauro Barni

From Language to Learning: An LLM-Driven Framework for Multilingual Educational

Settore scientifico disciplinare: Intelligenza Artificiale

Candidato
Kamyar Zeinalipour
Università degli Studi di Siena

Kamyar Zeinalipour
Firma digitale del/della candidato/a

Supervisore
Prof. Marco Gori
Università degli Studi di Siena

Anno accademico di conseguimento del titolo di Dottore di ricerca
2024/2025

Università degli Studi di Siena

Dottorato in Ingegneria e Scienze dell'Informazione

38° Ciclo

Data dell'esame finale

_____

Commissione giudicatrice

_____

_____

_____

_____

_____

Supplenti

_____

_____

UNIVERSITÀ DEGLI STUDI DI SIENA

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

From Language to Learning: An LLM-Driven Framework for Multilingual Educational Content

Kamyar Zeinalipour

PhD in Information Engineering and Science

Supervisor

Prof. Marco Gori

Examination Committee

Prof.

Prof.

Prof.

Thesis reviewers

Prof. Hava Siegelmann

Prof. Hanan Salam

Siena, 10/02/2026

Abstract

This thesis introduces ***"From Language to Learning,"*** *a unified, LLM-driven framework designed to address the critical challenge of creating scalable, personalized educational content across diverse linguistic and pedagogical contexts. It proposes a systematic methodology for adapting general-purpose large language models (LLMs) into specialized, open-source educational aids, particularly for under-resourced languages.*

The cornerstone of this framework is a task-specific adaptation of distillation for creating high-quality, task-specific instructional datasets—the ***`Instruct` series****. First, a state-of-the-art LLM is prompted to generate structured educational materials from raw, unstructured text. This synthetic data is then used to fine-tune smaller, open-source models, efficiently transferring sophisticated capabilities for specific learning objectives.*

The framework's pedagogical versatility is demonstrated through a progression of applications. To foster engagement and vocabulary reinforcement, we developed systems for generating educational crossword puzzles in ***English, Italian, Arabic, and Turkish****. Evolving from gamification to formal assessment, the framework was then extended to create models for generating multiple-choice and short-answer questions in* ***Persian and Turkish****. Finally, to provide direct pedagogical intervention, the framework was applied to the nuanced task of delivering automated syntax feedback on student essays in* ***English****.*

The efficacy of this approach is validated through systematic human and automatic evaluations for each language and task. The results consistently and significantly show that fine-tuning smaller LLMs on our generated ***`Instruct`*** *datasets makes them highly effective and reliable specialized tools. Ultimately, this thesis contributes a scalable and validated blueprint that bridges the gap between general-purpose language models and the specific needs of multilingual education, democratizing the creation of next-generation, AI-powered learning technologies. Across this thesis, we produced* ***11*** *publications, released* ***11*** *datasets (including* ***7*** *in the Instruct series), and trained/evaluated* ***29*** *fine-tuned generator models across* ***5*** *languages, enabling reproducible, multilingual educational NLP at scale. All code, datasets, and models developed in this thesis are publicly available to the research community at:* `https://github.com/KamyarZeinalipour/From-Language-to-Learning`

Acknowledgements

I would like to express my deepest and most sincere gratitude to my supervisor, **Prof. Marco Gori**. His visionary guidance, profound knowledge, and unwavering support have been the cornerstone of this research. Beyond his technical expertise, I am grateful for the academic freedom he granted me, allowing me to explore the intersections of language and learning with curiosity. His mentorship has not only shaped this thesis but has also fundamentally influenced my approach to scientific research.

I am also immensely grateful to **Prof. Marco Maggini**, whose expertise and collaboration have been invaluable throughout my doctoral journey. I deeply appreciate the time he dedicated to discussing my work; his constructive feedback and insightful suggestions significantly improved the quality of this research.

My sincere thanks go to **Prof. Monica Bianchini** and **Prof. Stefano Melacci** for their continuous support and for creating a stimulating academic environment at the University of Siena. Their encouragement, advice, and the scientific discussions we shared have been a constant source of motivation and learning during my time in the department.

I would like to extend a special thank you to **Prof. Hava Siegelmann** and **Prof. Nina Tahmasebi**. I am grateful for their valuable perspectives, scientific inputs, and for the opportunity to learn from their extensive expertise in the field. Their feedback has been instrumental in refining the contributions of this work.

Finally, I owe a debt of gratitude that can never be fully repaid to my family, whose love and sacrifice made this journey possible. To my father, **Hossien**, and my mother, **Azita**: thank you for your unconditional love, your endless patience, and for believing in me even when I doubted myself. You have always been my safe haven. To my brother, **Amir**, thank you for your encouragement and for always being there to support me.

To my friends, near and far, thank you for the moments of joy, the listening ears, and for keeping me grounded. This accomplishment is as much yours as it is mine.

List of Acronyms

AES automated essay scoring

AI artificial intelligence

ASAP Automated Student Assessment Prize

AWA automated writing assessment

BART Bidirectional and Auto-Regressive Transformers

BERT Bidirectional Encoder Representations from Transformers

BERTScore BERT-based semantic similarity metric

GEC grammatical error correction

GPT Generative Pretrained Transformer

GPT-4 Generative Pretrained Transformer 4

GPT-4o GPT-4 omni

JSON JavaScript Object Notation

LLaMA Large Language Model Meta AI

LLM large language model

LoRA low-rank adaptation

mBERT Multilingual BERT

MCQ multiple-choice question

NER named entity recognition

NLP natural language processing

PEFT parameter-efficient fine-tuning

QA question answering

QG question generation

RACE ReAding Comprehension from Examinations

ROUGE Recall-Oriented Understudy for Gisting Evaluation

RTL right-to-left

SAQ short-answer question

SQuAD Stanford Question Answering Dataset

T5 Text-to-Text Transfer Transformer

XLM-R Cross-lingual Language Model—RoBERTa

Contents

List of Figures

List of Tables

Part I

Motivation, Background, and Related Work

Chapter 1

Introduction

1.1 The Grand Challenge: Scalable and Personalized Multilingual Education

The promise of modern educational technology—to deliver truly personalized learning to anyone, anywhere—hinges on a persistent bottleneck: the scalable creation of high-quality content. Platforms have matured as delivery mechanisms, but they are only vessels. The fuel for learning—pedagogically sound, culturally relevant, and engaging material—remains scarce, and current authoring workflows cannot keep pace with technological potential.

The constraint is not volume alone but *specificity*. Effective education must serve the *long tail* of learning: the vast spectrum of subjects, proficiency bands, and pedagogical contexts that fall outside standardized curricula. For every mainstream topic, countless niche domains, vocational skills, and foundational concepts are underserved. Handcrafted, bespoke materials—lesson explanations, practice items, formative feedback, and assessments—are costly to produce and difficult to maintain, rendering large-scale personalization economically and logistically infeasible.

Quality compounds the challenge. Useful materials must (i) align with explicit learning objectives; (ii) be calibrated for difficulty and progression; (iii) respect cultural and linguistic norms; and (iv) appear in formats that actively support learning (e.g., guided practice, formative assessment, and actionable feedback). Meeting these criteria via manual authoring does not scale.

This bottleneck widens into a chasm for non-English languages, creating a digital language divide that obstructs educational equity. Scarcity in languages such as Italian, Arabic, Turkish, and Persian is systemic, driven by:

- **Data scarcity:** Limited, curated educational datasets impede the training and evaluation of effective models.
- **Economic misalignment:** Commercial incentives favor high-resource languages, leaving smaller markets underserved.
- **Linguistic complexity:** Typological properties—agglutinative morphology (e.g., Turkish) and right-to-left abjad scripts (Arabic, Persian)—introduce tokenization, orthographic, and layout challenges often ignored by one-size-fits-all pipelines.

The result is not merely a technical shortfall but a barrier to the democratization of knowledge: the benefits of AI-enabled learning remain concentrated among speakers of a

few dominant languages.
Large Language Models (LLMs) lower the barrier to content generation, but naïve prompting of general-purpose models is insufficient. Unconstrained outputs can hallucinate facts, drift from source texts, misalign with objectives, and yield uneven difficulty or vague feedback. Exclusive reliance on proprietary "teacher" models further limits accessibility, auditability, and sustainable deployment in real educational settings.

This thesis contends that closing the gap requires a *systematic pathway* from general LLM competence to reliable, language-aware, pedagogy-aligned tools. We therefore pursue a framework guided by the following design objectives:

- **Scale:** Transform abundant raw text into structured educational artifacts for the long tail of topics.
- **Pedagogical fidelity:** Encode task structure (e.g., clue formats, MCQ distractor quality, syntactic feedback rationales) and calibrate difficulty.
- **Multilingual equity:** Operate across typologically diverse languages without bespoke, hand-engineered pipelines per language.
- **Efficiency and openness:** Distill capabilities into smaller, open-source models suitable for local, cost-effective deployment.
- **Reliability and evaluation:** Promote grounding in source material, mitigate hallucination risks, and validate with both human and automatic measures

It is important to explicitly state that within this framework, 'grounding' refers to strict adherence to the provided source text and not a semantic understanding of the world. Framed this way, the grand challenge is to convert unstructured text and general LLM ability into *trustworthy* educational artifacts—crossword clues, assessment questions, and targeted writing feedback—at scale and across languages. The remainder of this thesis develops and validates a unified, distillation-based approach that operationalizes these objectives.
Across this thesis, we produced **11** publications—crosswords/puzzles [3, 4, 5, 6, 7, 8, 9, 10], assessment [11, 12], and feedback [13]—released **11** datasets (including **7** in the *Instruct* series), and trained/evaluated **29** fine-tuned generator models across **5** languages—quantitatively grounding the scope and impact of the framework introduced here.

1.1.1 Educational Crossword Generation

Crossword generation has long been explored in NLP, spanning both solving and clue/puzzle construction. For this thesis, we focus on *educational* crosswords—puzzles whose clues are factual, source-grounded, and pedagogically oriented—rather than cryptic, entertainment-style clueing. This distinction matters: educational clues must privilege accuracy, clarity, and alignment with a given text or curriculum, not wordplay.
Early generation systems leveraged dictionaries, web-mined definitions, and rule-based ranking to assemble clue–answer pairs [14, 15, 16, 17]. While influential, these pipelines required substantial manual tuning, transferred poorly across domains and languages,

and were not designed to produce *text-grounded* educational clues. With the advent of LLMs, clue generation became more flexible, yet a core gap persisted: the absence of large, high-quality datasets of educational clue–answer pairs derived from source texts and spanning multiple languages.

Our work closes this gap with a unified, LLM-driven approach. First, we introduce a two-stage distillation pipeline that transforms raw passages into structured educational data, instantiating the mapping `(Text, Keyword)` $\rightarrow$ `(Clue)`. We then instruction-tune open models on this synthetic, text-grounded data, and integrate verification (classifier and LLM-as-judge filters) to reduce ambiguity and hallucinations. This methodology produced the `Instruct` series of datasets and models for English, Italian, Arabic, and Turkish, enabling scalable generation of pedagogically sound clues directly from curriculum-aligned texts [5, 4, 8, 9].

Beyond scalability, the framework addresses language-specific challenges that matter in education: right-to-left scripts and diacritic normalization for Arabic; agglutinative morphology for Turkish; and style-controlled clueing for Italian. Across languages, human and automatic evaluations show that instruction-tuned students substantially outperform general-purpose baselines on clarity, correctness, and textual grounding. Chapters 4–7 detail datasets, models, and evaluations per language.

1.1.2 Question Generation for Education

Question Generation (QG) has progressed from rule and template based systems to neural sequence-to-sequence models and, most recently, to instruction-following LLMs trained on large comprehension corpora. While this evolution has enabled impressive fluency—fueled by datasets such as SQuAD and RACE [18, 19]—*educational* QG introduces requirements that extend beyond generic question writing. In assessment settings, items must be syllabus-aligned, unambiguous, and difficulty-calibrated; for MCQs, distractors must be plausible yet discriminative rather than trivially wrong. These constraints make educational QG a distinct problem space rather than a straightforward transfer of general-domain methods.

A useful way to situate the literature is to distinguish (i) **answer-aware** vs. **answer-agnostic** QG, (ii) **format** (multiple-choice vs. short-answer), and (iii) **distractor construction** (generation vs. retrieval/selection). Early answer-aware systems relied on handcrafted templates; neural approaches introduced attention and copy mechanisms to better tie questions to salient spans; transformer-based models further improved coherence and coverage. Yet three challenges remain central for education: (1) **distractor quality**, where options should reflect common misconceptions while avoiding ambiguity; (2) **depth of comprehension**, ensuring items assess inference and integration rather than surface recall; and (3) **difficulty control and curriculum alignment**, so that items target intended proficiency bands and learning objectives.

These challenges are amplified in low-resource and morphologically rich languages. Persian's orthography, limited pedagogical corpora, and scarcity of high-quality MCQ resources—and Turkish's agglutinative morphology and rich derivational processes— complicate key selection, distractor formation, and unambiguous phrasing. As a result,

English-centric methods do not transfer cleanly, and validated educational QG resources/-models for these languages remain limited.
This thesis addresses that gap by extending our LLM-driven distillation framework from gamified learning to formal assessment. Concretely, we introduce *PersianMCQ-Instruct*, a large, instruction-style corpus produced by a controlled multi-stage prompting pipeline (content transformation, MCQ drafting, distractor refinement), and use it to fine-tune open models for Persian MCQ generation; human and automatic evaluations indicate substantial gains after fine-tuning [11]. We further contribute *Turkish-Quiz-Instruct*, pairing Turkish educational texts with both multiple-choice and short-answer items, and demonstrate similar improvements when distilling from a teacher LLM into smaller student models [12]. Together, these resources and models operationalize the core claim of this thesis: distilling task-specific pedagogy into compact LLMs yields reliable, education-ready generators for under-resourced languages.
In summary, while the broader QG literature provides the modeling substrate, educational QG in Persian and Turkish requires dedicated data creation, distractor design, and evaluation protocols. The contributions in Part II meet these needs and empirically validate the proposed framework in two typologically distinct, low-resource settings [11, 12].

1.1.3 Writing Assessment and Feedback

Automated Writing Assessment (AWA) has been an active research area for decades, evolving from systems built on hand-crafted features to modern neural approaches (often framed as Grammatical Error Correction, GEC). While recent large language models can provide high-level comments on student writing, they frequently over-correct, hallucinate rules, or offer advice that is too general to be instructionally useful. For classroom use, where learners need concrete and trustworthy guidance, such behavior undermines reliability.
This thesis adopts a focused scope: *syntax-level feedback* that identifies a problem, proposes a *minimal* correction, and explains the change in clear pedagogical terms. The objective is not holistic scoring or stylistic critique, but dependable, fine-grained support that helps learners understand *why* a sentence should be revised. In this setting, precision and calibrated restraint—preferring *no change* when uncertain—are essential for building instructor and learner trust.
Within the proposed framework, we instantiate a two-stage process tailored to this task. In Stage 1, a high-capability teacher model is prompted to transform learner-like inputs into structured supervision of the form

$$\texttt{(Essay Sentence)} \rightarrow \texttt{(Corrected Sentence, Feedback)},$$

emphasizing minimal edits and concise, rule-aware explanations. In Stage 2, this synthetic supervision is used to instruction-tune a smaller, open-source student model that performs the same mapping with greater efficiency and consistency. Applied to English student essays, this targeted distillation yields a specialized model that provides accurate, actionable syntax feedback with markedly reduced over-correction and clearer justifications [13]. This positions the feedback component as a natural complement to the framework's other applications, extending it from content generation to direct pedagogical intervention.

1.2 Research Questions and Objectives

The central intellectual pursuit of this thesis is to overcome the limitations of prior work by establishing a unified, scalable, and multilingual framework for educational content generation. This ambition is encapsulated in a main research question, which is subsequently decomposed into three targeted sub-questions that align directly with the three parts of this thesis.

Main Research Question (MRQ):

> *How can general-purpose Large Language Models be systematically and scalably adapted to generate a diverse range of high-quality, pedagogically-varied educational content, particularly for under-resourced languages?*

To comprehensively answer this overarching question, the following specific research questions are posed:

- **RQ1 (Crosswords):** To what extent can an LLM-driven distillation framework generate pedagogically valid crossword puzzles across multiple languages (English, Italian, Arabic, Turkish), and how does the performance of specialized, fine-tuned models compare to their general-purpose baselines?
- **RQ2 (Question Generation):** Can this framework be effectively extended from gamified learning to formal assessment by generating multiple-choice and short-answer questions in morphologically rich, low-resource languages like Persian and Turkish?
- **RQ3 (Writing Assistance):** How effective is the proposed framework when applied to the nuanced, feedback-oriented task of identifying syntactic errors and providing constructive, automated feedback for English student essays?

1.3 The "From Language to Learning" Framework

In response to the challenges of scalability, multilingual support, and pedagogical specificity, this thesis introduces the **"From Language to Learning"** framework. To ensure reproducibility and facilitate future research, the official repository containing the system prompts, filtering scripts, sample inputs/outputs, and links to all released datasets and models is available at: `https://github.com/KamyarZeinalipour/From-Language-to-Learning`
This framework presents a systematic methodology for transforming powerful, general-purpose LLMs into highly specialized, efficient, and open-source educational tools. The core idea is to bridge the capability gap between colossal, often proprietary "teacher" models (e.g., GPT-4) and smaller, more accessible "student" models (e.g., Llama, Mistral) that can be deployed and customized by the broader educational and research community.
At the heart of this framework is a task-specific adaptation of distillation designed to create high-quality, task-specific instructional datasets, which we term the `Instruct` series.

- **Stage 1: Synthetic Data Generation.** In the first stage, a state-of-the-art "teacher" LLM is prompted with unstructured source text (such as a Wikipedia article) along with a detailed set of instructions defining a specific educational task. The

model's role is to act as an expert content creator, transforming the raw text into high-quality, structured educational material. For example, for crossword generation, the transformation is `(Text) -> (Clue, Answer, Category)`. For question generation, it is `(Text) -> (Question, Correct Answer, Distractors)`. For syntax feedback, the task is `(Essay Sentence) -> (Corrected Sentence, Feedback)`. This step leverages the advanced reasoning and generation capabilities of the teacher model to create a large-scale, structured dataset that perfectly aligns with the target task.

- **Stage 2: Instruction Fine-Tuning.** In the second stage, the synthetic dataset generated in Stage 1 is used as the training corpus to fine-tune a smaller, open-source "student" model. This process of instruction fine-tuning efficiently "distills" the sophisticated, task-specific capabilities of the large teacher model into its smaller counterpart. The resulting model is a specialized expert, optimized to perform its designated educational function with high accuracy and efficiency.

This framework offers several compelling advantages. It is **scalable**, as it can be applied to any domain or language where unstructured text is available. It is **cost-effective**, as fine-tuning is orders of magnitude cheaper than pre-training a model from scratch. It promotes the **democratization** of AI in education by producing powerful, open-source models that can be freely used, audited, and improved by the community. Finally, it yields **specialized** expert models that consistently outperform general-purpose chatbots on their specific tasks, providing reliable and high-quality educational content.

1.4 Summary of Key Results and Validation

The efficacy and versatility of the "From Language to Learning" framework are empirically validated across the three distinct parts of this thesis. Each part demonstrates that the two-stage distillation process yields specialized models that significantly outperform their general-purpose baselines.

- **Part I (Crosswords):** The framework was first applied to generate educational crossword puzzles in four languages. The results were consistently positive, confirming the framework's robustness. For instance, in the Arabic crossword generation task, the proportion of top-rated ("A") clues produced by the fine-tuned Llama3-8B model surged from 36% to an impressive 79% [8]. Similar significant gains in performance and quality were observed for English [5], Italian [4], and Turkish [9], validating the framework's cross-lingual capabilities.
- **Part II (Question Generation):** The framework's adaptability was tested by extending it from gamified learning to formal assessment. When tasked with generating MCQs and SAQs for Persian and Turkish, the specialized models again showed dramatic improvement. In the Persian MCQ generation task, the overall human evaluation score for the fine-tuned Llama3.1-8b model rose from 3.31 to 4.51 (out of 5), demonstrating the framework's capacity to handle the complexities of a new, assessment-focused task in low-resource languages [11].

- **Part III (Writing Assistance):** Finally, the framework was applied to the highly nuanced task of generating automated syntax feedback. The results underscored its effectiveness even for feedback-oriented applications. For the Llama2 13B model, the percentage of "good" feedback (rated "B" by human evaluators) jumped from just 11% in the base model to 60% after fine-tuning [13]. This successful application to a complex, corrective task validates the framework's broad applicability across a spectrum of pedagogical needs.

1.5 Contributions and Thesis Outline

This thesis presents a unified and validated solution to the pressing challenge of scalable, multilingual educational content generation. By designing, implementing, and rigorously evaluating the "From Language to Learning" framework, this work makes several key contributions to the fields of artificial intelligence, computational linguistics, and educational technology.
The primary contributions of this research are:

1. **A Methodological Contribution:** The design, implementation, and validation of a novel, scalable two-stage distillation framework for adapting LLMs to multilingual educational content generation.

2. **An Empirical Contribution:** The first large-scale, multilingual demonstration of a unified framework across three distinct pedagogical tasks: gamified learning (crosswords), formal assessment (question generation), and writing support (syntax feedback).

3. **A Resource Contribution:** The creation and public release of seven large-scale instructional datasets and multiple specialized, open-source models for English, Italian, Arabic, Turkish, and Persian, directly addressing the critical resource scarcity in these areas.

4. **A Practical Contribution:** A validated blueprint that democratizes the development of AI-powered educational tools, making the creation of next-generation learning technologies more accessible for educators, researchers, and developers worldwide.

The remainder of this thesis is structured to elaborate on these contributions. **Chapter 2** provides a detailed survey of the three related fields—crossword generation, question generation, and automated writing feedback—that form the academic context for this work. The thesis is then divided into three main parts. **Part I** focuses on educational crossword generation. **Chapter 3** introduces the core Clue-Instruct method for English. **Chapters 4, 5, and 6** then demonstrate its successful application to Italian, Arabic, and Turkish, respectively, reporting on the unique challenges and evaluation results for each language. **Part II** pivots from gamification to assessment. **Chapter 7** details the generation of multiple-choice questions for Persian, while **Chapter 8** extends this to both multiple-choice and short-answer questions for Turkish. **Part III** addresses the

task of writing assistance, with **Chapter 9** presenting the application of the framework to generate automated syntax feedback for student essays. Finally, **Chapter 10** candidly discusses the limitations of this work. **Chapter 11** outlines promising directions for future research, and **Chapter 12** concludes by summarizing the key findings and reiterating the significance of the "From Language to Learning" framework.

Chapter 2

Related Work

2.1 Educational Content Generation Frameworks

2.1.1 Historical Evolution

Early computer-based education relied on rule-driven Intelligent Tutoring Systems (ITS) that encoded domain knowledge and pedagogical strategies as expert rules or scripts, often with high authoring cost and limited flexibility [20, 21]. Meta-analyses later showed that well-engineered ITS could approach the effectiveness of human tutoring in controlled settings, despite relying largely on pre-authored or templated content [22]. In parallel, automatic question generation (AQG) began with linguistic and template-based transformations of source sentences into factual questions, paired with statistical ranking to ensure answerability and grammaticality [23].

The shift to neural sequence-to-sequence models transformed content generation. Du et al. demonstrated end-to-end neural question generation from reading passages, surpassing rule-based systems in fluency and coverage [24]. Subsequent transformer-based approaches leveraged large-scale pretraining (e.g., BERT/T5/BART), enabling stronger context modeling and effective fine-tuning on relatively small educational datasets [25]. Instruction-tuned large language models (LLMs) further broadened capabilities: models such as InstructGPT and GPT-4 can follow high-level pedagogical prompts (e.g., "generate a quiz for grade 5 on photosynthesis"), substantially reducing task-specific engineering [26, 27]. Comprehensive surveys document this progression from rules to neural and instruction-following LLMs across learning tasks (quiz generation, feedback, tutoring dialogue) [28].

2.1.2 Multilingual and Low-Resource Settings

Most early systems targeted English and a few high-resource languages. Extending content generation to under-resourced languages introduces challenges of data scarcity, morphology (e.g., agglutination), and orthography (e.g., abjad scripts), which complicate tokenization, agreement, and error-sensitive feedback. Position papers and roadmaps argue that multilingual LLMs can act as an "off-ramp" toward digital language equality by enabling cross-lingual transfer and few-shot generation while dedicated datasets are developed [29]. In practice, multilingual models (e.g., mBERT/XLM-R/T5 variants) and workflows that combine synthetic data, translation, and post-editing are increasingly used to bootstrap grade-appropriate texts, questions, and explanations for low-resource lan-

guages, while preserving curricular alignment and cultural relevance [28, 29].

2.1.3 Frameworks and Systems

Beyond point techniques, recent work emphasizes frameworks that integrate LLMs with explicit pedagogy, curricula, and quality control. Curriculum-oriented generation frameworks constrain LLMs with target concepts, readability bands, and vocabulary lists to produce grade-appropriate materials, pairing controllable generation with multi-dimensional evaluation [30]. Human-in-the-loop production pipelines—as popularized in large-scale language learning platforms—combine prompt templates, content rules, and expert review to accelerate lesson and item creation while maintaining quality and adherence to learning objectives [31].
LLM outputs are also used upstream to synthesize *instructional* datasets (e.g., student-like responses, error patterns, diverse prompts) that train smaller task models or power large-scale evaluation of item validity and difficulty [32, 33]. To make deployment practical in schools and on local devices, knowledge distillation pipelines transfer the capabilities of large teacher models into compact student models, achieving substantial efficiency gains with minimal loss in accuracy for tasks such as short-answer scoring or feedback classification [34]. Taken together, these directions illustrate a shift from ad-hoc prompting to *pedagogy-aware* generation frameworks that scaffold LLM creativity with curricular structure, guardrails, and human oversight—a trajectory that aligns with scalable, multilingual educational content creation [28, 30].

2.2 Crossword puzzles Generation

Crossword puzzles, traditionally enjoyed for their challenge and entertainment value, are increasingly recognized for their educational potential. These puzzles facilitate learning in multiple disciplines, such as history, science, and linguistics, and are particularly effective in enhancing vocabulary and spelling skills [35, 36, 37]. Their capacity to engage and educate simultaneously makes them invaluable in pedagogical contexts.
In language acquisition and the mastery of specialized terminology, crossword puzzles stand out as exceptional tools. They offer an interactive learning experience that promotes both the retention of technical jargon and general language skills [38, 39, 40]. This dynamic learning approach supports the cognitive development of learners by improving critical thinking abilities and memory retention [41, 37, 42, 43, 36, 44, 45].
The field of crossword puzzle generation has drawn significant academic interest due to its complex nature. Researchers have experimented with a wide array of approaches including the use of vast lexical databases and embracing the potentials of modern Large Language Models (LLMs), which accommodate innovative methods like fine-tuning and zero/few-shot learning.
Pioneer works in this domain have been diverse and innovative. For instance, Rigutini et al. successfully employed advanced natural language processing (NLP) techniques to automatically generate crossword puzzles directly from internet-based sources, marking

a foundational advancement in this field [14, 15]. In addition, Ranaivo et al. devised an approach based on NLP that utilizes text analytics alongside graph theory to refine and pick clues from texts that are topic-specific [16]. Meanwhile, aimed at catering to Spanish speakers, Esteche and his group developed puzzles utilizing electronic dictionaries and news articles for crafting clues [17]. On another front, Arora et al. introduced SEEKH, which merges statistical and linguistic analyses to create crossword puzzles in various Indian languages, focusing on keyword identification to structure the puzzles [46]. While the literature shows progress in utilizing LLMs for crossword generation, existing approaches often rely on generalized models or few-shot learning techniques. The unique characteristics and challenges presented by the Arabic language, however, have remained largely unexplored in the context of crossword puzzle generation from a given text. Previous attempts in Arabic have typically employed LLMs "as they are" using few-shot learning. In contrast, our current project introduces a dataset specifically designed for this task in Arabic. Furthermore, we introduce open-source models that are fine-tuned to better accommodate and enhance performance for this specific application.
This study breaks new ground by employing cutting-edge language modeling techniques explicitly to create English, Italian, Turkish and Arabic crossword puzzles from a given text. This approach not only fills a gap within the scholarly literature but also enhances the tools available for language-based educational initiatives, thereby advancing the field of Multilingual crossword puzzle generation.

2.3 Question Generation

The automatic generation of high-quality questions, a task known as Question Generation (QG), is a cornerstone of educational technology and natural language processing (NLP). This section reviews the evolution of QG methodologies, the datasets that have enabled this progress, and the specific frameworks and considerations relevant to creating effective educational assessments, particularly for low-resource languages like Persian and Turkish.

2.3.1 The Evolution of Question Generation Methods

Early approaches to QG were predominantly based on rule-matching and template-based systems, which, while effective, lacked flexibility and scalability [47]. The advent of neural networks marked a significant shift, with **Sequence-to-Sequence (Seq2Seq) models**, often enhanced with attention mechanisms, becoming the standard [48]. These models learn to transform a given context into a question, and further research integrated linguistic features to improve grammatical and semantic coherence [49].
Subsequent advancements have explored more complex architectures and training paradigms. **Multi-task learning** frameworks were developed to jointly train QG with related tasks like Question Answering (QA), leveraging their inherent synergy [50]. Other approaches have incorporated **reinforcement learning** to refine question quality based on downstream metrics [51] and utilized **multi-modal inputs** to generate questions from both text and images [52]. The rise of language models (LMs) like BERT and GPT-3 has fundamentally transformed the field, enabling the generation of more natural and contextually

sophisticated questions through pre-training and fine-tuning [53, 54].

2.3.2 Datasets for Question Generation

Progress in QG has been closely tied to the availability of high-quality datasets. Many foundational models were trained on datasets originally created for QA, which provide rich context-question-answer triplets. Prominent examples include **SciQ**, **RACE**, and **FairytaleQA**, which have been widely repurposed for QG research [55, 19, 56, 57, 58, 59, 60].
Recognizing the unique requirements of QG, researchers have also developed specialized datasets. Datasets such as **LearningQ**, **KHANQ**, and **EduQG** were created specifically for educational QG, covering a wide range of subjects and academic levels [61, 62, 63]. However, these resources are overwhelmingly concentrated in high-resource languages, leaving a significant gap for languages like Persian and Turkish.

2.3.3 Advanced Frameworks and Answer-Aware Generation

A key challenge in QG is ensuring that the generated question is relevant to a specific piece of information, or "answer," within the context. This has led to a distinction between **answer-agnostic** (generating a general question from a passage) and **answer-aware** (generating a question about a specific answer) QG [64]. To incorporate answer information, researchers have proposed various techniques, such as encoding the answer along with the context, using answer position markers, or generating questions from text summaries that highlight key information [65, 66].
More recently, sophisticated data generation frameworks have emerged. **AgentInstruct**, for example, is an extensible framework that automatically generates large volumes of high-quality synthetic data from raw sources like text documents [67]. By transforming unstructured text into structured content and using iterative refinement, AgentInstruct can produce diverse and complex instruction-following datasets. This methodology is highly relevant for generating Multiple-Choice Questions (MCQs), as it can facilitate the creation of varied questions and plausible distractors, which is essential for building robust training data for MCQ generation models.

2.3.4 QG in Educational Applications

In educational settings, controlling the quality and characteristics of generated questions is paramount. A critical aspect is **question difficulty**, which can be assessed based on factors like answerability, the number of inference steps required, or learners' historical performance [68, 69, 70]. Another important consideration is ensuring that questions align with a given **syllabus** or curriculum. To this end, studies have focused on training models to classify or rank questions based on their relevance to specific learning objectives [71].
Furthermore, the goal of **personalized education** has driven the development of models that can generate customized questions tailored to individual students. This often involves knowledge-tracking models that adapt to a student's answer history or few-shot

models that incorporate sequences of student states to generate the most appropriate next question [72, 73]. The development of these advanced educational tools, however, remains a significant challenge for low-resource languages due to the lack of foundational datasets.

2.4 Automatic Syntax Feedback System

The integration of NLP with educational technology has produced two closely related strands of research: *Automated Essay Scoring* (AES) and *Essay Feedback Generation*. AES targets reliable, scalable assessment, whereas feedback generation aims to deliver formative guidance that improves learner writing. Below, we synthesize prior work in both areas and position our proposed *Automatic Syntax Feedback System* within this landscape.

2.4.1 Automated Essay Scoring

Classic AES systems follow a two-step pipeline: (i) derive features that capture lexical, syntactic, discourse, and stylistic properties, and (ii) train regressors or classifiers to predict holistic scores [74, 75, 76, 77, 78]. Early neural methods replaced much of the feature engineering with learned representations, using CNNs/LSTMs and attention to model document-level signals [79, 80, 81]. In parallel, pretrained word embeddings such as word2vec and GloVe provided stronger lexical priors for AES pipelines [82, 83].
Transformer-based language models have since become the dominant backbone. BERT and its derivatives have been adapted for essay scoring via fine-tuning schemes that combine regression and pairwise ranking, and via multi-scale essay representations [84, 85, 86, 87]. Data augmentation has further improved robustness, especially for short answers [88]. A persistent challenge is *domain shift* across prompts: prompt-agnostic and cross-prompt generalization remain non-trivial [89].
The emergence of instruction-following large language models (LLMs) has renewed interest in zero-/few-shot AES. Empirical studies report mixed outcomes: while models like GPT-4 can approximate human ratings in some settings (e.g., CEFR for short L2 essays or discourse coherence proxies), they do not uniformly surpass strong supervised AES baselines and are sensitive to rubric design and prompt wording [90, 91, 92]. These findings underscore a gap between general-purpose LLM capabilities and rubric-grounded, reliable assessment.
Our *Automatic Syntax Feedback System* leverages LLMs not primarily for holistic scoring but for *targeted, rubric-informed syntactic guidance*. By emphasizing sentence-level syntactic quality and clarity—dimensions long recognized as important to coherence and readability [75]—our approach complements AES: it can support scoring pipelines while providing actionable, fine-grained suggestions that directly address common learner needs.

2.4.2 Essay Feedback Generation

Recent surveys highlight both the promise and risks of LLMs in education, especially for scalable, personalized feedback [93, 94]. Beyond raw generative ability, two strategies appear crucial for educational feedback quality: (i) grounding the model in *golden knowledge* (e.g., explicit rubrics, canonical examples, and accurate model answers) and (ii) closing the loop with automated critique and revision [95]. Systems that jointly tackle scoring and feedback (e.g., FABRIC) illustrate the potential of pairing rubric-aware evaluation with constructive suggestions [96], while broader analyses argue for a shift from *automation* to *augmentation*—using LLMs to amplify teachers' and learners' efforts rather than replace them [97].

Our system operationalizes these insights for syntax-focused feedback: it enriches LLMs with rubric explanations and pedagogically grounded exemplars, then elicits concise, constructive suggestions that are localized to sentences and phrases. This rubric-grounded, syntactic emphasis is designed to make feedback immediate, specific, and instructionally useful, thereby complementing holistic AES and advancing practical support for writing improvement [93, 95, 96].

2.5 Integrative Synthesis and Thesis Positioning

Across the four strands reviewed—pedagogy-aware content generation frameworks, crossword generation, question generation (QG), and automated syntax feedback—three themes recur: (i) scale versus specificity, (ii) multilingual equity, and (iii) pedagogy and reliability. Early ITS and rule-/template-heavy pipelines showed that automation is possible but depend on costly authoring and struggle to meet classroom constraints on difficulty, grounding, and format without heavy human oversight [20, 21, 22, 23]. Recent instruction-following LLMs broaden capability, yet naïve prompting can drift from sources, produce uneven difficulty, and weaken quality control, especially outside English [26, 27, 28]. These tensions are most acute in low-resource, morphologically complex, or right-to-left languages where tokenization, orthography, and evaluation complicate deployment [29].

In crossword generation, much prior work either couples rules and dictionaries or treats LLMs as generalists with few-shot prompts [14, 15, 16, 17, 46]. Under-explored is *educational* clueing from source texts with clear accuracy, clarity, and text-grounding guarantees—and doing so across typologically diverse languages. In QG, the field moved from templates and early neural models to transformer-based and instruction-following LLMs [24, 48, 25, 54, 28], but educational settings impose extra constraints: answer-awareness, distractor plausibility without ambiguity, depth of comprehension beyond surface recall, and difficulty control aligned to syllabi—challenges compounded in Persian and Turkish due to resource scarcity and morphology [64, 65, 66, 29, 71]. For writing support, research on AES and feedback has advanced, yet general-purpose LLMs can over-correct or give generic advice; sentence-level, minimal-edit, rule-aware feedback that "does no harm" when uncertain remains a gap for trustworthy classroom use [74, 90, 94, 93].

These observations motivate a *framework*, not just a model: one that converts abundant raw text into *pedagogy-aligned artifacts* at scale; operates reliably across languages; and

is feasible to deploy as open, efficient systems. This thesis responds with a two-stage distillation pipeline that (1) uses a high-capacity teacher LLM to synthesize *task-structured, source-grounded* supervision (e.g., (Text) → (Clue, Answer);
(Text) → (MCQ with calibrated distractors);
(Essay sentence) → (Minimal correction, explanation)), and (2) instruction-tunes compact open models (students) that inherit these capabilities with better efficiency and controllability [26, 32, 33, 34]. Beyond reducing reliance on proprietary teachers, this approach institutionalizes *guardrails*—grounding, task schemas, and verification—so outputs serve explicit learning objectives [30, 31, 95, 96].

How this framework reconciles gaps in prior work.

- **Scale with structure.** Rather than ad-hoc prompting, the pipeline mass-produces *Instruct*-style datasets that encode the target artifact and rubric, turning general LLM competence into consistent educational outputs [26, 32, 33].
- **Multilingual equity.** By distilling into open models and curating language-aware instructions (e.g., diacritic normalization, agglutinative morphology, RTL handling), the approach mitigates English-centric bias and enables tools for Italian, Arabic, Turkish, and Persian [29, 28].
- **Pedagogical fidelity and reliability.** Tasks are designed to be answer-aware, minimally edited, and difficulty-calibrated; verification filters and rubric-grounded rationales reduce ambiguity, over-correction, and hallucinations that undermine classroom trust [95, 96, 94].

This synthesis sets up the thesis' **main research question**—how to systematically adapt general LLMs to generate diverse, high-quality educational content for under-resourced languages—and three task-specific sub-questions: multilingual educational crosswords (RQ1), assessment-grade QG in Persian/Turkish (RQ2), and precise syntax feedback in English (RQ3). Each task inherits the same design DNA (text grounding, task schemas, calibration), demonstrating that a *single distillation methodology* can span gamified practice, formal assessment, and corrective feedback [28, 29, 30].

2.6 Knowledge Distillation: Historical Roots and Modern Architectures

The democratization of Large Language Models (LLMs) is currently predicated on the paradigm of *Knowledge Distillation* (KD)—the process of transferring capabilities from a high-capacity "teacher" model to a more efficient "student" model. While frequently framed as a recent innovation necessitated by the size of Transformer models, the theoretical and methodological foundations of distillation were established nearly a decade prior to the deep learning revolution. This section provides a rigorous historical review of

KD, traces its evolution from ensemble compression to generative sequence approximation, and analyzes the specific algorithmic challenges of distilling probabilistic generative models.

2.6.1 The Pre-Deep Learning Era: Model Compression (2006)

The intellectual lineage of distillation begins with the work of [98] on "Model Compression." In the mid-2000s, the state-of-the-art in machine learning was dominated by massive ensembles of decision trees and bagged classifiers. While these ensembles offered superior predictive performance by averaging out the variances of individual estimators, they incurred prohibitive computational costs during inference, requiring the storage and execution of thousands of sub-models.

Bucilă et al. proposed a solution that remains the blueprint for modern KD: they demonstrated that a single neural network (the student) could approximate the function learned by a massive ensemble (the teacher) if trained on a sufficiently large dataset labeled by the teacher. Crucially, they introduced the method of *MUNGE* (Model Unknown Data Generation), which synthesized pseudo-data to probe the decision boundaries of the teacher in regions sparse in the original training data. They observed that the teacher's predictions provided a "richer" supervisory signal than the original one-hot ground truth labels. By training the student to minimize the Root Mean Squared Error (RMSE) between its logits and the ensemble's logits, they successfully compressed the complex decision surface of the ensemble into a deployable neural network, achieving accuracy comparable to the ensemble with a fraction of the inference latency.

2.6.2 The Formalization of Dark Knowledge (2015)

The resurgence of deep neural networks led [99] to formalize and generalize these findings in their seminal paper, "Distilling the Knowledge in a Neural Network." Hinton et al. argued that the "knowledge" of a trained model is not solely contained in its final discrete decisions but in the *Dark Knowledge*—the rich information hidden in the relative probabilities assigned to incorrect classes.

In a standard classification task (e.g., MNIST), a high-capacity model might classify an image of a "3" with a probability of 0.99. However, the remaining 0.01 probability mass is not uniformly distributed; the model might assign 10^{-3} to "8" (because 3s look like 8s) and 10^{-9} to "7". This ratio (10^{-3} vs 10^{-9}) encodes structural similarity information about the data manifold that is destroyed when using hard labels. To expose this dark knowledge to the student, Hinton introduced the *temperature* hyperparameter T in the softmax function. The probability q_i of class i is computed from logits z_i as:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{2.1}$$

As $T \to \infty$, the probability distribution flattens, amplifying the gradients from the non-target classes. The student model is trained to minimize a combined objective function consisting of the standard cross-entropy loss with ground truth labels (to ensure

accuracy) and the Kullback-Leibler (KL) divergence with the teacher's softened output distribution (to inherit generalization).

2.6.3 Distillation in the Transformer Era: Sequence-Level and Feature-Based Approaches

The application of KD to Natural Language Processing (NLP) introduced unique challenges, primarily the shift from classification (discriminative) to generation (generative). [100] pioneered *Sequence-Level Knowledge Distillation* (SeqKD) for Neural Machine Translation. They demonstrated that training a student to mimic the teacher's token-level probabilities (Word-Level KD) was insufficient for capturing long-range dependencies. Instead, they proposed training the student on the *beam search outputs* of the teacher. Effectively, the teacher's generated sequence becomes the new ground truth. This simplifies the learning landscape for the student: the teacher's output distribution typically has lower entropy and fewer modes than the real data distribution, allowing the student to approximate the teacher's greedy decoding path with high fidelity.

Parallel to response-based distillation, *Feature-Based Distillation* emerged to address the depth of modern networks. [101] introduced "FitNets," which utilized regressor heads to align the intermediate hidden states of the student with those of the teacher. This forces the student to mimic not just the final decision, but the hierarchical feature extraction process of the teacher. In the context of BERT and Transformers, this evolved into mapping attention matrices, ensuring that the student model attends to the same linguistic dependencies (e.g., subject-verb agreement heads) as the teacher [102].

2.6.4 The Divergence Problem in Generative LLMs

A critical theoretical divergence exists in distilling autoregressive LLMs. Standard KD minimizes the *Forward KL Divergence* ($D_{\mathrm{KL}}(P \parallel Q)$), where P is the teacher and Q is the student. Forward KL has a "zero-avoiding" or "mean-seeking" behavior: it forces the student to cover the entire support of the teacher's distribution.

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{2.2}$$

Because high-capacity teachers (e.g., GPT-4) assign non-zero probability to a vast range of diverse text continuations, a low-capacity student attempting to cover this wide distribution often spreads its probability mass too thin, resulting in incoherent text (the "void" region problem).

Recent advancements, such as **MiniLLM** [103], propose minimizing the *Reverse KL Divergence* ($D_{\mathrm{KL}}(Q \parallel P)$). Reverse KL has a "zero-forcing" or "mode-seeking" behavior. It heavily penalizes the student for assigning probability to regions where the teacher has low probability, but it does not penalize the student for missing some of the teacher's modes. This encourages the student to focus on the most likely, high-quality responses (the modes) and ignore the long tail, resulting in more precise and reliable generation at the cost of diversity.

2.6.5 Black-Box Distillation: Self-Instruct and Evolutionary Heuristics

In the current ecosystem, access to the weights or logits of state-of-the-art models (like GPT-4) is often restricted, giving rise to *Black-Box Distillation* or Instruction Tuning. Here, the student learns solely from the text generated by the teacher.

- **Self-Instruct:** [104] demonstrated that a model could bootstrap its own supervision. Starting with a small seed of human-written tasks, the model generates new instructions and input-output pairs, filters them for diversity, and then fine-tunes itself on this synthetic dataset.
- **Evol-Instruct:** To prevent the student from plateauing on simple tasks, [105] introduced "Evol-Instruct." This method uses specific heuristics (In-Depth Evolving) to rewriting instructions to be explicitly more complex—adding constraints, increasing reasoning steps, or complicating inputs. This creates a curriculum of increasing difficulty, effectively distilling the teacher's reasoning capabilities rather than just its linguistic surface forms.

2.7 Reliability and Failure Modes: Hallucination and Bias in Probabilistic Models

The deployment of Large Language Models (LLMs) in educational settings is often predicated on an assumption of reliability that contradicts the fundamental architecture of these systems. As probabilistic next-token predictors, LLMs operate without a ground-truth model of the world, leading to intrinsic failure modes that are not merely bugs but artifacts of their design. This section provides a taxonomy of these failures, specifically *Hallucination* and *Bias*, and analyzes their implications for educational content generation.

2.7.1 The Stochastic Parrot Hypothesis

The foundational critique of LLM reliability is articulated by [106] in their characterization of LLMs as "Stochastic Parrots." They argue that LLMs stitch together linguistic forms based on probabilistic dependencies observed in training data, without access to meaning (semantics) or communicative intent. Unlike a human educator who constructs an explanation based on a verifyable truth, an LLM generates text based on what is *statistically plausible*. This disconnect between form and meaning is the root cause of hallucination: the model can generate fluent, authoritative-sounding assertions that are factually vacuous.

2.7.2 Taxonomy of Hallucinations

In the context of Natural Language Generation (NLG), hallucination is defined as generated content that is nonsensical or unfaithful to the source. [107] and [108] provide a

rigorous taxonomy distinguishing between two primary categories:

Factuality Hallucinations (Open-Domain)

Factuality hallucinations occur when the model fabricates information that contradicts established real-world knowledge. This is prevalent in open-domain question answering where the model relies on its compressed parametric memory.

- **Fabrication:** The invention of non-existent entities, events, or citations. For example, in academic writing tasks, LLMs frequently generate plausible-looking but non-existent bibliographic references (the "hallucinated citation" phenomenon).
- **Misattribution:** The conflation of properties between two real entities (e.g., attributing a theory by Piaget to Vygotsky). This often occurs because the entities share similar embedding spaces in the model's high-dimensional representation.

Faithfulness Hallucinations (Closed-Domain)

Faithfulness hallucinations occur in grounded tasks, such as summarization or Reading Comprehension, where the model is provided with a specific source text.

- **Intrinsic Hallucination:** The generated output logically contradicts the provided source text. This represents a failure of reasoning or comprehension.
- **Extrinsic Hallucination:** The generated output contains information not present in the source text. While the information might be factually true in the real world (e.g., adding the capital of a country mentioned in the text), it constitutes a hallucination in the context of the task because it violates the closed-world assumption of the prompt.

2.7.3 Sociotechnical Bias: Allocational vs. Representational Harms

Beyond factual errors, LLMs reproduce and amplify the systemic biases present in their training corpora. Following the framework established by [109] and [110], these biases can be categorized into two distinct types of downstream harm:

Allocational Harms

Allocational harms arise when an automated system unfairly distributes resources or opportunities among social groups. In the educational context, this is a critical risk in **Automated Essay Scoring (AES)** and admissions screening.

- **Linguistic Bias:** Studies have shown that LLM-based detectors and graders systematically penalize text written by non-native English speakers [111]. These models often use *perplexity* (a measure of text complexity and unpredictability) as a proxy for quality or human-likeness. Non-native speakers, who may use simpler or more formulaic constructs, receive higher perplexity scores (interpreted as "AI-generated") or lower quality ratings, regardless of the argumentative coherence of their work.

Representational Harms

Representational harms involve the reinforcement of subordination, negative stereotypes, or erasure, independent of resource allocation.

- **Associative Hallucination:** LLMs exhibit strong associative biases. When generating content about specific roles (e.g., "nurse," "engineer"), models often hallucinate gender or racial attributes based on stereotypes, even when such information is not specified in the prompt.

- **Cultural Erasure:** The dominance of English and Western-centric data in training corpora (e.g., Common Crawl) leads to a "hegemonic worldview" where the model defaults to Western cultural norms, obscuring or misrepresenting indigenous or non-Western perspectives in history or social studies content.

2.8 Pedagogical Validity: The Tension Between Generation and Learning

The integration of Generative AI into education has precipitated a crisis of validity. While AI tools demonstrate an unprecedented capacity for *content generation*—rapidly producing lesson plans, quizzes, and feedback—this capability is distinct from, and occasionally antithetical to, the *validation of learning outcomes*. This section examines the pedagogical implications of AI through the lenses of the TPACK framework, psychometric validity, and cognitive science.

2.8.1 Framework Evolution: From TPACK to AI-TPACK

The effective integration of technology in education has traditionally been theorized using the **Technological Pedagogical Content Knowledge (TPACK)** framework [112]. TPACK posits that expert teaching emerges from the intersection of Content Knowledge (CK), Pedagogical Knowledge (PK), and Technological Knowledge (TK). However, traditional TPACK treats technology as a passive tool (e.g., a whiteboard or calculator) that is operated by the teacher.
Generative AI disrupts this model because it possesses *agency*; it acts as a semi-autonomous content creator. To address this, scholars have proposed **AI-TPACK** or Intelligent-TPACK [113]. This extended framework requires:

- **AI-TK (AI-Technological Knowledge):** Understanding the specific affordances and limitations of LLMs, such as their probabilistic nature, token limits, and hallucination rates.

- **AI-PK (AI-Pedagogical Knowledge):** Knowledge of how to use AI to support specific pedagogical strategies (e.g., scaffolding, personalized feedback) without undermining student agency.

- **Ethical Knowledge:** A critical component often added to AI-TPACK is the ability to evaluate the bias and integrity of AI-generated materials before they reach the student.

Without this specialized knowledge, teachers risk what is termed "automation bias"—the tendency to uncritically accept the output of an automated system as correct, leading to the propagation of errors in the classroom.

2.8.2 The Validity of Automatic Item Generation (AIG)

A primary use case for LLMs is *Automatic Item Generation* (AIG)—the creation of Multiple Choice Questions (MCQs) and assessments. [114] established that valid AIG requires rigid cognitive models to ensure that items map correctly to specific learning objectives. LLM-based generation often bypasses this rigorous structure.

Recent empirical studies utilizing **Item Response Theory (IRT)** have highlighted significant threats to the validity of AI-generated items [115].

- **Distractor Efficiency:** A critical component of MCQ validity is the quality of distractors (wrong answers). Effective distractors must be plausible to a student with partial knowledge. [116] found that while LLMs generate grammatically perfect items, the distractors are often nonsensical or easily eliminated by test-taking strategies (e.g., length cues), resulting in items with low *discrimination indices* (a-parameter in IRT).
- **Construct Irrelevance:** LLMs tend to generate items that test surface-level recall (Bloom's Taxonomy Level 1) rather than higher-order thinking, even when prompted otherwise. This introduces construct under-representation, where the test fails to measure the intended depth of student understanding.

2.8.3 Cognitive Offloading and "Desirable Difficulties"

Perhaps the most profound pedagogical risk is the phenomenon of **Cognitive Offloading**. Learning requires what [117] termed *Desirable Difficulties*—conditions of instruction that appear to slow the rate of acquisition but actually enhance long-term retention and transfer. These include retrieval practice (struggling to recall information) and generative processing (struggling to formulate an answer).
Generative AI acts as a "friction-reduction" technology. By providing instant summaries, code completion, or essay drafts, it removes the very struggle necessary for memory encoding. The "Lazy Thinker" hypothesis suggests that humans are cognitive misers who will naturally offload cognitive effort to the path of least resistance. Recent controlled experiments [118, 119] confirm this: students who used AI tutors to solve problems performed better during the practice phase (performance) but significantly worse on subsequent unassisted exams (learning) compared to students who struggled without AI. This dissociation between immediate performance and long-term learning indicates that unguided AI use can create an "illusion of competence" while actively atrophyng deep learning capabilities.

Part II

Methodological Framework

Chapter 3

A Unified Methodological Framework: From Language to Learning

3.1 Introduction: The Grand Challenge and a Principled Solution

The promise of modern educational technology to deliver truly personalized and scalable learning experiences has long been hindered by a fundamental bottleneck: the creation of high-quality, pedagogically sound, and culturally relevant content. While digital platforms have matured as delivery mechanisms, the educational "fuel"—the vast array of practice items, formative assessments, and targeted feedback—remains a scarce resource, particularly for non-English languages. This scarcity creates a digital language divide, obstructing educational equity and concentrating the benefits of AI-enabled learning among speakers of a few dominant languages. Hand-crafting bespoke materials is economically and logistically infeasible at the scale required to serve the long tail of learning, which encompasses a vast spectrum of subjects, proficiency levels, and pedagogical contexts.

The advent of Large Language Models (LLMs) presents a paradigm shift, offering a potential solution to this content generation challenge. However, the naïve prompting of general-purpose models is insufficient for the rigorous demands of education. Unconstrained outputs frequently hallucinate facts, drift from source materials, and produce feedback that is too vague to be instructionally useful, thereby undermining the reliability required for classroom use. Moreover, an exclusive reliance on proprietary, closed-source "teacher" models limits accessibility, auditability, and sustainable deployment in real-world educational settings.

This thesis contends that bridging the gap between general LLM competence and the specific, nuanced needs of multilingual education requires a systematic, replicable, and robust methodology. In response, this work introduces and validates the **"From Language to Learning" framework**, a unified, LLM-driven approach for transforming powerful, general-purpose models into highly specialized, efficient, and open-source educational tools. The framework is architected around a novel **two-stage distillation process** designed to systematically transfer the sophisticated capabilities of a large "teacher" model into a smaller, more accessible "student" model, optimized for a specific pedagogical task.

This chapter provides a comprehensive exposition of this framework. It begins by outlining the conceptual foundations and design principles that guide its architecture. It then delves into a detailed examination of the two-stage distillation process: Stage 1,

the synthesis of high-quality, task-specific instructional datasets (the *Instruct series*); and Stage 2, the instruction fine-tuning process that creates specialized student models. Subsequently, it abstracts the framework's core functional components—ingestion, generation, validation, and evaluation—which are consistently applied across all tasks. Finally, it synthesizes how this unified methodology is instantiated across the three distinct pedagogical domains explored in this thesis: gamified learning, formal assessment, and writing assistance.
The framework is explicitly designed to meet five critical objectives:

- **Scale:** To transform abundant, unstructured raw text into structured, pedagogically valuable artifacts for a wide array of topics.
- **Pedagogical Fidelity:** To encode specific task structures (e.g., crossword clue formats, MCQ distractor quality) and allow for the calibration of difficulty and content alignment.
- **Multilingual Equity:** To operate effectively across typologically diverse languages, including under-resourced ones, without requiring bespoke, hand-engineered pipelines for each language.
- **Efficiency and Openness:** To distill capabilities into smaller, open-source models that are suitable for cost-effective and local deployment, thereby democratizing access to AI-powered educational tools.
- **Reliability and Evaluation:** To ground generated content in source material, control for hallucinations, and validate outputs through a combination of systematic human and automatic measures.

By operationalizing these principles, the "From Language to Learning" framework offers a validated blueprint that bridges the gap between general-purpose language models and the specific needs of multilingual education, paving the way for the next generation of AI-powered learning technologies.

3.2 Architectural Deep Dive: The Two-Stage Distillation Process

At the heart of the framework lies a systematic, two-stage distillation process conceived to create high-quality, task-specific instructional datasets and use them to specialize smaller, open-source models. This teacher-student paradigm is designed to be both scalable and cost-effective, leveraging the advanced reasoning of a frontier model to create supervision data that is perfectly aligned with the target educational function.

3.2.1 Stage 1: Synthetic Data Generation via Expert Prompting

The first and most critical stage of the framework is the creation of a large-scale, structured, and high-quality instructional dataset, which we term the ***Instruct series***. This

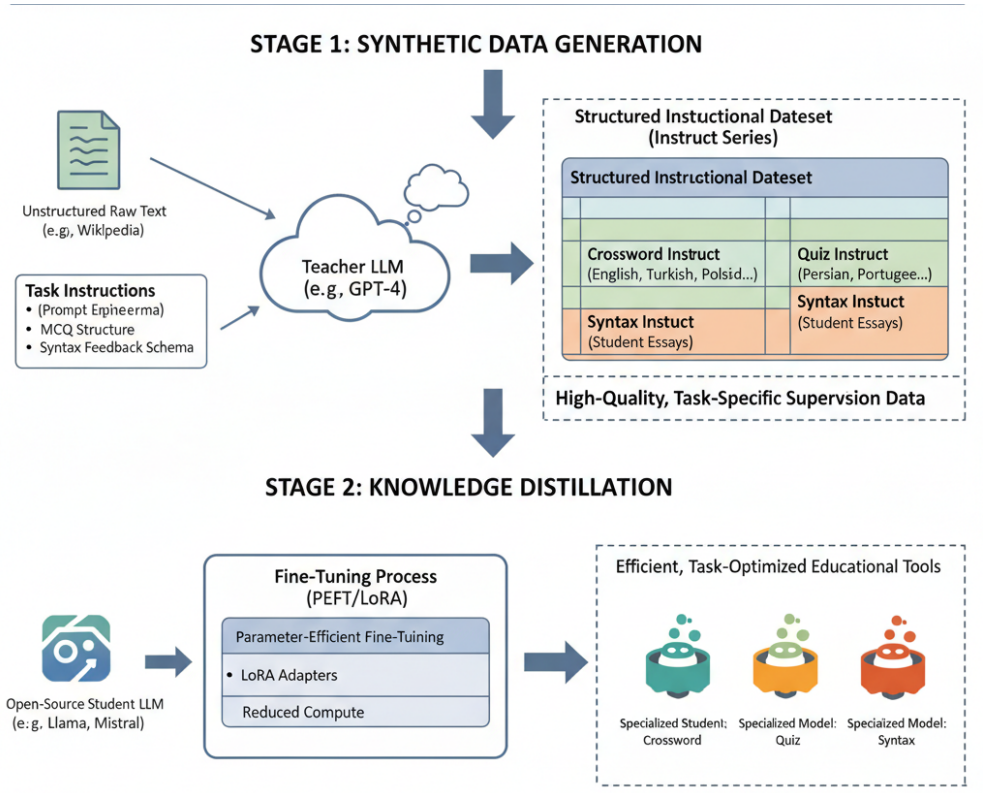


Figure 3.1: A high-level schematic of the "From Language to Learning" framework. Stage 1 involves a "teacher" LLM transforming unstructured text and task instructions into a structured *Instruct Series* dataset. Stage 2 uses this dataset to fine-tune a smaller, open-source "student" LLM, resulting in a specialized and efficient educational tool.

process transforms raw, unstructured text into tailored supervision data by prompting a state-of-the-art "teacher" LLM to act as an expert content creator.

The Role of the "Teacher" Model

The success of this stage is fundamentally dependent on the capabilities of the teacher model, which is typically a large, proprietary LLM (e.g., GPT-4) chosen for its superior instruction-following, reasoning, and multilingual generation abilities. Its role is to execute a complex transformation, mapping unstructured input to a structured output format defined by the pedagogical task. For the diverse applications in this thesis, these transformations included:

- **For Crossword Generation:** $(Text) \rightarrow (Clue, Answer, Category)$.
- **For Question Generation:** $(Text) \rightarrow (Question, CorrectAnswer, Distractors)$.

- **For Syntax Feedback:** $(EssaySentence) \rightarrow (CorrectedSentence, Feedback)$.

The quality ceiling of the entire framework is set by the proficiency of this teacher model. If the teacher model hallucinates facts, misunderstands pedagogical nuances, or introduces biases from its pre-training data, these flaws are directly encoded into the synthetic dataset, with the risk of being learned and amplified by the student model in Stage 2. This dependence constitutes a foundational limitation, creating a "proprietary bottleneck" and underscoring the need for careful validation of the generated seed data. For instance, a quality spot-check was performed on the seed feedback for the syntax task, where human raters confirmed that the vast majority of items were of high quality (rated 'A' or 'B'), thereby validating its suitability as supervision.

The Art of Prompt Engineering for Pedagogy

The generation of high-quality synthetic data is highly sensitive to the design of the instructional prompts. Achieving the desired output structure, style, and pedagogical quality requires extensive iteration and sophisticated prompt engineering, a process that is more akin to crafting a detailed rubric or a software specification than simply asking a question. The brittleness of prompts—where minor changes in wording can lead to significant variations in output—necessitates a meticulous design process. This engineering overhead is a significant aspect of the framework, requiring expertise in both the educational domain and LLM behavior.

Across the thesis, several advanced prompting strategies were employed:

- **Multi-Stage Prompt Cascades:** For complex tasks like Persian MCQ generation, a single prompt proved insufficient. A three-stage cascade was designed: (1) an *augmentation* prompt to normalize content, (2) a *generation* prompt to draft the MCQ, and (3) a *refinement* prompt to control for leakage and calibrate distractors. This division of labor ensured higher quality and consistency.

- **Category-Structured Schemas:** For the syntax feedback task, the prompt enforced a strict seven-category schema (e.g., Misspelled Words, Punctuation, Articles). It required the model to return either a list of 'Error -> Correct version' pairs or 'N/A' for each category, a format that imposes structure and ensures comprehensive yet precise feedback.

- **Constraint-Driven Generation:** For all tasks, prompts included explicit positive and negative constraints. For educational crosswords, prompts instructed the model to generate factual, non-leaking clues grounded in a source text. For Turkish quizzes, the prompt enforced clarity, key uniqueness, and plausible distractors.

Source Material Curation and Preprocessing

The framework's scalability hinges on its ability to leverage abundant unstructured text. The choice and preparation of this source material significantly influence the final educational content.

- **Source Selection:** A substantial portion of the source material for the crossword and question generation tasks was derived from Wikipedia across all five languages. This choice provides broad topical coverage but also inherits Wikipedia's known biases in style, perspective, and domain representation. For the syntax feedback task, the source material was a corpus of authentic student essays, which required careful ethical handling and anonymization.

- **Preprocessing and Refinement:** Raw text is rarely suitable for direct use. Preprocessing steps were essential, including filtering pages by length and relevance, and cleaning markup. A notable example is the **placeholder repair** step for student essays, where a dedicated prompt was used to replace anonymization tokens (e.g., '@PERSON1') with contextually plausible, non-identifying surrogates to improve readability and prevent the model from misinterpreting them as errors.

3.2.2 Stage 2: Instruction Fine-Tuning and Knowledge Distillation

In Stage 2, the synthetic *Instruct* data $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ from Stage 1 is used to specialize a smaller, open-source *student* model p_θ (e.g., Llama/Mistral, 7–13B parameters) via instruction fine-tuning. The goal is to transfer the teacher's task competency into a compact model that is efficient, auditable, and deployable in educational settings. As shown in our experiments, this step is decisive for reliability: base models perform poorly off-the-shelf, while fine-tuned students achieve large gains in automatic and human evaluations.[1]

The Role and Advantages of the "Student" Model

The student models are typically open-source, instruction-tuned LLMs from families like Llama or Mistral, with parameter counts in the 7B to 13B range. The choice to distill knowledge into these models is driven by several key advantages that are central to the framework's mission:

- **Accessibility and Cost-Effectiveness:** Fine-tuning is orders of magnitude cheaper than pre-training a model from scratch. Using open-source models eliminates the costs and access restrictions associated with proprietary APIs, making the final tool deployable in a wider range of educational settings.

- **Democratization and Auditability:** The resulting specialized models are open-source, allowing them to be freely used, audited for biases, and improved by the broader educational and research community. This transparency is crucial for building trust and fostering collaborative development in educational technology.

- **Specialization and Efficiency:** The fine-tuning process creates a specialized expert, optimized to perform its designated educational function with high accuracy,

[1] See Chapter 8 results for Turkish MCQ/SAQ and the associated human ratings.

consistency, and efficiency. As demonstrated across all tasks, these specialized models consistently outperform their general-purpose base versions, which often struggle with specific formatting and pedagogical constraints.

The Mechanism of Parameter-Efficient Fine-Tuning (PEFT)

To make the fine-tuning process computationally tractable and efficient, the framework employs **Parameter-Efficient Fine-Tuning (PEFT)** techniques, most notably **Low-Rank Adaptation (LoRA)**. Instead of updating all of the model's billions of parameters, LoRA freezes the pre-trained model weights and injects trainable, low-rank decomposition matrices into the layers of the Transformer architecture. This approach drastically reduces the number of trainable parameters, leading to:

- **Reduced Computational Cost:** Training requires significantly less GPU memory and time, making it feasible on commodity hardware.
- **Modularity:** The small, trained LoRA "adapters" can be stored and swapped out easily, allowing a single base model to be adapted for multiple tasks without storing many copies of the full model.
- **Prevention of Catastrophic Forgetting:** By keeping the original weights frozen, PEFT helps the model retain its general language capabilities while learning the new task.

The empirical results consistently show that even with this efficient tuning method, the student models achieve substantial performance gains. For example, in the Turkish quiz generation task, fine-tuning with LoRA led to sharp increases in both ROUGE scores and the proportion of top-rated ('A') items as judged by humans.

Training Objectives: SFT and Logit-Space Distillation

Let x denote the input (e.g., source text and task spec) and $y = (y_1, \ldots, y_T)$ the target output (e.g., an MCQ with the required schema). The standard *supervised fine-tuning* (SFT) objective maximizes the conditional likelihood under teacher forcing:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) \;=\; -\sum_{(x,y)\in\mathcal{D}} \sum_{t=1}^{T} \log p_\theta(y_t \mid y_{<t}, x)\,. \tag{3.1}$$

When teacher logits are available (or are recomputed in a frozen pass), we can augment SFT with *knowledge distillation* (KD) using temperature scaling $\tau > 0$. Let $q_T^\tau(\cdot \mid z)$ and $p_\theta^\tau(\cdot \mid z)$ be the teacher and student token distributions at temperature τ given context $z = (y_{<t}, x)$. The KD term minimizes the KL divergence between teacher and student *soft* targets:

$$\mathcal{L}_{\mathrm{KD}}(\theta) \;=\; \tau^2 \sum_{(x,y)\in\mathcal{D}} \sum_{t=1}^{T} \mathrm{KL}(q_T^\tau(\cdot \mid y_{<t}, x) \,\|\, p_\theta^\tau(\cdot \mid y_{<t}, x))\,. \tag{3.2}$$

The total loss interpolates SFT, KD, and standard regularization:

$$\mathcal{L}_{\text{total}}(\theta) \;=\; (1-\lambda)\,\mathcal{L}_{\text{SFT}}(\theta) \;+\; \lambda\,\mathcal{L}_{\text{KD}}(\theta) \;+\; \beta\,\|\theta\|_2^2, \quad \lambda \in [0,1]. \tag{3.3}$$

Remark. In our experiments, we primarily use SFT on the synthetic references; KD is an optional extension that can further stabilize learning, especially for stricter format adherence or to reduce exposure bias.

Parameter-Efficient Fine-Tuning (PEFT) via LoRA

To make Stage 2 tractable on commodity GPUs, we adopt **Low-Rank Adaptation (LoRA)**. [120] Consider a pretrained linear projection [121] $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ inside a Transformer block (e.g., W_q, W_k, W_v, W_o in attention; $W_{\text{up}}, W_{\text{down}}$ in the MLP). LoRA *freezes* W and learns a low-rank update ΔW factored as:

$$\Delta W \;=\; BA, \qquad B \in \mathbb{R}^{d_{\text{out}} \times r}, \quad A \in \mathbb{R}^{r \times d_{\text{in}}}, \quad r \ll \min(d_{\text{in}}, d_{\text{out}}). \tag{3.4}$$

The effective weight at training/inference is

$$W' \;=\; W \;+\; \alpha\,\frac{1}{r}\,BA, \tag{3.5}$$

where $\alpha > 0$ is a scaling hyperparameter. Only A, B are trainable; W remains frozen. The number of trainable parameters introduced per linear map is

$$\#\text{params}_{\text{LoRA}}(W) \;=\; r\,(d_{\text{in}} + d_{\text{out}}), \tag{3.6}$$

which is orders of magnitude smaller than $d_{\text{in}} d_{\text{out}}$. If LoRA is applied to m projections per layer (e.g., $m \in \{2, 4, 6\}$ for {Q,V}, {Q,K,V,O}, or {Q,K,V,O,up,down}), and there are L layers with approximately square maps ($d_{\text{in}} \approx d_{\text{out}} = h$), the total trainable parameters scale as

$$\#\text{trainable} \;\approx\; 2\,m\,L\,h\,r, \tag{3.7}$$

which grows *linearly* in the hidden size h, the rank r, and the number of adapted projections m.

Why this helps. (i) *Compute/memory:* Activations dominate memory; training only A, B slashes optimizer state and gradient memory for frozen weights. (ii) *Modularity:* LoRA adapters are small artifacts that can be swapped/merged, enabling one base to serve many tasks. (iii) *Forgetting:* Freezing W preserves general linguistic competence while A, B learn task structure.

Optimization and Practical Configuration

We optimize $\mathcal{L}_{\text{total}}$ with AdamW and a cosine decay schedule with warmup. Gradient accumulation and checkpointing stabilize long sequences; mixed precision reduces memory. We enforce output schemas (e.g., JSON blocks for MCQs/SAQs) at training time via targets and at inference via constrained decoding and validators.

Typical hyperparameters used in this thesis.

- *Ranks/scales.* Turkish QG: LoRA rank r=16, α=32 on Llama-2 (7B/13B), 3 epochs, LR 1×10^{-4}; GPT-3.5 was fine-tuned via API.[2]
- *Persian MCQ.* LoRA r=16, α=32; cosine LR 1×10^{-4}; weight decay 1×10^{-4}; seq. len. $\approx$ 3500; 3 epochs; training on dual A6000 GPUs with DeepSpeed/FlashAttention.[3]
- *Syntax feedback.* LoRA r=32, α=64; LR 3×10^{-4}; total batch size 16; cosine schedule with warmup 0.1; dual A6000 with DeepSpeed and FlashAttention 2.[4]

Acknowledging the Semantic Gap

A key limitation of distillation, which this framework inherits, is the potential for a "semantic gap". The student model becomes highly proficient at mimicking the surface structure, format, and lexical choices present in the teacher's synthetic data. However, it may not fully acquire the deeper reasoning or nuanced judgment of the much larger teacher model. This means its ability to generalize to out-of-distribution examples or handle complex edge cases might be more limited. This is particularly relevant for tasks requiring deep understanding, such as generating feedback for sophisticated writing errors or crafting questions that test multi-hop inference.

3.3 Conclusion: A Replicable Blueprint for Educational AI

The "From Language to Learning" framework provides a robust and validated methodological blueprint for addressing the critical challenge of scalable, multilingual educational content generation. Its two-stage distillation architecture systematically converts the general competence of large LLMs into the specialized expertise of smaller, open-source models, guided by principles of scalability, pedagogical fidelity, and equity. By deconstructing the framework into its architectural stages and functional components, this chapter has illuminated the design choices, trade-offs, and practical considerations that underpin its success across diverse tasks and languages.

While acknowledging its limitations—such as its dependence on the teacher model and the overhead of prompt engineering—the framework's core contribution is a practical, replicable pathway that democratizes the development of AI-powered educational tools. It shifts the paradigm from ad-hoc prompting of black-box models to a structured, transparent, and pedagogy-driven process of model specialization. The empirical evidence presented throughout this thesis confirms that this approach yields reliable, high-quality, and specialized tools that significantly outperform their general-purpose counterparts. Ultimately, the "From Language to Learning" framework offers not just a set of techniques, but a vision for a future where advanced AI can be safely and effectively harnessed to make personalized, high-quality education accessible to all learners, in any language.

[2] See Section 8.2.3 for full details.
[3] See Section 9.2.3.
[4] See Section 10.2.3.

Part III

Multilingual Crossword Generation

Chapter 4

A Unified Framework for English Educational Crossword Generation

4.1 Motivation and Scope

Crossword puzzles, when reoriented from entertainment to pedagogy, offer an engaging and effective medium for reinforcing knowledge across a multitude of academic domains, including science, geography, history, and vocabulary [38, 40, 39]. Unlike their traditional counterparts, which often rely on cryptic wordplay and thematic ambiguity, *educational* crosswords are characterized by a preference for clarity, factual content, and topical grounding, enabling learners to form direct connections between clues and curricular materials. This pedagogical format encourages independent learning, strengthens memory recall by compelling students to access previously learned information, and cultivates critical thinking and problem-solving skills through a low-stakes, interactive challenge that provides immediate feedback.

Despite their established educational value, the widespread adoption of high-quality crossword puzzles in learning environments has been historically constrained by the significant labor and specialized expertise required for their creation. Furthermore, existing computational resources and datasets are overwhelmingly tailored to traditional, entertainment-focused crosswords. These conventional corpora of clue-answer pairs, while extensive, lack a crucial element for educational applications: a direct link between a clue and a specific source text from which its answer can be verifiably inferred. This textual grounding is essential for two primary reasons. First, it ensures that the generated content is aligned with a specific curriculum, allowing educators to create puzzles that directly support their lesson plans. Second, it provides a vital mechanism for steering Large Language Models (LLMs) to generate factually accurate, contextually relevant clues while significantly mitigating the risk of model hallucination—a critical requirement for any tool intended for educational use.

This chapter introduces a novel, unified, and end-to-end framework designed to overcome these limitations by automating the generation of English educational crosswords. The framework's principal innovation is a high-fidelity, context-grounded clue generator powered by instruction-tuned LLMs. This advanced generator is made possible by a purpose-built educational dataset, which we term `clue-instruct`, created specifically for this task. The instruction-tuned model is integrated into a modular and robust pipeline that includes components for data processing, clue validation, and final grid construction. Together, these elements constitute a practical, scalable, and reproducible recipe for transforming unstructured educational texts into polished, classroom-ready crossword

puzzles.

4.2 Framework Overview

The system is architected as a modular pipeline that accommodates two primary operational modes, catering to different pedagogical scenarios and available inputs. This dual-path design, illustrated in Figure 4.1, ensures both flexibility for educators and robustness in content generation.

- **Path (a): Text-Conditioned Generation.** This is the primary and most sophisticated mode of operation, designed for instances where a source text—such as a textbook chapter, an academic article, or a set of lecture notes—is available. In this path, the framework first performs automated keyword extraction to identify salient terms and concepts within the text. Subsequently, it employs a specialized, instruction-tuned LLM to generate clues that are factually grounded in the provided context. This path is the preferred method for creating deeply curriculum-aligned puzzles that reinforce specific learning materials.
- **Path (b): Answer-Only Generation.** This mode is designed for scenarios where an educator provides a pre-defined list of answers or keywords without an accompanying source text. This is common for vocabulary reviews, end-of-unit summaries, or topic-based activities. In this path, the framework utilizes LLMs that have been fine-tuned on a vast historical corpus of clue-answer pairs to generate relevant and well-formed clues based solely on the input answers.

Crucially, the outputs from both generation paths converge into a unified, two-stage validation and filtering process. This shared quality assurance mechanism ensures that all candidate clue-answer pairs, regardless of their origin, meet stringent pedagogical standards. The validated pairs are then passed to a final schema generator, which algorithmically assembles them into a compact and coherent crossword grid. The system also provides an optional interface for users to review and manually select their preferred clue-answer pairs before the final grid is constructed, allowing for a human-in-the-loop approach to puzzle creation.

4.3 Foundational Data Resources

The development and efficacy of the framework are critically dependent on two large-scale, complementary data resources. One is a massive historical corpus that captures the breadth of conventional crossword styles, and the other is a novel, purpose-built dataset designed specifically for the nuances of educational, text-grounded clue generation.

4.3.1 Century-Scale Clue–Answer Corpus (1913–2021)

To support the answer-only generation path and to train the system's validation components, a comprehensive database of historical crosswords was constructed. The data

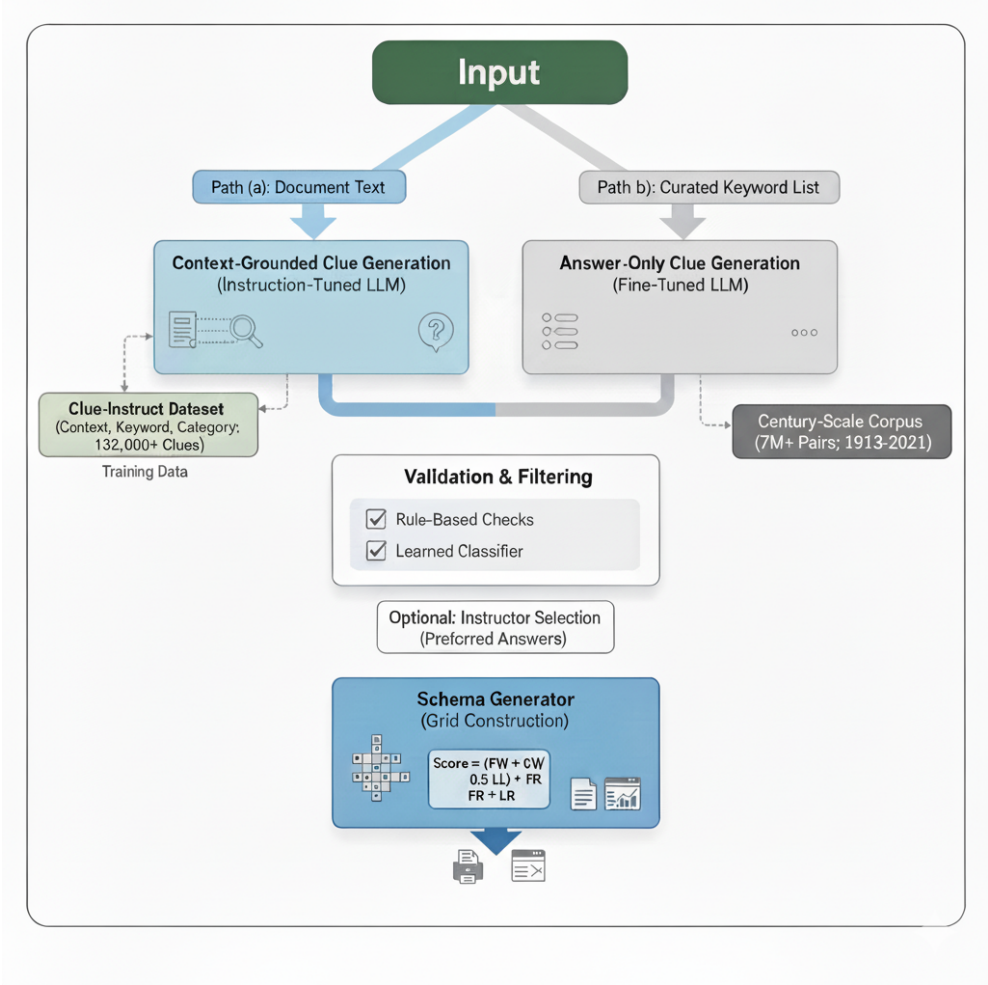


Figure 4.1: System architecture: The framework supports two generation paths that converge on shared validation and schema generation components. (a) Text-conditioned generation for curriculum-aligned puzzles. (b) Answer-only generation for keyword-based reviews.

collection process involved a combination of manual extraction from physical print sources and automated web scraping of online crossword archives and databases. The resulting corpus is extensive, comprising **7,327,448** clue-answer pairs and covering a temporal span from 1913 to mid-2021. This vast collection includes approximately **300,000** unique answers and showcases a wide diversity of clue styles, such as fill-in-the-blank, conditional, and synonymic definitions.

A meticulous preprocessing pipeline was applied to the raw data to ensure its quality and utility. This involved de-duplication of identical pairs, the removal of cross-referenced

clues (i.e., clues that refer to other entries within the same puzzle), and the filtering of outliers. The cleaned corpus serves two critical functions within the framework:

1. It provides the large-scale, supervised training data required to fine-tune the generative models for the **Answer-Only path (b)**.

2. It is used to train the **learned validator**, a key component of the quality assurance pipeline that is applied to outputs from both generation paths.

Figure 4.2 visualizes the distribution of entries by answer length, providing insights into common word lengths in English crosswords for the Century-Scale Clue–Answer Corpus.

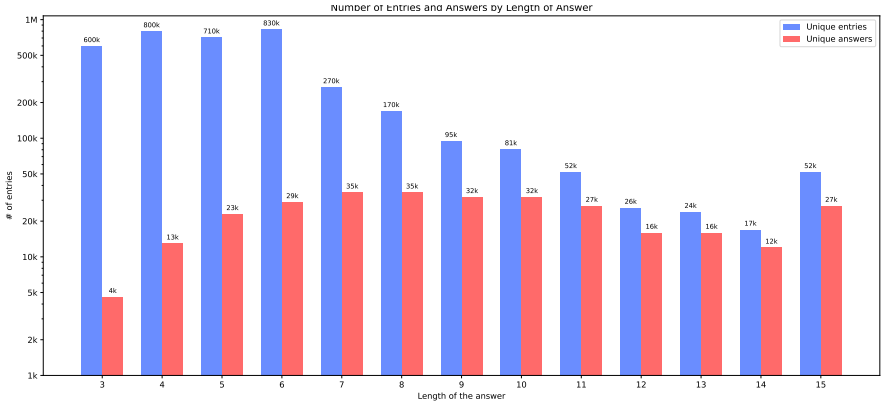


Figure 4.2: Distribution of database entries by answer length, showing unique clue–answer pairs (blue) and unique answers (red).

4.3.2 Clue-Instruct: A Dataset for Context-Grounded Instruction Tuning

The most significant limitation of existing resources is the absence of datasets for educational, text-grounded clues. To address this gap and to power the high-fidelity **Text-Conditioned path (a)**, we developed `clue-instruct`, a novel dataset specifically designed for this purpose. To our knowledge, it is the first publicly available corpus that systematically associates crossword clues with both a corresponding answer and a source context from which that answer can be inferred.

The dataset was constructed using a semi-automated pipeline that begins with information extraction from English Wikipedia. The initial sections of articles, which typically contain concise definitions and key facts, were mined for relevant content. A rigorous screening process was applied to filter this content based on page metadata (e.g., retaining pages with high view counts or a "Top" importance rating), context length (keeping texts between 30 and 1000 words), and keyword constraints (e.g., keywords must be between 3 and 20 characters and contain no more than three words).

For each screened ‘(context, keyword, category)‘ triplet, a powerful LLM (GPT-3.5-Turbo) [122] was prompted to generate three distinct, educational-style clues. The final

`clue-instruct` dataset contains **44,075** unique examples, totaling **132,225** context-grounded clues across 20 broad categories. As shown in Figure 4.3, the dataset has strong coverage in academic subjects like 'Geography', 'Science', and 'Applied Science'. This dataset is the cornerstone resource that enables the instruction-tuning of LLMs to perform the nuanced task of educational clue generation with high fidelity.

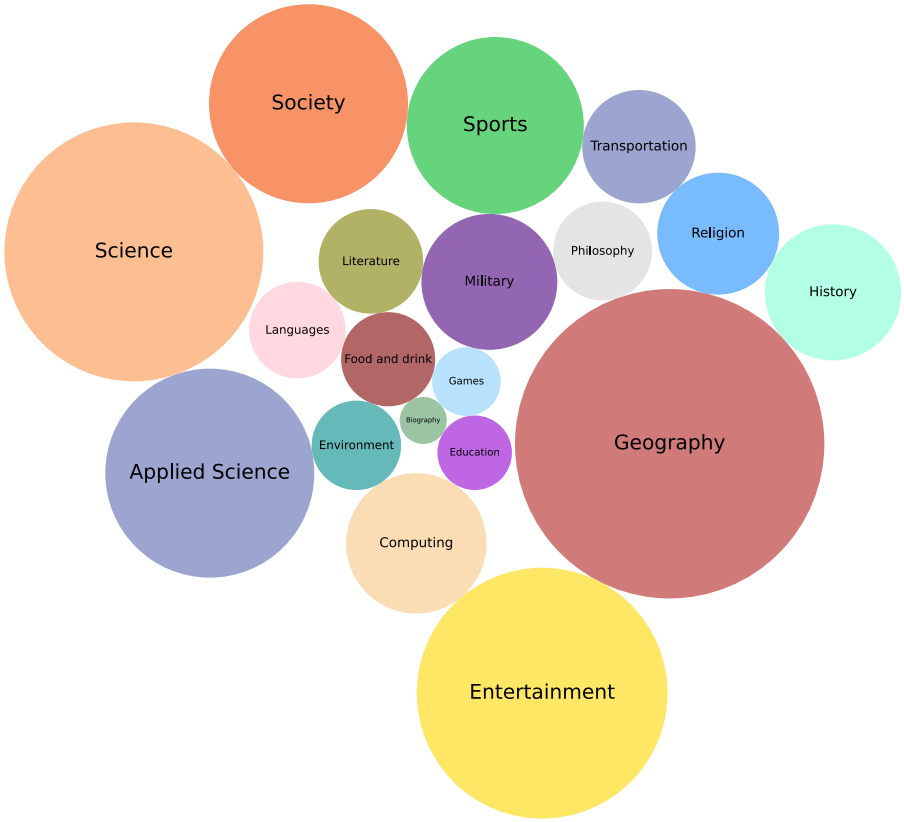


Figure 4.3: Category distribution for the 44,075 examples in `clue-instruct`, showing a dominance of academic subjects.

4.4 Methodology: From Text to Puzzle

The framework's methodology is divided into three sequential stages: (1) Clue Generation, which operates via one of the two distinct paths; (2) Validation and Filtering, a shared quality control gate; and (3) Schema Generation, the final assembly of the puzzle.

4.4.1 Clue Generation

Path (a): Context-Grounded Generation via Instruction Tuning

For the text-conditioned path, the goal is to produce clues that are faithfully entailed by a source text. The original implementation of this path relied on zero/few-shot prompting of general-purpose LLMs. While functional, this approach can suffer from inconsistent output quality and reliability. To create a more robust and professional-grade generator, we developed an advanced methodology centered on instruction-tuning specialized models, enabled by the `clue-instruct` dataset.

Data Preparation and Prompting. The process begins with the data construction pipeline developed for `clue-instruct`, as shown in Figure 4.5. An input text is segmented, and salient keywords are automatically extracted. For each keyword, a structured prompt is dynamically created. This prompt, depicted in Figure 4.4, provides the model with the keyword, its associated category, and the source context paragraph. It contains explicit instructions for the model to find relevant facts in the context and generate three short, clever clues that do not contain the keyword itself, formatted as a JSON object.

```
You are a crosswords expert.

Generate short and clever definitions for crosswords, based on a given keyword, a category and a
keyword-related context, following the instructions provided below.

        KEYWORD: {keyword}

        CATEGORY: {category}

        CONTEXT: {text}

Follow these steps:

1. Find parts of the given context related to the {keyword} and {category}.

2. Select three key pieces of information related to {keyword} and {category} that are present in
the context.

3. Create short clues from these key facts, making sure not to include the keyword in the clues.

4. Put these clues into a JSON file under the key: 'clues'.
```

Figure 4.4: The structured prompt used to guide the LLM in generating educational clues for the `clue-instruct` dataset .

Validation of the `clue-instruct` Dataset. To assess the quality of the synthetically generated dataset, we conducted a rigorous human evaluation on a sample of its contents . A subset of 600 examples, corresponding to 1,800 individual clues, was annotated by human evaluators according to a five-level rating scale . The rating guidelines were defined as follows:

- **RATING-A:** The clue is valid, well-formed, and coherent with the given context, answer, and category .
- **RATING-B:** An acceptable clue with minor imperfections, such as a loose correlation with the specified category .
- **RATING-C:** The clue is relevant to the answer but is too generic or correlates only loosely with the source context .
- **RATING-D:** The clue is incorrect or irrelevant with respect to the answer or context.

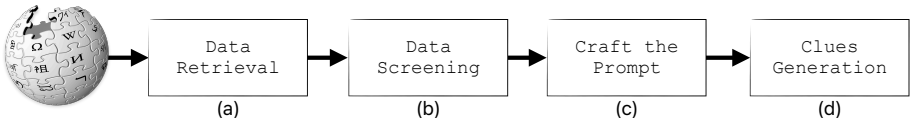


Figure 4.5: The data construction pipeline for `clue-instruct`, which forms the basis of the text-conditioned generation path: (a) information extraction from Wikipedia, (b) data screening and filtering, (c) prompt design, and (d) clue generation using a teacher model (GPT-3.5-Turbo).

- **RATING-E:** The clue is unacceptable because it contains the answer or a direct variant of it.

The results of this evaluation, shown in Figure 4.6, confirmed the high quality of the dataset. Over 72% of the clues received the highest score (RATING-A), and this figure rises to 81% when considering clues rated A or B as acceptable. This validation was critical in confirming that `clue-instruct` is a high-quality resource suitable for supervised fine-tuning. Table 4.1 provides illustrative examples of generated clues and their corresponding human-assigned ratings.

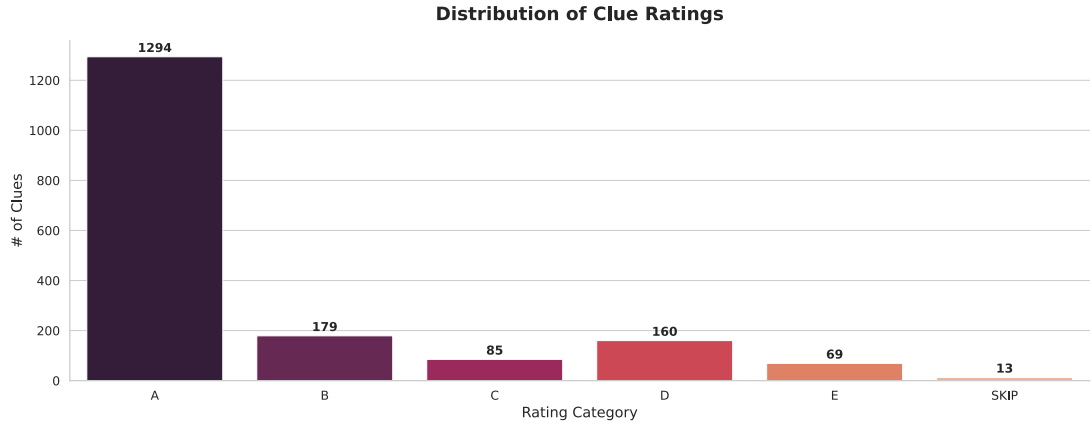


Figure 4.6: Distribution of human-assigned ratings on a sample of 1,800 clues from the `clue-instruct` dataset, demonstrating high overall quality.

Improving Reliability with Instruction Tuning. To move beyond the limitations of zero-shot or few-shot prompting, we leverage the `clue-instruct` dataset to fine-tune a range of open-source, instruction-following LLMs, including variants of LLaMA2-chat (7B, 13B) [123] and MPT-instruct (7B, 30B) [124]. The fine-tuning process is performed efficiently using Low-Rank Adaptation (LoRA) [121] , with a rank of $r = 16$ and alpha of $\alpha = 32$, over two epochs. This instruction-tuning process specializes the models for the specific task of educational clue generation. As our experimental results demonstrate, this

Table 4.1: Examples of generated clues from the `clue-instruct` dataset with their assigned human ratings, illustrating the quality criteria.

Answer	Category	Clue	Rating
Robocall	Society	May be blocked by phone companies to prevent scams	A
Ministry Of Magic	Literature	Corrupt and incompetent government in J.K. Rowling's Wizarding World	A
Lovesick	Literature	Renewed for a third season, released exclusively on Netflix	C
South American tapir	Science	One of the four recognized species in the tapir family	E

alignment dramatically improves their ability to consistently adhere to the required JSON output format, generate clues of higher quality, and ensure strong factual grounding in the source text, significantly outperforming their off-the-shelf counterparts.

Path (b): Answer-Only Generation

In the absence of a source text, the framework's second path generates clues based solely on a provided list of answers. To enable this functionality, we fine-tuned a suite of generative models with varying parameter counts—specifically, GPT3-DaVinci (175B), GPT3-Curie (13B), and GPT2-XL (1.5B)—on a large subset of the historical crossword corpus described in Section 4.3.1. This training process exposes the models to the vast diversity of lexical patterns and stylistic conventions found in human-authored crosswords, enabling them to produce plausible and well-formed clues even without contextual grounding.

4.4.2 Shared Component: Validation and Filtering

A critical step in ensuring the pedagogical utility of the generated content is a rigorous validation process. To this end, outputs from both generation paths are passed through a shared, two-stage validation filter designed to maximize precision and eliminate low-quality pairs.

1. **Rule-Based Checks:** A set of deterministic heuristics provides a first-pass filter for common and easily identifiable errors. These rules include rejecting answers composed of more than three words, filtering out clues that explicitly contain the answer string, and checking for basic grammatical and length constraints.

2. **Learned Classifier:** To address more nuanced quality issues, we developed a supervised binary classifier. This validator was trained on a human-labeled dataset of 6,000 clue-answer pairs (evenly balanced between acceptable and unacceptable examples) drawn from the historical corpus. We trained several models for this

classification task, including various sizes of GPT-3 (Davinci, Curie, Babbage, Ada) and BERT-base, to serve as an effective quality gate.

This cascaded validation approach significantly reduces the number of flawed pairs, thereby minimizing the need for manual curation by educators.

4.4.3 Shared Component: Schema (Grid) Generator

The final stage of the framework, schema generation, is handled by an algorithmic layout engine. This engine iteratively places validated answer-clue pairs into a grid, aiming to maximize several quality metrics to produce a compact and highly interlocked crossword puzzle suitable for educational use. The process is guided by a composite scoring function:

$$\text{Score} \;=\; (\text{FW} + 0.5 \cdot \text{LL}) \times \text{FR} \times \text{LR},$$

where:

- **FW (Filled Words):** The total number of words successfully placed in the grid.
- **LL (Crossing Letters):** The count of letters that serve as intersections between horizontal (Across) and vertical (Down) words. This metric is weighted to emphasize interlocking.
- **FR (Fill Ratio):** The proportion of the grid cells occupied by letters, indicating the density of the puzzle.
- **LR (Intersection Density):** A measure of how frequently words intersect, contributing to the puzzle's compactness and challenge.

The algorithm employs a randomized constructive search. Stopping conditions for grid assembly include reaching a target fill ratio, capping the number of relocations or resets during the search, or exhausting a predefined time budget. Additionally, educators may specify "must-include" answers, allowing the system to prioritize placing key vocabulary terms to align with specific learning objectives. This scoring-based approach ensures the creation of aesthetically pleasing, solvable, and pedagogically effective crossword puzzles.

4.5 Experimental Validation

We conducted a comprehensive set of experiments to rigorously evaluate each component of the framework and demonstrate its overall effectiveness.

4.5.1 Validation of Clue Generation Models

Path (a): Instruction-Tuned Models. The performance of the instruction-tuned models for text-conditioned generation was assessed using both automatic metrics and human evaluation. As shown in Table 4.2, fine-tuning on the `clue-instruct` dataset results in a dramatic improvement in text adherence, as measured by ROUGE-L [125] scores

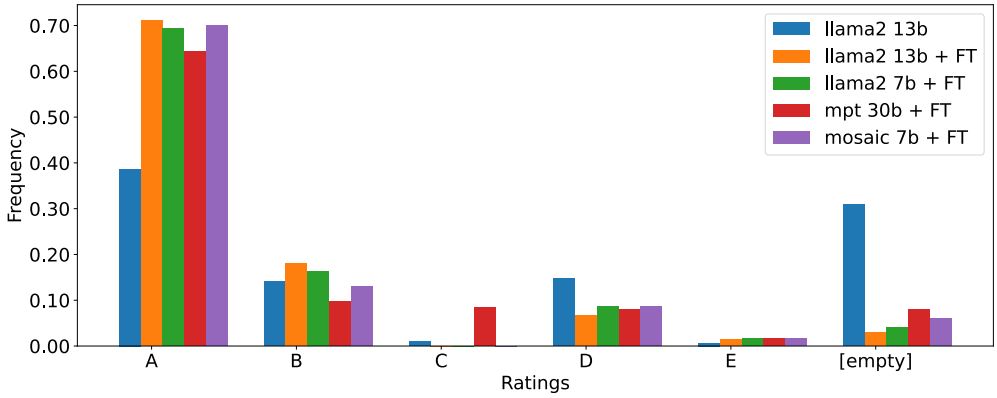


Figure 4.7: Human evaluation across LLMs; "+ FT" indicates fine-tuned on `clue-instruct`. Fine-tuning reduces empty/malformed outputs and improves A-rates.

against a strong oracle. For instance, the LLaMA2-13B model's ROUGE-L score jumped from 25.27 to 55.40 after fine-tuning. The fine-tuned LLaMA2-13B model emerged as the top performer overall.

Human evaluations corroborate these quantitative findings. As illustrated in Figure 4.7, the fine-tuned models produce a significantly higher proportion of A-rated (highest quality) clues compared to their base versions, while drastically reducing the frequency of malformed or empty outputs. The base LLaMA2-13B model, for instance, produced empty or low-quality (D-rated) outputs for nearly 50% of cases, a failure rate that drops to around 15% after fine-tuning. Furthermore, our experiments revealed that the instruction-tuning process is highly sample-efficient: as shown in Figure 4.8, using just 10% of the training data is sufficient to achieve a substantial performance leap, aligning the models to the task effectively.

Table 4.2: Off-the-shelf vs. fine-tuned LLMs on `clue-instruct`. Fine-tuning markedly improves adherence and format reliability; LLaMA2-13B performs best overall.

Model type	Model name	#Params	ROUGE-1	ROUGE-2	ROUGE-L
Off-the-shelf	MPT-instruct	7B	23.98	11.79	19.69
	LLaMA2-chat	13B	**31.80**	**15.32**	**25.27**
	MPT-instruct	30B	29.92	14.47	24.30
Fine-tuned	LLaMA2-chat	7B	59.92	40.98	52.28
	MPT-instruct	7B	59.26	40.37	51.68
	LLaMA2-chat	13B	**62.97**	**44.97**	**55.40**
	MPT-instruct	30B	61.42	42.63	53.77

Path (b): Answer-Only Models. The performance of the answer-only generators was evaluated on their ability to produce acceptable clues for a set of 4,000 keywords, as judged by human evaluators. As shown in Table 4.3, the model performance correlates

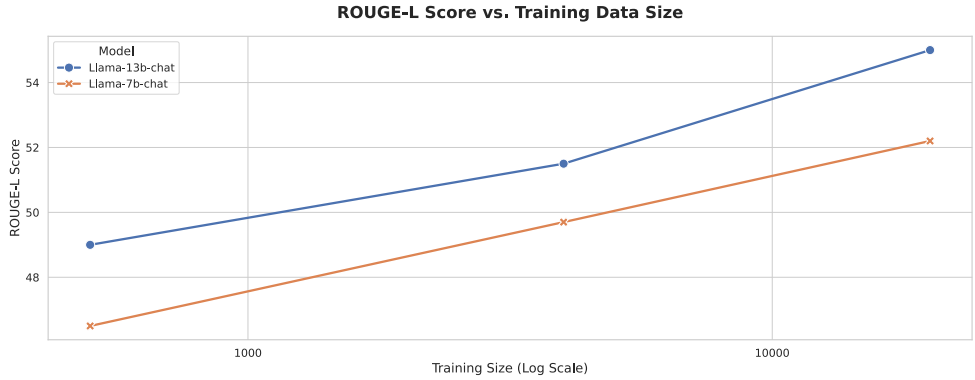


Figure 4.8: Impact of training data size (1%, 10%, 100%) on ROUGE-L for LLaMA2 models. A small amount of data yields a large performance gain.

with parameter count, with the largest model, GPT3-DaVinci, achieving the highest acceptability rate of **71.63%**.

Table 4.3: Acceptable clue rates for answer-only generators, as determined by human evaluation.

Model	*% acceptable clues*
GPT3-DaVinci	71.63
GPT3-Curie	60.18
GPT2-XL	38.65

4.5.2 Validation of the Filtering Component

The learned validator's effectiveness in distinguishing between acceptable and unacceptable clue-answer pairs was tested on a held-out set of 20% of the 6,000 labeled examples. As shown in Table 4.4, the GPT3-DaVinci-based classifier achieved the highest overall performance, with an accuracy of **76.8%** and an F1-score of **0.79**, confirming its capability as a reliable filter to enhance the quality of the candidate clue pool.

Table 4.4: Performance of the learned validator in distinguishing acceptable vs. unacceptable pairs.

Model	*Accuracy*	*Precision*	*Recall*	*F1*
GPT3-DaVinci	76.8%	0.73	0.87	0.79
GPT3-Curie	75.5%	0.75	0.78	0.76
GPT3-Babbage	72.6%	0.71	0.79	0.75
GPT3-Ada	70.5%	0.70	0.73	0.72
BERT-base	59.7%	0.59	0.67	0.63

4.5.3 End-to-End Pipeline Performance

When all components of the text-conditioned pipeline (Path a) are integrated, the framework demonstrates strong end-to-end performance. In an evaluation conducted on 500 paragraphs from Wikipedia, the system achieved an overall success rate of **79.73%** in producing validated, human-judged acceptable clue-answer pairs. Figure 4.9 provides a concrete visualization of this process, tracing the transformation from an input paragraph on atomic structure to a final, validated set of clues and answers. These high-quality pairs are then seamlessly assembled by the schema generator into a polished and pedagogically valuable educational puzzle, as exemplified in Figure 4.10.

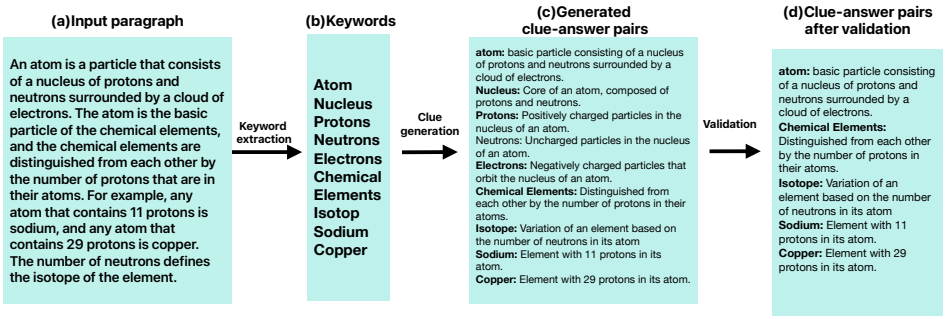
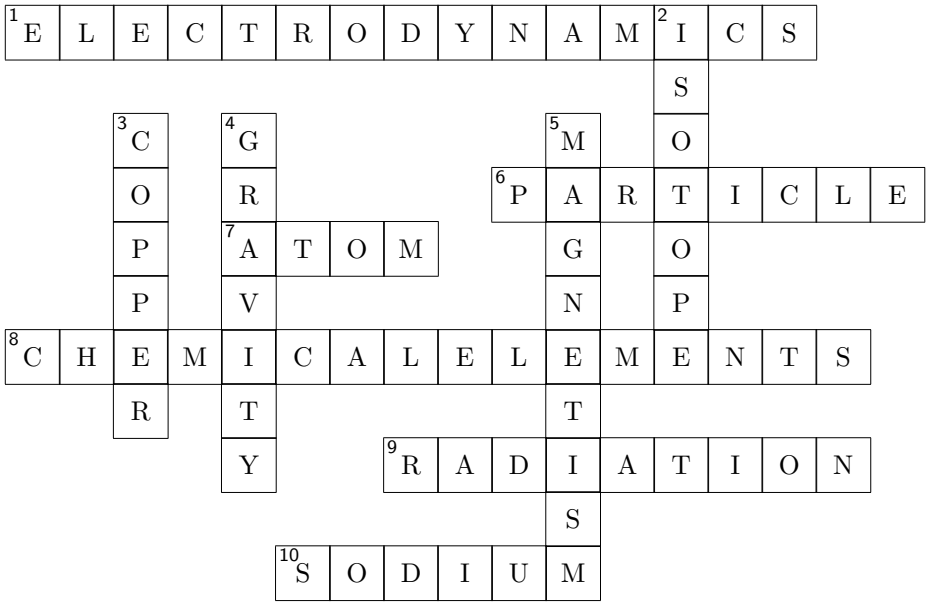


Figure 4.9: An example of the end-to-end text-conditioned pipeline: from input paragraph (a), to keyword extraction (b), to initial clue generation (c), and finally to the set of validated clue–answer pairs (d).

4.6 Conclusions

This chapter introduced a unified and robust framework for the automated generation of English educational crosswords. The system effectively integrates a modular, dual-path pipeline with a state-of-the-art, instruction-tuned clue generator. The framework's central innovation lies in the creation and application of the `clue-instruct` dataset, a novel resource that enables the specialization of Large Language Models to produce concise, factual, and *text-entailed* clues with unprecedented reliability and quality. The system's capabilities are further enhanced by a flexible answer-only generation path and a rigorous validation cascade trained on a vast historical corpus.

When combined with an efficient schema generator, these components provide a practical, scalable, and powerful pathway from unstructured lesson texts to polished, classroom-ready educational tools. The empirical results, validated through both automatic metrics and extensive human evaluation, confirm the effectiveness of each component and the success of the integrated system. By addressing the critical need for high-quality, automated pedagogical content, this work represents a significant step forward in leveraging advanced AI to create engaging and effective learning experiences.



Across 1 ⋆ Study of the interaction between electricity and magnetism 6 ⋆ Fundamental unit of matter 7 basic particle consisting of a nucleus of protons and neutrons surrounded by a cloud of electrons 8 Distinguished from each other by the number protons in their atoms 9 ⋆ X-ray 10 Element with 11 protons in its atom **Down** 2 Variation of an element based on the number of neutrons in its atom 3 Element with 29 protons in its atom 4 ⋆ Falling force 5 ⋆ Attraction of iron for steel

Figure 4.10: An example of a final educational crossword produced by the schema generator, using validated pairs from both generation paths.

Chapter 5

A Unified Framework for Italian Educational Crossword Generation

5.1 Motivation and Scope

Educational crossword puzzles serve as a highly effective pedagogical tool, fostering an interactive learning environment that enhances vocabulary acquisition, spelling proficiency, and the retention of technical terminology across diverse disciplines. In the context of the Italian language, these puzzles are particularly valuable for reinforcing grammatical structures and specialized lexicons in subjects ranging from history and geography to the arts and sciences. By requiring students to engage in critical thinking to match clues with appropriate answers, educational crosswords boost memory recall and improve problem-solving abilities, transforming the learning process into an engaging and motivating experience.

Despite these benefits, the creation of high-quality educational crosswords in Italian presents significant challenges. The process is traditionally labor-intensive, demanding not only subject-matter expertise but also linguistic creativity to craft clues that are concise, unambiguous, and grammatically precise. This manual authoring bottleneck has limited the widespread availability of customized, curriculum-aligned puzzles for Italian educators. Furthermore, the advent of Large Language Models (LLMs) has been hampered by a lack of specialized resources; existing digital corpora are typically designed for entertainment-style crosswords and do not provide the essential link between a clue and a source text, which is paramount for educational validity.

To address this critical gap, this chapter presents a novel and comprehensive framework for the automated generation of Italian educational crosswords. This system is distinguished by its ability to produce contextually grounded and stylistically controlled clues, leveraging a purpose-built dataset named **Italian-Clue-Instruct**. Our approach integrates state-of-the-art LLMs, including GPT-4o, Mistral-7B, and Llama3-8B, [126] into a modular pipeline that encompasses data preparation, style-specific clue generation, rigorous validation, and automated grid construction. By introducing fine-tuned, open-source models specifically for this task, this work moves beyond the earlier reliance on few-shot learning with general-purpose models, offering a more robust and professional solution for creating engaging, interactive, and pedagogically sound learning tools for the Italian language.

5.2 Framework Overview

The framework is architected as a coherent, end-to-end system that offers two distinct operational pathways, providing educators with the flexibility to generate puzzles from different types of source material. This dual-path design, first conceptualized in our preliminary work and illustrated in Figure 5.1, ensures the system can be adapted to a variety of classroom needs.

- **Path (a): Text-Conditioned Generation.** This is the primary and most advanced mode, designed for creating deeply curriculum-aligned puzzles from a given Italian text. In this path, the system first extracts salient keywords from the source material and then employs specialized, instruction-tuned LLMs to generate clues that are factually grounded in the provided context. A key innovation of this path is its ability to generate clues in four distinct syntactic styles, allowing for unprecedented control over the puzzle's linguistic complexity.
- **Path (b): Answer-Only Generation.** This path caters to scenarios where an instructor provides a list of keywords or vocabulary terms without an accompanying source text. This is a common requirement for creating topic reviews or vocabulary reinforcement exercises. Here, the framework utilizes LLMs fine-tuned on a large legacy corpus of Italian clue-answer pairs to generate relevant, definition-style clues.

A cornerstone of the framework's design is that both generation paths feed into a common, multi-stage validation pipeline. This shared quality assurance process, which combines automated checks with learned classifiers, ensures that all clue-answer pairs meet high pedagogical and linguistic standards before they are used. The final, validated pairs are then passed to a schema generator, which algorithmically assembles the puzzle grid. This modular architecture, which abstracts functionality into distinct roles (e.g., generator, validator, layout engine), allows for straightforward updates and makes the framework easily adaptable.

5.3 Foundational Data Resources for Italian

The framework's capabilities are built upon two substantial and complementary Italian-language datasets. The first is a large-scale collection of traditional clue-answer pairs, while the second is a novel, text-aligned corpus created specifically for educational clue generation.

5.3.1 Legacy Clue-Answer Collection

To power the answer-only generation path and to train the system's learned validators, a comprehensive dataset of Italian crossword clues was compiled. The data was sourced from a variety of online and offline resources, including scraping Italian crossword websites like ‘dizy.com‘ and ‘cruciverba.it‘, and processing PDF versions of renowned Italian puzzle publications such as ‘La Settimana Enigmistica‘. After a thorough cleaning process to

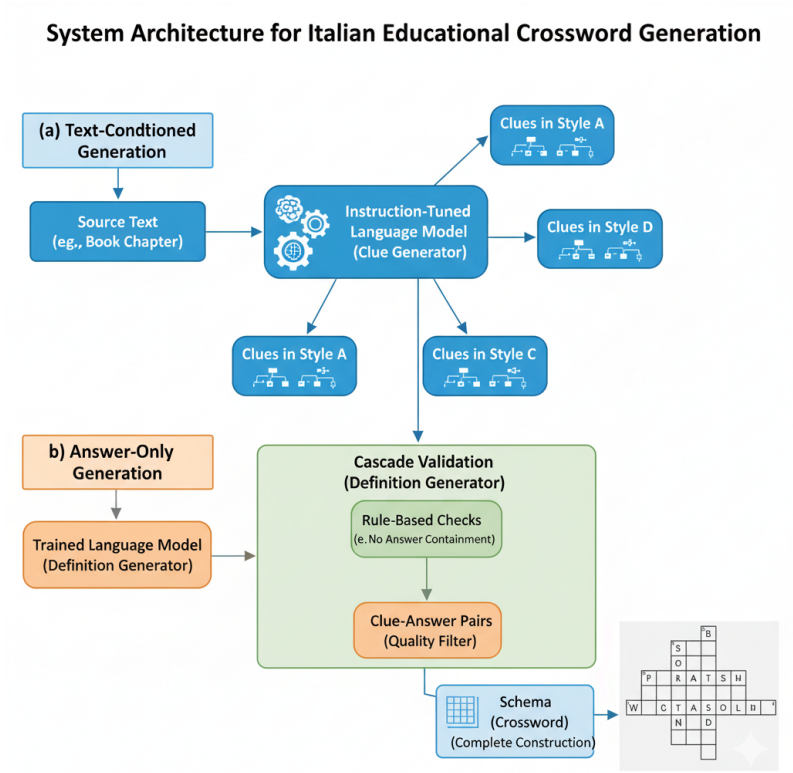


Figure 5.1: Overall system architecture for Italian crossword generation, illustrating the two main paths for clue creation: (a) from a source text and (b) from a list of keywords.

merge sources and remove duplicates, the final dataset comprises **125,600** unique Italian clue-answer pairs.
This corpus captures a wide range of domains, including history, geography, literature, and pop culture, and exhibits diverse linguistic and structural patterns. As shown in Figure 5.2, the distribution of clues is skewed towards shorter answers, which are more common in Italian crosswords. This dataset is a critical resource, providing the necessary supervision for training models to understand the stylistic conventions of Italian crossword clues.

5.3.2 Italian-Clue-Instruct: A Text-Aligned Educational Corpus

The central limitation of existing resources is their unsuitability for educational, text-grounded tasks. To overcome this, we created **Italian-Clue-Instruct**, a novel dataset that, for the first time in Italian, systematically links crossword clues to a source text.

Data Collection and Curation. The dataset was built using a meticulous data collection pipeline, as illustrated in Figure 5.7. The process began by extracting the introduct-

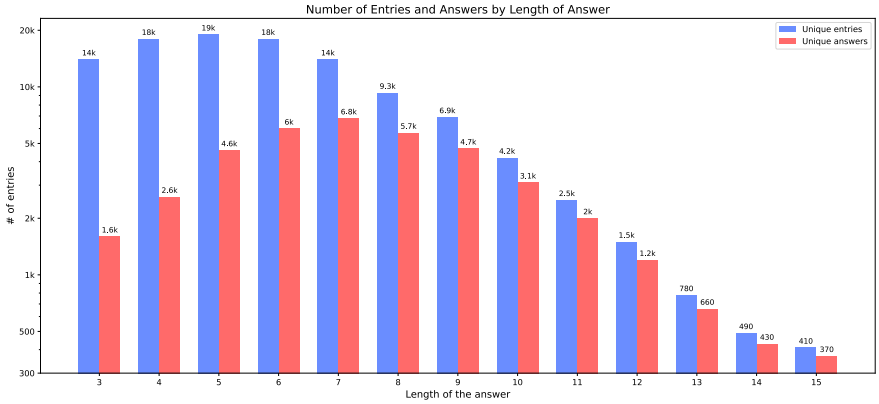


Figure 5.2: Distribution of the 125,600 entries in the legacy Italian clue-answer database by answer length. Shorter answers have a higher frequency of associated clues.

ory sections of 88,403 articles from the Italian Wikipedia. A multi-stage filtering process was then applied to ensure data quality. We prioritized articles based on popularity and relevance, discarded texts shorter than 50 words, and constrained keywords to be between 3 and 20 characters in length and contain no more than two words. This process yielded a refined set of 11,413 high-quality articles, we selected randomly 5,000 articles for clue generation.

Synthetic Clue Generation. From this curated set, we used a state-of-the-art LLM (GPT-4o) as a "teacher model" to generate a variety of clues for each article. Guided by a series of four highly structured prompts - illustrated in Figures 5.4, 5.5, 5.6, and 5.3 — To control for prompt-induced sampling bias, we assigned one of the four prompt templates to each article uniformly at random, yielding a perfectly balanced split across prompt types (5,000/4 entries per template, i.e., 1,250 each), the model produced a minimum of three distinct clues per article, resulting in a final dataset containing over **30,000 entries** across 29 thematic categories. The four prompt families differ in their constraints on the clue's surface form and the generation procedure: (i) a *bare noun phrase* with zero determiner, headed by a common or proper noun and optionally followed by a relative clause or other complements; (ii) a *definite determiner phrase* that begins with the definite article, followed by a noun (possibly with adjectives) and optional modifiers; (iii) a *copular schema* with an elided subject of the form "è . . . "; and (iv) a *general-purpose, task-agnostic* prompt that operates via an 8-step pipeline while enforcing the absence of the keyword in the clues. The dataset statistics, shown in Figures 5.8 and 5.9, reveal a broad topical coverage, with a strong presence in "Entertainment," "Geography," and "History." This rich, text-aligned corpus is the foundational resource that enables the instruction-tuning of smaller, open-source models for the high-fidelity, text-conditioned generation path.

```
You are a crossword expert.
Generate concise and clever clues in Italian for educational crossword puzzles based on a specified Keyword and its relation to an
assigned Text. To execute this task properly, replicate the guidelines below:

KEYWORD: {keyword}
TEXT: {text}

Observe the following steps:
1. Substitute every pronoun in the text with full phrases expressing their referents.
2. Split the text into small independent sentences that could be understood out of context.
3. Pinpoint three concise sentences that contain the Keyword and best characterize the keyword. Try to select sentences from
different parts of the Text.
4. Generate short and clever crossword clues in Italian from the selected sentences. Make sure that  the keyword remains absent
from the clues. If the Keyword is not the subject of the sentence, make sure that it is substituted with an appropriate clitic,
possessive or demonstrative pronoun. Generate clues from all the parts of the text and use all of the information provided to
generate the clues.
5. Ensure that each clue functions as a description or definition of the keyword rather than a query, focusing on details about
the keyword.
6. Make sure that each clue's information can be traced back to the text. Make sure that the clues are relevant and that they are
sufficient to identify the keyword. Make sure that the keyword does not appear in the clues. Make sure that any part of the
keyword is not present in the clues.
7. Select only the three best clues for educational purposes.
8. Compile these clues into a list formatted as follows: [clue1, clue2, clue3]  into a JSON file under the key: 'clues'. Make sure
the output is in the requested format and do not include the whole process in the output, but only the clues.
```

Figure 5.3: **Prompt template: General-purpose (task-agnostic).** A pipeline that expands coreference, segments sentences, selects key sentences containing the keyword, rewrites them into short Italian clues that exclude the keyword, and outputs `[clue1, clue2, clue3]` as JSON under key `clues`.

```
You are a crossword expert.
Generate concise and clever clues in Italian for educational crossword puzzles based on a specified Keyword and its relation to an
assigned Text. To execute this task properly, replicate the guidelines below:
KEYWORD: {keyword}
TEXT: {text}

Observe the following steps:
1. Substitute every pronoun in the text with full phrases expressing their referents.
2. Split the text into small independent sentences that could be understood out of context.
3. Pinpoint three concise sentences that contain the Keyword and best characterize the keyword. Try to select sentences from
different parts of the Text.
4. Generate short and clever crossword clues in Italian from the selected sentences. Make sure that the keyword remains absent
from the clues. Each clue must have the syntax of a bare noun phrase (zero determiner): the root node of each clue must be a
common or proper noun and it can be followed by a relative clause or other complements or adjuncts.  Generate clues from all the
parts of the text and use all of the information provided to generate the clues.
5. Ensure that each clue functions as a description or definition of the keyword rather than a query, focusing on details about
the keyword.
6. Make sure that each clue's information can be traced back to the text. Make sure that the clues are relevant and that they are
sufficient to identify the keyword. Make sure that the keyword does not appear in the clues. Make sure that any part of the
keyword is not present in the clues.
7. Select only the three best clues for educational purposes.
8. Compile these clues into a list formatted as follows: [clue1, clue2, clue3]  into a JSON file under the key: 'clues'. Make sure
the output is in the requested format and do not include the whole process in the output, but only the clues.
```

Figure 5.4: **Prompt template: Bare Noun Phrase.** Require a determinerless NP headed by a common or proper noun; post-nominal material (e.g., relative clauses, PPs) is allowed.

Human Evaluation of Italian-Clue-Instruct Dataset Quality. To ensure the high quality and pedagogical suitability of the synthetically generated Italian-Clue-Instruct dataset, a rigorous human evaluation was performed on a significant subset of its content. A random sample of 600 examples, corresponding to 1,800 individual clues, three clue per example using the three introduced prompts, was manually annotated by human evaluators. Each clue was assigned a rating based on a five-level scale, meticulously defined to capture various aspects of clue quality and educational relevance. The rating criteria were as follows:

- **RATING-A:** The clue is impeccable: valid, well-formed, and perfectly coherent

```
You are a crossword expert.
Generate concise and clever clues in Italian for educational crossword puzzles based on a specified Keyword and its relation to an
assigned Text. To execute this task properly, replicate the guidelines below:
KEYWORD: {keyword}
TEXT: {text}

Observe the following steps:
1. Substitute every pronoun in the text with full phrases expressing their referents.
2. Split the text into small independent sentences that could be understood out of context.
3. Pinpoint three concise sentences that contain the Keyword and best characterize the keyword. Try to select sentences from
different parts of the Text.
4. Generate short and clever crossword clues in Italian from the selected sentences. Make sure that  the keyword remains absent
from the clues. Each clue must have the syntax of a determiner phrase with the definite article (followed by a noun and possibly
adjectives). It can be followed by a relative clause or other complements or adjuncts. Generate clues from all the parts of the
text and use all of the information provided to generate the clues.
5. Ensure that each clue functions as a description or definition of the keyword rather than a query, focusing on details about
the keyword.
6. Make sure that each clue's information can be traced back to the text. Make sure that the clues are relevant and that they are
sufficient to identify the keyword. Make sure that the keyword does not appear in the clues. Make sure that any part of the
keyword is not present in the clues.
7. Select only the three best clues for educational purposes.
8. Compile these clues into a list formatted as follows: [clue1, clue2, clue3]  into a JSON file under the key: 'clues'. Make sure
the output is in the requested format and do not include the whole process in the output, but only the clues.
```

Figure 5.5: **Prompt template: Definite Determiner Phrase.** Require a DP beginning with the definite article (e.g., *il/la/lo/l'/i/le/gli*), followed by a noun (optionally with adjectives) and possible modifiers.

```
Generate concise and clever clues in Italian for educational crossword puzzles based on a specified Keyword and its relation to an
assigned Text. To execute this task properly, replicate the guidelines below:
KEYWORD: {keyword}
TEXT: {text}

Observe the following steps:
1. Substitute every pronoun in the text with full phrases expressing their referents.
2. Split the text into small independent sentences that could be understood out of context.
3. Pinpoint three concise sentences that contain the Keyword and best characterize the keyword. Try to select sentences from
different parts of the Text.
4. Generate short and clever crossword clues in Italian from the selected sentences. Make sure that  the keyword remains absent
from the clues. Each clue must be a copular sentence, in which the keyword constitutes the subject. The syntax of each clue then
must corresponds to a copular sentence without the subject. For example: "è <clue>". Generate clues from all the parts of the text
and use all of the information provided to generate the clues.
5. Ensure that each clue functions as a description or definition of the keyword rather than a query, focusing on details about
the keyword.
6. Make sure that each clue's information can be traced back to the text. Make sure that the clues are relevant and that they are
sufficient to identify the keyword. Make sure that the keyword does not appear in the clues. Make sure that any part of the
keyword is not present in the clues.
7. Select only the three best clues for educational purposes.
8. Compile these clues into a list formatted as follows: [clue1, clue2, clue3]  into a JSON file under the key: 'clues'. Make sure
the output is in the requested format and do not include the whole process in the output, but only the clues.
```

Figure 5.6: **Prompt template: Copular schema.** Require a copular frame with an elided subject of the form "è . . . ", allowing predicative nominals/adjectives or prepositional phrases; no overt subject.

with the provided context, the target answer, and the specified category. These clues represent the gold standard for educational puzzles.

- **RATING-B:** The clue is acceptable but contains minor imperfections, such as a slightly loose correlation with the specified category or a minor stylistic issue that does not impede understanding.
- **RATING-C:** The clue is relevant to the answer but is overly generic or correlates only loosely with the source context. While understandable, it may not effectively reinforce specific textual knowledge.
- **RATING-D:** The clue is largely incorrect or irrelevant with respect to either the

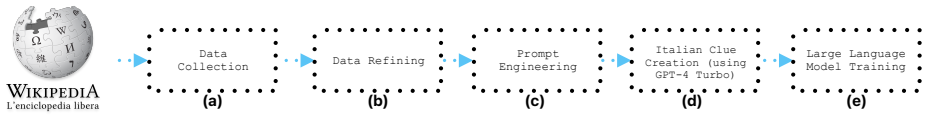


Figure 5.7: The methodology for creating the Italian-Clue-Instruct dataset: (a) data collection from Italian Wikipedia, (b) rigorous data refining and filtering, (c) engineering of specialized prompts for different clue styles, (d) clue generation using a powerful teacher model (GPT-4o), and (e) using the resulting dataset to fine-tune smaller, open-source student models.

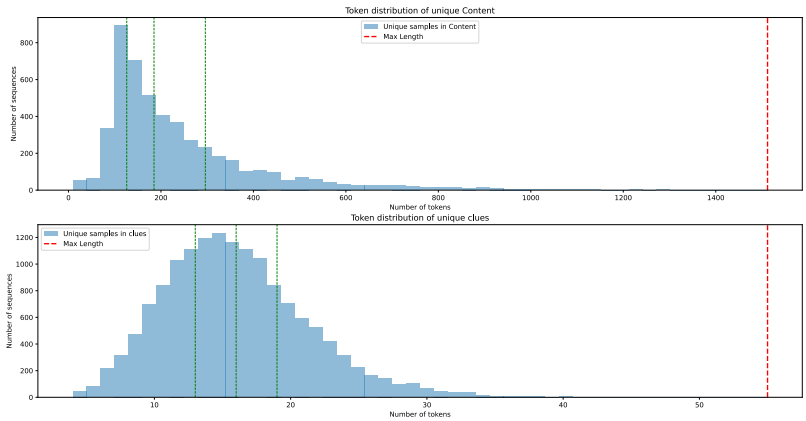


Figure 5.8: Token distributions for contexts and clues in the Italian-Clue-Instruct dataset, processed using the Llama3 tokenizer.

answer or the provided source context, making it unsuitable for educational use.

- **RATING-E:** The clue is unacceptable because it contains the target answer itself or a direct variant of it, rendering the puzzle trivial or unsolveable in a meaningful way.

The results of this comprehensive evaluation, depicted in Figure 5.10, robustly confirm the high quality of the Italian-Clue-Instruct dataset. When considering both RATING-A and RATING-B clues as pedagogically acceptable, the overall acceptance rate rises to an impressive **54%**. This high level of quality is crucial, as it provides a strong foundation for the subsequent instruction-tuning of smaller, open-source models. Table 5.1 provides concrete examples of clues from the dataset, showcasing their variety and illustrating the application of the rating system.

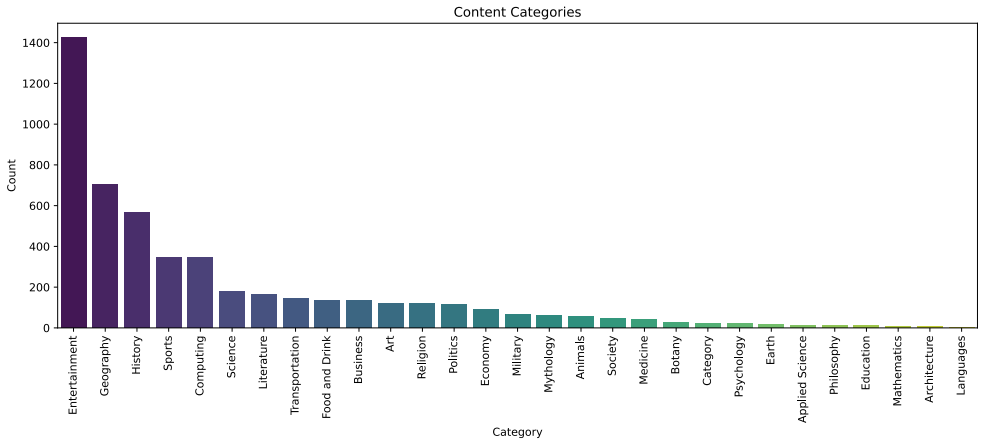


Figure 5.9: Distribution of the Italian-Clue-Instruct dataset across 29 categories, with "Entertainment," "Geography," and "History" being the most represented .

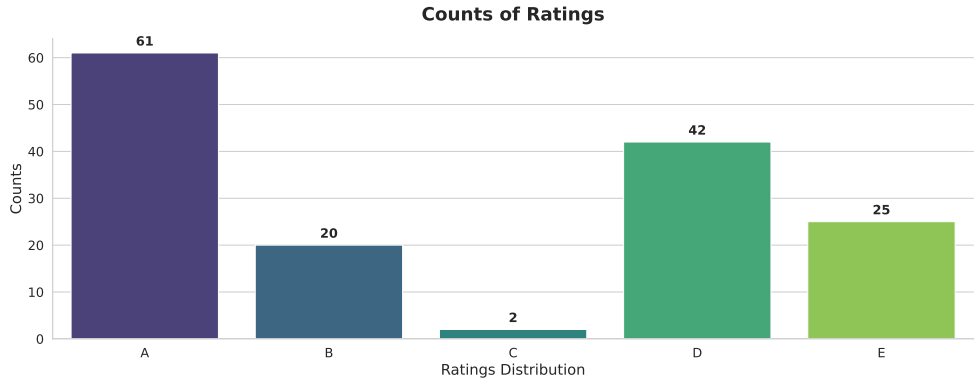


Figure 5.10: Distribution of human-assigned ratings on a sample of 1,800 clues from the Italian-Clue-Instruct dataset, demonstrating the high overall quality and suitability for educational purposes.

5.4 Methodology: From Italian Text to Puzzle

We employ the unified 'From Language to Learning' pipeline established in Chapter 4 (see Section 4.4). While the core architecture for validation and grid assembly remains consistent with the English framework, the Italian implementation introduces a critical adaptation within the text-conditioned generation path: Syntactic Style Control.
The framework's methodology is composed of three primary stages: Clue Generation, which can follow one of two paths; a shared Validation and Filtering stage; and the final Schema Generation stage.

Table 5.1: Examples of generated clues from the Italian-Clue-Instruct dataset with their assigned human ratings, illustrating the quality criteria applied during evaluation.

Answer	**Category**	**Clue**	**Rating**
Robocall	Society	Può essere bloccata dalle compagnie telefoniche per prevenire truffe.	A
Ministero della Magia	Literature	Governo corrotto e incompetente nel Mondo Magico di J.K. Rowling.	A
Lovesick	Literature	Rinnovata per una terza stagione, rilasciata esclusivamente su Netflix.	C
Tapirro Sudamericano	Science	Una delle quattro specie riconosciute nella famiglia dei tapiri.	E

5.4.1 Clue Generation

Path (a): Text-Conditioned Generation with Style Control

The text-conditioned path is the most innovative component of the framework, offering fine-grained control over the linguistic style of the generated clues. This is particularly valuable in an educational context, as it allows teachers to create puzzles with varying levels of syntactic complexity, tailored to the proficiency of their students.

Syntactic Style Control. The system is capable of generating clues in four distinct styles, each guided by a specialized prompt. These styles are based on specific linguistic structures in Italian:

1. **Unrestricted Format:** A default style with no explicit syntactic constraints, serving as a baseline.
2. **Definite Determiner Phrases:** These clues are nominal phrases headed by a definite article (e.g., *La repubblica asiatica con capitale Tashkent*, 'The Asian republic with Tashkent as capital'). This structure presupposes a unique or maximal referent, making the clues highly specific.
3. **Bare Noun Phrases:** These clues consist of a noun phrase without a determiner (e.g., *Grande centro commerciale di lusso con sede a Londra*, 'Luxury shopping mall based in London'). In Italian, this structure denotes a predicate and does not carry a uniqueness presupposition, potentially allowing for more ambiguity and challenge .
4. **Copular Sentences:** These clues are structured as a clausal definition with an elliptical subject (e.g., *è una salsa piccante tipica della Tunisia*, '(It) is a spicy sauce typical of Tunisia'). The answer to the clue is the elided subject of the sentence .

Fine-Tuning for Style Adherence. To reliably generate clues in these specific styles, we fine-tuned two open-source LLMs, **Mistral-7B-Instruct-v0.3** and **Llama3-8b-Instruct**, on the Italian-Clue-Instruct dataset . The fine-tuning was performed efficiently using LoRA [121] ($r = 16, \alpha = 32$) over three epochs. This process specialized the models not only to generate contextually relevant clues but also to adhere to the specified syntactic structures with high fidelity.

Path (b): Answer-Only Generation

For the answer-only path, the framework relies on a different set of fine-tuned models. We fine-tuned **GPT3-DaVinci**
and **GPT3-Curie** on a 50,000-pair subset of the legacy Italian clue-answer corpus. The training process involved providing the model with an answer and tasking it to generate the corresponding clue. This method trains the models to capture the diverse patterns and styles of human-authored clues, enabling them to generate high-quality definitions from a single keyword.

5.4.2 Shared Component: The Validation Cascade

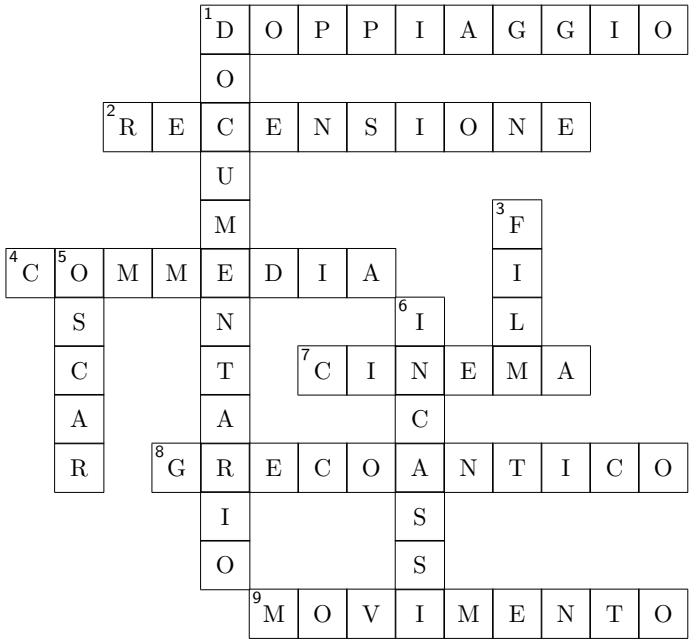
To ensure the quality and pedagogical suitability of the generated content, outputs from both paths undergo a rigorous validation process.

- **For Path (a):** In our initial work, validation was performed using a zero-shot prompting approach, where an LLM was asked to verify if the content of a generated clue was present in the source text. The fine-tuned models developed in our later work, however, showed a much higher degree of faithfulness, reducing the need for this explicit check.

- **For Path (b):** Since there is no source text, a learned classifier is essential. We trained a suite of models—including GPT3-DaVinci, GPT3-Curie, GPT3-Babbage, GPT3-Ada, [122] and BERT-uncased-base—as binary classifiers on a human-labeled dataset of 6,000 clue-answer pairs. This allows the system to effectively filter out unacceptable clues generated by the answer-only models.

5.4.3 Shared Component: Schema Generation

The final stage of the framework, schema generation, is handled by an algorithm consistent with the one described for the English framework in the previous chapter in the section4.4.3. It employs a randomized constructive search to assemble the validated clue-answer pairs into a compact and highly interlocked grid. The process is guided by a composite scoring function that optimizes for the number of filled words, the density of letter crossings, and the overall filled ratio of the grid. This ensures the creation of aesthetically pleasing and solvable puzzles that are well-suited for educational use. An example of a final generated crossword is shown in Figure 5.11.



Orizzontali 1 ⋆ Traduzione Simultanea 2 ⋆ Una valutazione critica 4 ⋆ Lo è uno spassoso racconto 7 Insieme delle arti, tecniche e attività industriali produttive di un film 8 Lingua originale da cui deriva la parola cinema 9 Uno dei termini estratti dall'antica lingua greca, utilizzato per descrivere il cinema **Verticali** 1 ⋆ Un film... come documento 3 Prodotto commerciale finale di un insieme di lavoro compredente arti, tecniche e attività industriali 5 ⋆ Un premio assai ambito 6 ⋆ Entrano nelle casse del botteghino

Figure 5.11: An illustrative Italian educational crossword, generated by the system using clue-answer pairs from both the text-conditioned and answer-only paths.

5.5 Experimental Validation

We conducted a series of experiments to evaluate the performance of each component of the Italian crossword generation framework.

5.5.1 Validation of Clue Generation Models

Path (a): Instruction-Tuned Models. The effectiveness of fine-tuning for text-conditioned clue generation was measured using both automatic metrics and human evaluation.

- **Automatic Metrics:** We used ROUGE scores to measure the lexical similarity between clues generated by our fine-tuned models and those generated by the powerful GPT-4o [27] teacher model. As shown in Table 5.2, the fine-tuned Mistral-7B [127] and Llama3-8B [126] models produce outputs that are significantly more

similar to the high-quality reference clues than their base versions, demonstrating the success of the fine-tuning process.

- **Human Evaluation:** A human evaluation was conducted on a sample of 100 Italian contexts. The results, depicted in Figure 5.12, show a dramatic improvement in quality after fine-tuning. For example, the base Llama3-8B model produced a large number of low-quality clues (rated 'E'), while the fine-tuned version produced a majority of high-quality 'A'-rated clues. The fine-tuned Mistral-7B model emerged as the top performer, with the highest number of 'A'-rated clues.

Table 5.2: Mean ROUGE scores comparing the clues from base and fine-tuned models against a strong reference (GPT-4o). Fine-tuning significantly increases the similarity.

Model Type	Model Name	R-1	R-2	R-L
Base LLMs	Mistral-7B	0.342	0.176	0.261
	Llama3-8b	0.258	0.112	0.198
Fine-tuned	Mistral-7B	**0.611**	**0.458**	**0.556**
	Llama3-8b	0.552	0.403	0.501

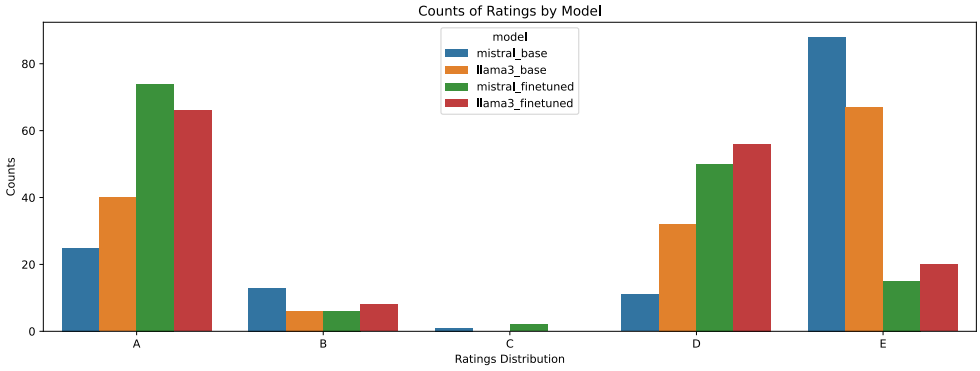


Figure 5.12: Human evaluation results for the base vs. fine-tuned models on Italian clue generation. Fine-tuning leads to a significant increase in high-quality 'A'-rated clues and a reduction in low-quality ones.

Path (b): Answer-Only Models. The answer-only generators were evaluated through human supervision. As shown in Table 5.3, the larger GPT3-DaVinci model significantly outperformed the smaller GPT3-Curie model, achieving a **60.1%** acceptability rate.

5.5.2 Validation of the Filtering Component

The learned classifiers for the answer-only path were tested on a held-out set of 20% of the 6,000 labeled examples. The results, detailed in Table 5.4, show that the GPT3-DaVinci

Table 5.3: Acceptable clue rates for answer-only Italian generators, determined by human evaluation.

Model	% acceptable clues
GPT3-DaVinci	60.1
GPT3-Curie	34.9

model was the most effective classifier, achieving an accuracy of **79.88%** and an F1-score of **0.7838**.

Table 5.4: Performance of the learned validators for Italian on distinguishing acceptable vs. unacceptable clue-answer pairs.

Model	Accuracy %	Precision %	Recall %	F1
GPT3-DaVinci	79.88	80.16	76.67	0.7838
GPT3-Curie	77.82	78.80	72.99	0.7578
GPT3-Babbage	74.12	72.58	73.25	0.7291
GPT3-Ada	69.17	67.77	67.06	0.6741
BERT-uncased	65.62	63.71	64.47	0.6409

5.6 Conclusions

This chapter presented a comprehensive, unified framework for the automated generation of Italian educational crossword puzzles. A key contribution of this work is the creation and release of the **Italian-Clue-Instruct** dataset, the first large-scale, text-aligned corpus for this task in Italian. This resource enabled a significant methodological advancement: the move from less reliable few-shot prompting to a robust instruction-tuning paradigm.

Our experiments demonstrate that by fine-tuning open-source models like Mistral-7B and Llama3-8B on this dataset, we can generate high-quality, contextually grounded clues with fine-grained control over their syntactic style. The framework's flexibility is further enhanced by an alternative path for answer-only generation and a rigorous validation cascade that ensures the pedagogical quality of the outputs. The resulting system represents a powerful and practical tool for Italian educators, enabling them to create customized, engaging, and effective learning materials with ease. This work not only addresses a critical resource gap for the Italian language but also provides a replicable blueprint for developing similar educational tools in other languages.

Chapter 6

An LLM-Driven Framework for Turkish Educational Crossword Generation

6.1 Motivation and Scope

Educational crossword puzzles are highly effective pedagogical tools, crucial for reinforcing terminology, concepts, and named entities through active recall. They foster an interactive learning environment that significantly enhances vocabulary acquisition, spelling proficiency, and the retention of specialized knowledge across various disciplines. In the context of the Turkish language, these puzzles are particularly valuable for strengthening grammatical structures and specialized lexicons in subjects ranging from history and geography to science and literature. By engaging students in critical thinking to match clues with appropriate answers, educational crosswords boost memory recall and improve problem-solving abilities, transforming the learning process into an engaging and motivating experience.

Despite these recognized benefits, the automated creation of high-quality educational crosswords in Turkish presents a unique set of challenges. Turkish, an agglutinative language, poses distinct hurdles: its complex morphology (where suffixes are added to root words to convey grammatical function) significantly increases the risk of "leakage"—where the answer or its stem inadvertently appears within the clue itself, making the puzzle trivial or unsolveable. Furthermore, the inherent ambiguity that can arise from highly inflected words makes crafting concise, unambiguous, and grammatically precise clues a labor-intensive task. Traditionally, the creation of such puzzles in Turkish has been a manual process, demanding not only deep subject-matter expertise but also considerable linguistic creativity, which limits the widespread availability of customized, curriculum-aligned puzzles for Turkish educators.

This manual authoring bottleneck is further exacerbated by a significant lack of specialized digital resources for the Turkish language in the context of Large Language Models (LLMs). Existing digital corpora are typically designed for entertainment-style crosswords and do not provide the essential link between a clue and a source text, which is paramount for educational validity and contextually grounded learning. The morpho-syntactic complexity of Turkish, combined with its relatively limited educational data resources, makes off-the-shelf LLM prompting unreliable for generating pedagogically sound content.

To address these critical gaps—data scarcity, linguistic complexity, and automated assembly—this chapter presents an comprehensive framework for the automated generation of Turkish educational crosswords. This system is distinguished by its ability to produce contextually grounded and stylistically controlled clues, leveraging two purpose-built

Turkish datasets: a large corpus of expert-authored answer–clue pairs (TAC) and a novel text-aligned corpus (T4TAC). Our approach integrates state-of-the-art LLMs, including fine-tuned GPT-3.5-Turbo and LLaMA-2 models, into a modular pipeline that encompasses data preparation, style-specific clue generation, rigorous validation, and automated grid construction. By introducing fine-tuned, open-source models specifically for this task, this work moves beyond earlier reliance on few-shot learning with general-purpose models, offering a robust and professional solution for creating engaging, interactive, and pedagogically sound learning tools for the Turkish language.

6.2 Framework Overview

The presented framework is architected as a coherent, end-to-end system that offers two distinct operational pathways for Turkish clue creation, providing educators with the flexibility to generate puzzles from various types of source material. This dual-path design ensures the system can be adapted to a wide range of classroom needs, from topic reviews to in-depth comprehension exercises. Both pathways feed into a shared validation and puzzle assembly stage, ensuring consistent quality and efficient grid construction. The overall system architecture is conceptually illustrated in Figure 6.1.

- **Path (a): Text-Conditioned Generation. (T4TAC)** This is the primary and most advanced mode, specifically designed for creating deeply curriculum-aligned puzzles directly from a given Turkish educational text. In this pathway, the system first extracts salient keywords from the source material. It then employs specialized, instruction-tuned Large Language Models to generate multiple concise and factual clues that are semantically grounded in the provided context, implicitly leading to the extracted keyword without explicitly stating it. A key innovation of this path is its ability to ensure contextual relevance, which is crucial for educational validity.

- **Path (b): Answer-Only Generation. (TAC)** This pathway caters to scenarios where an instructor provides a predefined list of keywords or vocabulary terms, without an accompanying source text. This is a common requirement for creating vocabulary reinforcement exercises, spelling drills, or quick topic reviews. Here, the framework utilizes LLMs fine-tuned on a large legacy corpus of Turkish clue-answer pairs (TAC) to generate relevant, definition-style clues for each keyword.

A cornerstone of the framework's design is that both clue generation paths converge into a common, multi-stage validation pipeline. This shared quality assurance process applies structural checks and leakage control to ensure that all clue-answer pairs meet high pedagogical and linguistic standards, particularly in guarding against the pitfalls of Turkish agglutinative morphology. The final, validated pairs are then passed to a schema (grid) generator, which algorithmically assembles the puzzle grid. This modular architecture, abstracting functionality into distinct roles (e.g., generator, validator, layout engine), allows for straightforward updates and makes the framework easily adaptable and extensible.

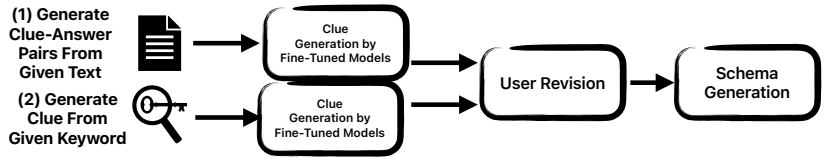


Figure 6.1: The overall system architecture for Turkish educational crossword generation, illustrating the two main paths for clue creation and the shared validation and schema generation stages.

6.3 Foundational Data Resources for Turkish

The capabilities of the Turkish crossword generation framework are built upon two substantial and complementary Turkish-language datasets. The first is a large-scale collection of traditional clue-answer pairs, while the second is a novel, text-aligned corpus created specifically for educational clue generation. These resources are critical for overcoming the data scarcity challenge in Turkish.

6.3.1 TAC: A Corpus of Expert-Authored Turkish Clue-Answer Pairs

To power the answer-only clue generation path (Path b) and to provide a foundational understanding of Turkish crossword clue styles for training models, a comprehensive dataset of expert-authored Turkish crossword clues was compiled. This dataset, named **TAC** (Turkish Answer–Clue pairs), aggregates 252,576 answer–clue pairs sourced from experienced constructors, including various online platforms and renowned newspaper sources such as Habertürk.

After a thorough cleaning process to merge sources and remove duplicates, the final TAC dataset comprises 54,984 unique answers, 178,431 unique clues, and 187,495 unique answer-clue entries. This extensive corpus captures a wide range of domains, including general knowledge, history, geography, literature, and popular culture, and exhibits diverse linguistic and structural patterns characteristic of human-authored Turkish crossword clues. The answer-length distribution within TAC, which is crucial for insights into typical grid sizing and expected crossing density in Turkish puzzles, is showcased in Figure 6.2. This dataset is a critical resource, providing the necessary supervision for training models to understand and replicate the stylistic conventions of Turkish crossword clues.

6.3.2 T4TAC: A Text-Aligned Corpus for Instruction Tuning

The central limitation of existing Turkish crossword resources, including TAC, is their unsuitability for educational, text-grounded clue generation tasks. To overcome this, we created **T4TAC** (Text for Turkish Answer–Clue pairs), a novel dataset that, for the first time in Turkish, systematically links crossword clues to a source text. This resource is

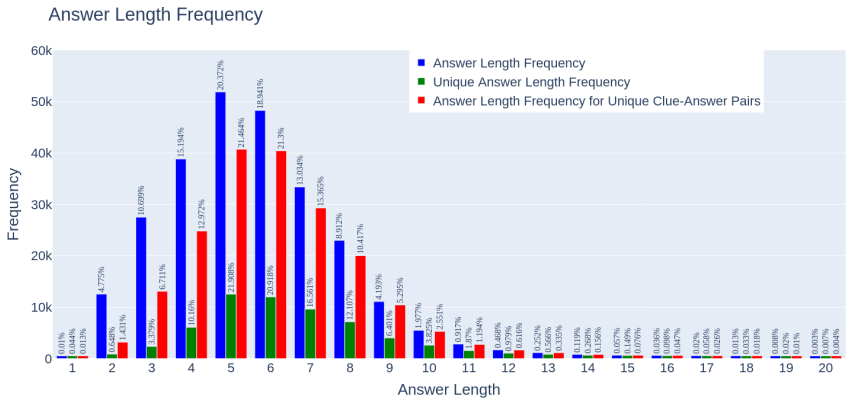


Figure 6.2: The dataset entries are showcased visually through the distribution of answer lengths in the TAC corpus. Blue bars represent all answer–clue pairs, green bars show the frequency of unique answers, and red bars display the frequency of unique answer–clue pairs.

specifically designed to enable instruction-tuning of LLMs for generating contextually relevant and pedagogically sound clues.

Data Collection and Curation. The T4TAC dataset was built using a meticulous data collection pipeline, as illustrated in Figure 6.3. The process began by extracting introductory sections from approximately 180,000 Turkish Wikipedia pages. A multi-stage filtering process was then rigorously applied to ensure high data quality and educational relevance. We prioritized articles based on popularity and relevance, discarded texts shorter than 50 words, and constrained keywords to be between 3 and 20 characters in length, containing no numerics or special characters. This stringent filtering pipeline yielded a refined set of 9,855 high-quality pages across 29 distinct thematic categories.

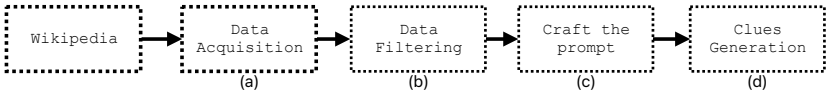


Figure 6.3: Diagram outlining the detailed steps followed in the construction of the T4TAC dataset, from initial data collection to refined output.

Synthetic Clue Generation. From this curated set of text-aligned data, we employed a powerful, state-of-the-art Large Language Model, GPT-4-Turbo, as a "teacher model" to generate a variety of educational clues for each article. This process was guided by highly

structured and "leakage-aware" prompt, designed using a self-instruct-style procedure. This prompt ensured that the generated clues were concise, factual, and educational, implying the keyword without directly revealing it—a crucial aspect for agglutinative languages like Turkish where leakage is a common pitfall. The exact prompt structure utilized for this task is provided in Figure 7.4. The teacher model produced multiple distinct clues for each (context, keyword, category) triplet, resulting in a final T4TAC dataset containing over **35,000 educational clues**.

```
Prompt: As an assistant skilled in creating Turkish crossword clues, your task is to generate concise, clever, and accurate clues
for crossword puzzles. This task is designed for crossword puzzle creators looking to engage their audience with intriguing clues.
Here are the details:

KEYWORD: {keyword}
CATEGORY: {category}
CONTEXT: {text}

Your task involves the following steps:

1. Analyze the context to identify the most relevant and informative connections between '{keyword}' and '{category}'.
Extract key facts from the context, considering the length and complexity.

2. Craft short, clever clues in Turkish that entice solvers to uncover '{keyword}'. Use wordplay, metaphors, puns, and other
creative techniques to make the clues engaging and thought-provoking.

3. '{keyword}' should be the answer to the clue. Avoid using the 'keyword' or its derivatives directly in the clues. Instead, use
descriptive phrases, synonyms, or creative wordplay that hints at '{keyword}' without revealing it outright.

4. Generate {count} clues. Aim for conciseness, cleverness, and accuracy in all clues. Remember, the best clues are not only
accurate but also fun and challenging for the solver.

5. Provide clues and answers in the following JSON structure:
{ 'clues': [{'clue': '', 'answer': ''}] }
```

Figure 6.4: The structured prompt utilized with the GPT-4-Turbo teacher model for generating educational Turkish clues within the T4TAC dataset. This prompt is designed to ensure contextual relevance, conciseness, and avoid clue leakage.

Dataset Statistics and Human Evaluation. The statistical properties of the T4TAC dataset are summarized in Figure 6.5. This figure provides insights into: (i) the word and character length distributions for contexts, generated clues, and keywords, (ii) the thematic distribution of the dataset across its 29 categories, and (iii) the results of a human evaluation conducted to ascertain the quality of the synthetically generated clues. The human evaluation for T4TAC indicated a high percentage of acceptable clues, validating the effectiveness of the teacher model and the prompt engineering strategy. This rich, text-aligned corpus is the foundational resource that enables the instruction-tuning of smaller, open-source models for the high-fidelity text-conditioned generation path.

Human Evaluation (T4TAC). From panel (iii) of Figure 6.5, the dataset-level ratings for the teacher-generated clues are: **A** = 68.0%, **B** = 18.7%, **C** = 9.3%, **D** = 3.0%, and **E** = 1.1%. These values (read from the figure) imply an overall *acceptance rate* of $A{+}B{+}C \approx \mathbf{96.0}\%$, indicating that the vast majority of clues are acceptable or better.

6.4 Methodology: From Turkish Text to Puzzle

This framework instantiates the methodological pipeline defined in Chapter 4, adapting the core generator and validator components to address the specific challenges of Turkish morphology. The framework's methodology is composed of three primary, interconnected

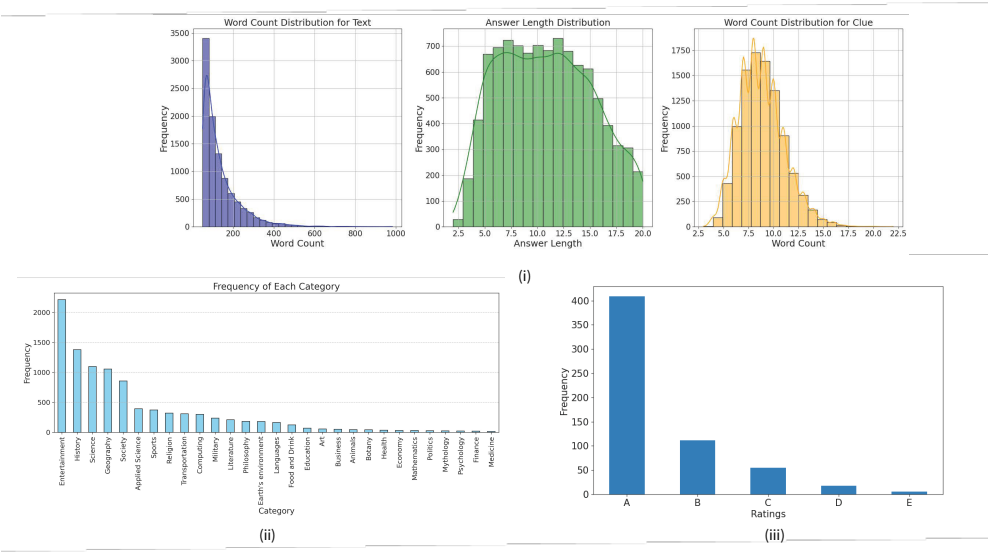


Figure 6.5: (i) Word and character length distributions for contexts, outputs, and keywords in the T4TAC dataset. (ii) Category distributions of T4TAC across its 29 themes. (iii) Human evaluation results for the quality of generated clues within the T4TAC dataset.

stages: Clue Generation, which offers two distinct pathways; a shared Validation and Filtering stage; and the final Schema (Grid) Generation stage.

6.4.1 Clue Generation

The core of the framework lies in its specialized LLM-based clue generators, each tailored for a specific authoring mode.

Path (a): Text-Conditioned Generation with Context

The text-conditioned path is designed to produce highly relevant and contextually grounded clues directly from educational texts. This is particularly valuable as it allows for the creation of puzzles that align directly with specific learning materials. To achieve this, we fine-tuned several LLMs on the T4TAC dataset.

We used a training split of 8,000 unique (context, keyword, category) items and a test set of 670 items from T4TAC. For the proprietary model, we fine-tuned **GPT-3.5-Turbo** with a batch size of 16 and a learning rate of 0.001 over 3 epochs. For open-source alternatives, we fine-tuned **LLaMA-2 7B** and **LLaMA-2 13B** models using Parameter-Efficient Fine-Tuning (PEFT) with the LoRA [121] technique. The LoRA configuration included a rank (r) of 16, an alpha (α) of 32, and a learning rate of 10^{-4} over 3 epochs. This fine-tuning process specialized the models not only to generate contextually relevant Turkish clues but also to adhere to the desired educational tone and conciseness, while

mitigating issues like leakage.

Path (b): Answer-Only Generation from Keyword Lists

For scenarios where educators provide a list of keywords without an explicit source text, the framework leverages the answer-only generation path. This mode is ideal for creating vocabulary review puzzles or general knowledge crosswords.
For this path, we fine-tuned **GPT-3.5-Turbo** on a subset of 60,000 expert-authored pairs from the TAC dataset. The training process involved providing the model with an answer (keyword) and tasking it to generate a corresponding clue. The fine-tuning was performed with a batch size of 16, a learning rate of 0.01, over 3 epochs. This method trains the model to capture the diverse patterns and styles of human-authored Turkish crossword clues, enabling it to generate high-quality, definition-style clues from a single keyword, independent of a specific context.

6.4.2 Shared Component: The Validation Cascade

The final stage—schema generation—uses the same algorithm as the English framework (Section 4.4.3). A randomized constructive search arrango ensure the quality and pedagogical suitability of the generated content, outputs from both clue generation paths undergo a rigorous validation process. Given the morphological complexity of Turkish, this stage is crucial for ensuring robustness and preventing clue leakage.

The validator applies several checks:

- **Structural Checks:** Ensures that clues adhere to format requirements (e.g., length constraints, character sets) and basic grammatical correctness.
- **Leakage Detection:** This is a critical component for Turkish. It performs surface-level and stem-aware checks to prevent the answer word or its morphological variants from appearing within the clue. This helps maintain the challenge and integrity of the puzzle. Near-synonym leakage is also guarded against.
- **Informativeness:** For answer-only clues, the validator also screens for minimal informativeness, ensuring the generated clue provides sufficient information without being overly generic.
- **LLM-as-Judge:** The framework's design allows for the integration of an LLM-as-judge or a learned acceptability classifier to further vet clue quality. For the Turkish framework, the high quality of the T4TAC dataset and the rigorous fine-tuning significantly reduce the need for an explicit post-generation validation LLM for Path (a).

Accepted and validated clue-answer pairs are then passed to the schema generation stage.

6.4.3 Shared Component: schema generation

The final stage—schema generation—uses the same algorithm as the English framework (Section 4.4.3). A randomized constructive search arranges validated clue–answer pairs

into a compact, highly interlocked grid, guided by a composite score that maximizes filled entries, crossing density, and overall fill ratio. This yields aesthetically pleasing, solvable puzzles suited for education.

6.5 Experimental Validation

We conducted a series of experiments to rigorously evaluate the performance of each component of the Turkish crossword generation framework, focusing on both automatic metrics and human evaluations.

6.5.1 Validation of Clue Generation Models

Path (a): Text-Conditioned Generation Models. The effectiveness of fine-tuning for text-conditioned clue generation was measured using both automatic metrics and human evaluation, comparing base LLMs against their fine-tuned counterparts on the T4TAC test split.

- **Automatic Metrics (ROUGE Scores):** We used ROUGE-1, ROUGE-2, and ROUGE-L F1 scores to measure the lexical similarity between clues generated by our fine-tuned models and strong reference clues (generated by a powerful teacher model). As summarized in Table 6.1, fine-tuned GPT-3.5-Turbo [122] substantially outperforms its base version, demonstrating significant gains across all ROUGE metrics. Similarly, the LLaMA-2 13B [123] model fine-tuned with PEFT [120] shows improved performance over its 7B counterpart, and both fine-tuned LLaMA models exceed their respective base versions. These results confirm the success of the fine-tuning process in aligning the models' output with high-quality, reference-style Turkish clues.

model type	model name	# params	ROUGE-1	ROUGE-2	ROUGE-L
Base LLMs	LLAMA2-CHAT	7B	–	–	–
	LLAMA2-CHAT	13B	–	–	–
	GPT3.5 TURBO	–	24.57	8.48	16.61
Finetuned LLMs	LLAMA2-CHAT	7B	22.50	5.00	14.80
	LLAMA2-CHAT	13B	25.66	6.45	17.24
	GPT3.5 TURBO	–	**39.96**	**18.08**	**23.48**

Table 6.1: Performance of LLMs with and without fine-tuning on text-conditioned Turkish clue generation (T4TAC test set), measured by ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. Fine-tuning consistently improves similarity to strong references.

- **Human Evaluation (Acceptability Ratings):** A human evaluation was conducted on a sample of 200 text-conditioned items, yielding 600 individual clues (3 clues per item). The clues were rated by native Turkish speakers based on quality, contextual relevance, and absence of leakage. The results, depicted in Figure 6.6,

demonstrate a dramatic improvement in clue quality after fine-tuning for all models. Fine-tuned models consistently produced a higher percentage of acceptable clues and significantly fewer malformed or unusable outputs compared to their base versions. Specifically, fine-tuned GPT-3.5-Turbo showed the highest overall acceptability, while the fine-tuned LLaMA-2 models, despite lower ROUGE scores, also demonstrated marked improvements over their base counterparts, becoming viable alternatives for Turkish clue generation.

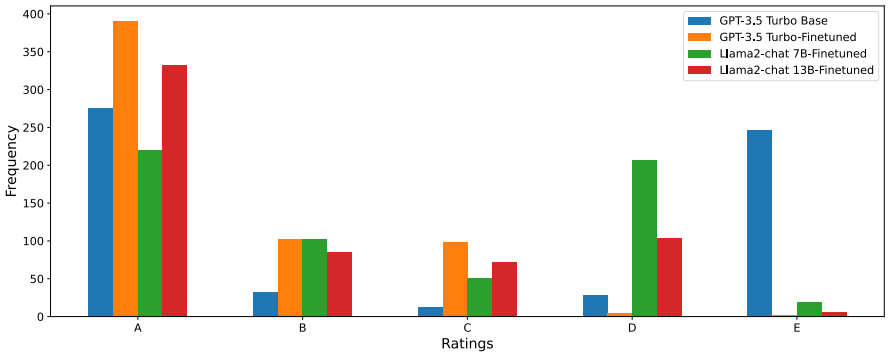


Figure 6.6: Human evaluation results comparing base vs. fine-tuned LLMs on text-conditioned Turkish clue generation. Fine-tuning leads to a significant increase in high-quality (acceptable) clues and a reduction in low-quality or malformed outputs.

Path (b): Answer-Only Generation Model. The answer-only generator (fine-tuned GPT-3.5-Turbo on TAC) was evaluated through human supervision on an out-of-domain keyword set of 2,135 academic keywords. Two native-speaker judges assessed the acceptability of the generated clues. This pathway yielded the **51.8% acceptability rate** for the generated clues. This finding highlights the viability and practical utility of this path, especially for educators who primarily work with canonical term lists or vocabulary sets without specific textual contexts.

6.5.2 Grid Quality and End-to-End Artifact

The schema (grid) generator's effectiveness was confirmed by its ability to produce dense, compact, and highly interlocked crossword puzzles.
To demonstrate the full end-to-end capabilities of the framework, a complete Turkish crossword puzzle generated by the system is presented in Figure 6.7. This artifact showcases the successful integration of clue generation (from either path), validation, and grid assembly into a unified, functional system capable of producing classroom-ready educational materials.

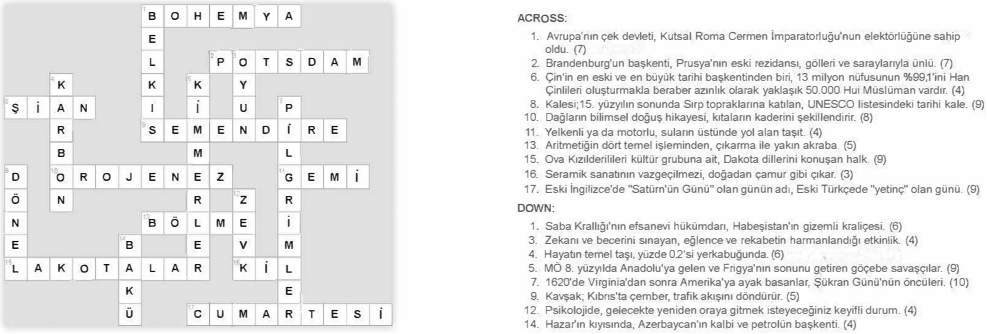


Figure 6.7: An illustrative Turkish educational crossword, generated by the unified framework. This artifact demonstrates the successful end-to-end integration of clue generation, validation, and grid assembly.

6.6 Conclusions

This chapter presented a comprehensive, unified framework for the automated generation of Turkish educational crossword puzzles. This work addresses a critical gap in resources and tooling for the Turkish language, a morphologically complex and under-resourced language in the context of LLM-driven educational content creation.

A key contribution of this work is the creation and release of two foundational Turkish-language datasets: **TAC**, a large corpus of expert-authored clue-answer pairs, and **T4TAC**, a novel, text-aligned corpus specifically designed for instruction-tuning LLMs for text-conditioned clue generation. This rich data ecosystem enabled a significant methodological advancement: the move from less reliable few-shot prompting to a robust instruction-tuning paradigm, specifically tailored for the linguistic nuances of Turkish.

Our extensive experiments robustly demonstrate that by fine-tuning open-source models like LLaMA-2 (7B and 13B) and proprietary models like GPT-3.5-Turbo on these purpose-built datasets, we can generate high-quality, contextually grounded Turkish clues with significant improvements in both automatic ROUGE scores and human acceptability. The framework's flexibility is further enhanced by its dual-path architecture, supporting both text-conditioned generation for curriculum-aligned content and answer-only generation for vocabulary-focused exercises. A rigorous validation cascade, crucial for guarding against clue leakage in an agglutinative language, and an optimized schema generator ensure the pedagogical quality and structural integrity of the final puzzles.

The resulting system represents a powerful and practical tool for Turkish educators, enabling them to easily create customized, engaging, and effective learning materials. This work not only addresses a critical resource and tooling gap for the Turkish language but also provides a replicable blueprint for developing similar educational tools in other agglutinative or low-resource languages, fostering broader AI-enhanced educational opportunities.

Chapter 7

Automated Generation of Arabic Educational Crosswords

7.1 Motivation and Scope

Crossword puzzles, when thoughtfully designed for pedagogical purposes, are powerful tools for reinforcing knowledge, enhancing vocabulary, and developing technical terminology recall. Their effectiveness spans multiple disciplines, including history, science, and linguistics, making them particularly apt for educational settings and language learning. By requiring active recall and accurate spelling, crosswords foster an interactive learning experience that promotes the retention of specialized jargon and improves critical thinking abilities.

Bringing this pedagogical value to the Arabic language at scale, however, presents significant challenges. Arabic, with its complex morphology, right-to-left (RTL) script, and orthographic nuances (such as diacritics), requires specialized computational approaches. Traditional automated generation methods often struggle with these complexities. Furthermore, there is a marked scarcity of advanced digital educational tools and datasets tailored specifically for Arabic, which hampers the development of reliable authoring systems.

The manual creation of high-quality Arabic crosswords is labor-intensive, demanding significant linguistic expertise and time. Existing automated tools often rely on generalized lexical databases and lack the ability to ground clues in specific instructional texts—a crucial requirement for ensuring that puzzles align with curriculum objectives and provide a verifiable learning experience.

This chapter addresses these needs by presenting a unified, end-to-end framework for the automated generation of Arabic educational crosswords. This framework harmonizes the methodologies and findings from two distinct research efforts to provide a robust and flexible solution. The system is designed to operate in two complementary modes, accommodating different educational scenarios:

1. **Text-Conditioned Generation (Path A):** Creating curriculum-aligned puzzles directly from instructional Arabic texts (e.g., textbook chapters or articles).

2. **Answer-Conditioned Generation (Path B):** Creating vocabulary or review puzzles from a list of teacher-provided keywords (answers).

The evolution of this framework represents a significant advancement. Initial work established the foundational architecture and explored the use of few-shot learning for

Path A and fine-tuning on a manually curated dataset for Path B. Subsequent research significantly enhanced Path A by introducing *Arabic-Clue-Instruct*, a large-scale, text-aligned dataset, and shifting the methodology from few-shot learning to a more robust instruction-tuning paradigm.
This chapter details the integrated framework, highlighting the specialized data resources developed, the advanced Large Language Model (LLM) driven methodologies for clue generation and validation, and the algorithmic approach to grid construction. The goal is to provide educators with a reliable tool to make learning Arabic more engaging and effective through gamification.

7.2 Framework Overview

The system is architected as a modular pipeline that integrates several specialized components to transform input materials (text or keywords) into a polished educational crossword puzzle. The architecture, illustrated in Figure 7.1, supports the two primary generation pathways (Path A and Path B), which converge on shared validation and schema generation modules.

7.2.1 Component Roles and Architecture

The framework abstracts functionality into distinct roles, allowing for flexibility and continuous improvement as new models become available.

- **Keyworder (Path A only):** Responsible for identifying educationally salient and crossword-suitable targets (answers) from an input instructional text. Initial implementations utilized few-shot prompting of LLMs for this task.
- **Generator:** The core component responsible for producing clues. It operates in two modes:
  - *Context-Grounded (Path A):* Generates clues based on a keyword and its surrounding context. This mode evolved from few-shot learning to utilizing specialized models instruction-tuned on the *Arabic-Clue-Instruct* dataset.
  - *Answer-Only (Path B):* Generates definition-style clues based solely on the answer string, utilizing models fine-tuned on a curated repository of Arabic crosswords.
- **Validator:** A critical quality assurance module that enforces constraints and ensures the pedagogical suitability of the generated clue–answer pairs. It employs a cascade of checks, including deterministic rules and learned acceptability classifiers.
- **Selector/User Revision:** Surfaces high-confidence pairs to the educator, allowing for manual review, selection, and the marking of "preferred answers" before grid generation.
- **Schema Generator:** An algorithmic layout engine that places the selected answers into a high-quality, dense, and interlocked grid under specified constraints.

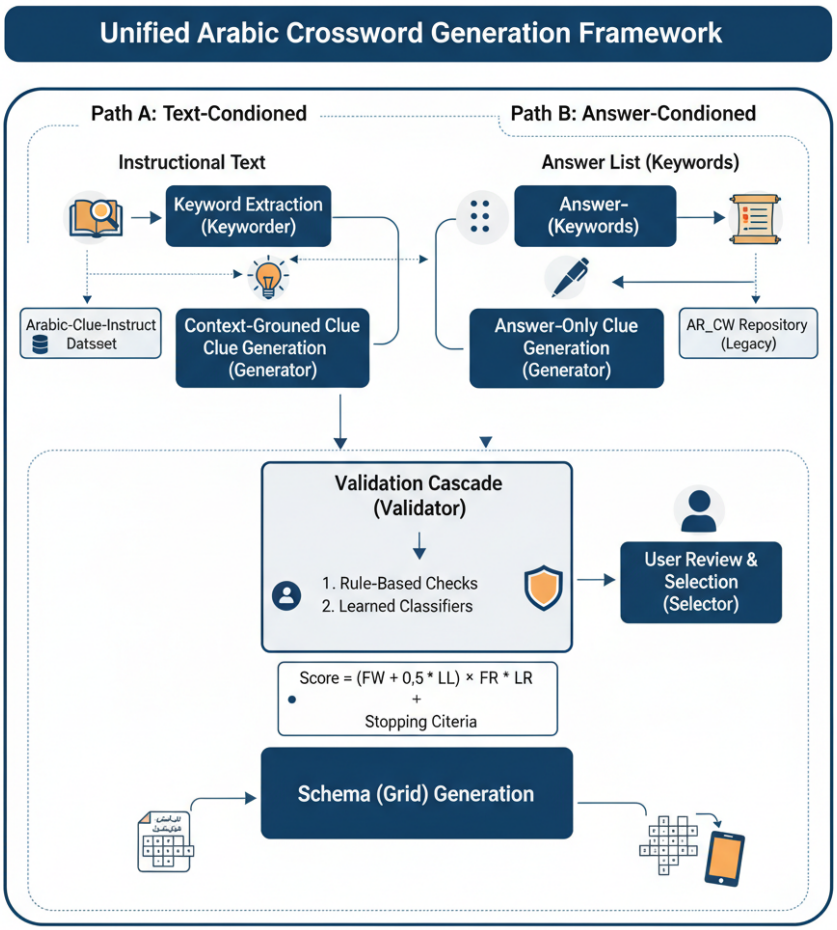


Figure 7.1: System architecture. Path (a): Text input leads to Keyword Extraction, followed by Clue Generation (using few-shot learning or instruction-tuned models), and Validation. Path (b): Keyword input leads to Clue Generation by fine-tuned models, followed by Validation. Both paths converge on User Revision and Schema Generation.

7.2.2 Data Flow and Pipeline Steps

The operational workflow differs depending on the available input material.

Path (a): Text-Conditioned Generation. This path ensures curriculum alignment and factual grounding.

1. **Text Segmentation and Keywording:** The input text is segmented, and the Keyworder extracts salient keywords.

2. **Context-Grounded Clue Generation:** For each keyword, the instruction-tuned Generator produces candidate clues entailed by the paragraph.

3. **Validation:** The Validator applies rule checks (e.g., leakage detection) and classifiers to ensure quality.

4. **Selection and Schema Generation:** Surviving pairs are selected and passed to the Schema Generator.

Path (b): Answer-Conditioned Generation. This path is utilized when only a list of solutions is available.

1. **Answer Normalization:** The input answer list is normalized (e.g., diacritic removal, length capping).

2. **Answer-Only Clue Generation:** The Generator (fine-tuned on the legacy repository) produces candidate clues for each answer.

3. **Validation:** The validation cascade is applied, relying heavily on the learned acceptability classifier to filter outputs. Thresholds are set for high precision.

4. **Selection and Schema Generation:** Selected pairs are passed to the Schema Generator.

7.3 Foundational Data Resources for Arabic

The efficacy of this framework is underpinned by two complementary, large-scale Arabic data resources. The scarcity of existing digital resources necessitated the development of both a repository of traditional clue–answer pairs and a novel text-aligned corpus for instruction tuning.

7.3.1 The Legacy Arabic Crossword Pair Repository

To support Path B and the training models, a comprehensive repository of existing Arabic crossword clue-answer pairs was compiled. This required a significant collection and curation effort.

Data Collection and Curation

The data collection process involved sourcing puzzles from various accessible mediums, including web-based games, journals, and magazines, spanning 2020 to 2023.
The collection involved two primary sources and methods:

- **Manual Extraction:** For sources like the *Al-Joumhouria Journal*, 5,661 clue-answer pairs were manually extracted and transcribed.

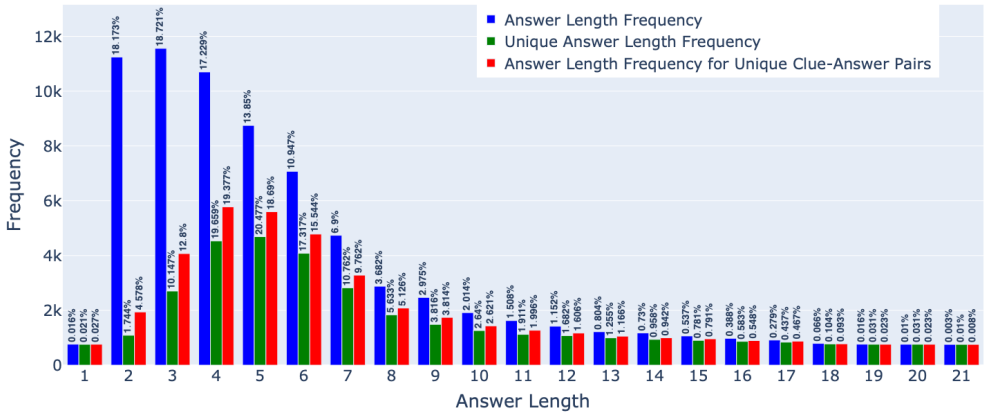


Figure 7.2: Answer-length distribution in the curated legacy Arabic crossword repository (AR_CW). Blue bars represent all entries, green bars represent unique answers, and red bars represent unique clue-answer pairs.

- **OCR-Assisted Extraction:** For digital images or scanned documents, such as those from the *Al-Ghad Electronic Journal*, Optical Character Recognition (OCR) tools were utilized.

Crucially, the extraction process was heavily supervised. Human validation was essential to correct OCR errors, verify spelling within the original sources, and assess the overall quality of the clue-answer pairs.

Preprocessing and Normalization

The raw collected data required meticulous preprocessing for model training:

- **Diacritic Removal:** Elimination of Arabic accents (diacritics) to normalize the text.
- **De-duplication:** Removal of redundant pairs.
- **Cleanup of Reversal Markers:** Removal of markers indicating a reversal of the answer, an idiosyncrasy sometimes found in Arabic crosswords but unsuitable for standardized formats.

Dataset Statistics and Structure

The final curated repository comprises 57,706 entries, yielding 25,908 unique clue-answer pairs. The dataset exhibits a variety of clue styles, including synonyms/antonyms, general information, anagrams/scattered letters, and partial words.
The distribution of answer lengths is visualized in Figure 7.2. Answers vary from 1 to 21 characters, with the vast majority falling within the 2 to 9 character range, informing the typical characteristics of Arabic crossword grids.

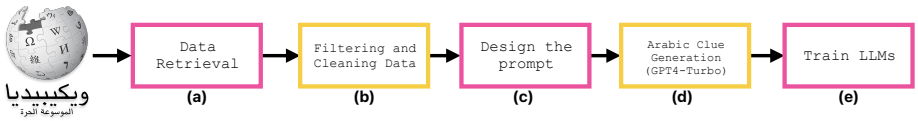


Figure 7.3: Methodology overview for creating Arabic-Clue-Instruct: (a) Data Retrieval from Arabic Wikipedia; (b) Filtering and Cleaning Data; (c) Prompt Design; (d) Arabic Clue Generation using GPT-4 Turbo (Teacher Model); (e) Training LLMs (Student Models).

7.3.2 Arabic-Clue-Instruct: A Text-Aligned, Category-Aware Corpus

While the legacy repository is valuable for Path B, it lacks the crucial element needed for Path A: a direct link between the clue and a source instructional text. To enable high-fidelity, context-grounded generation, the *Arabic-Clue-Instruct* dataset was created . This is the first dataset of its kind for Arabic, specifically designed to instruction-tune LLMs.

Dataset Construction Methodology

The dataset was constructed using a semi-automated pipeline, leveraging a high-capability "Teacher" LLM (GPT-4 Turbo) to synthesize structured training data. The methodology is visualized in Figure 7.3.

1. Data Acquisition. The process began by extracting the initial sections of approximately 211,000 Arabic Wikipedia articles, focusing on the prominently bolded keywords in the introduction, which typically correspond to the article's main focus.

2. Data Filtering and Refinement. A stringent filtration process was applied to ensure data quality:

- **Content Length:** Articles with fewer than 50 words were discarded to ensure sufficient context.
- **Keyword Constraints:** Keywords longer than two words, outside the 3 to 20 character limit, or containing special characters/numerals were excluded.

This refinement process reduced the dataset to 14,497 high-quality context-keyword pairs spanning 20 distinct thematic categories.

3. Prompt Engineering. A specialized prompt (Figure 7.4) was meticulously designed to guide the Teacher LLM (GPT-4 Turbo). The prompt provides the keyword, category, and context, and includes explicit instructions to generate concise (MAX 4 WORDS),

```
You are an Arabic Crossword Puzzle Author expert. your expertise involves taking any paragraph and generate
clues based on the Keywords and the Category. your task is to generate a clever and accurate clues which uses
wordplay, metaphors, puns, and other creative techniques to make the clues engaging and thought-provoking.This
task is designed for crossword puzzle authors looking to engage their audience with entertainment clues.

based on a given keywords, a category and a keyword-related context.
                    KEYWORDS: {keyword}
                    CATEGORY: {category}
                    CONTEXT: {text}
Follow these steps:
1. Analyze the context to identify the most relevant and informative connections between '{keyword}' and
'{category}'. Extract key facts from  the context, considering the length and complexity.

2. Create {no} clues from these keyfacts in Arabic (MAX 4 WORDS) that would be clue for the {keyword}, making
sure not to include the keyword in the clues.

3. '{keyword}' should be the answer to the clue. Avoid using the '{keyword}' or its derivatives directly in the
clues. Instead, use descriptive phrases, synonyms, or creative wordplay that hints at '{keyword}' without
revealing it outright.

4. Provide clues in the following structure: Clues: <clue1> , <clue2> , <clue3> ...
```

Figure 7.4: Illustrative prompt used with the Teacher LLM (GPT-4 Turbo) to elicit short, non-leaking, educational Arabic clues for the Arabic-Clue-Instruct dataset.

factually accurate clues based on the text, while strictly avoiding leakage of the keyword or its derivatives.

4. Clue Generation (Teacher Model). Using the curated inputs and the specialized prompt, GPT-4 Turbo generated at least three distinct clues for each entry, resulting in a total of 54,196 generated clues.

Dataset Statistics and Distribution

The final *Arabic-Clue-Instruct* dataset contains **14,497** content–keyword pairs spanning **20** distinct categories, for which we generated a total of **54,196** clues.
The distributions of context, keyword, and clue lengths are shown in Figure 7.5. Most contexts fall between 40 and 200 words; keywords are primarily between 5 and 15 characters; and clues typically center around 20 to 30 characters.

The topical distribution (Figure 7.6) shows broad coverage, with "Geography," "Science," and "Society" being the most represented categories.

Quality Validation of Arabic-Clue-Instruct

To ensure the reliability of the synthetic dataset, its quality was rigorously assessed.

Automatic Metrics . ROUGE [125] scores were used as a supportive metric to gauge the extractive overlap between the generated clues and the input context. For text vs. GPT-4 Turbo clues, the mean scores were **ROUGE-1** = 0.0281, **ROUGE-2** = 0.0055, and **ROUGE-L** = 0.0278. These relatively low values are expected, since clues are paraphrases rather than direct extractions, yet they indicate a measurable connection to the source text.

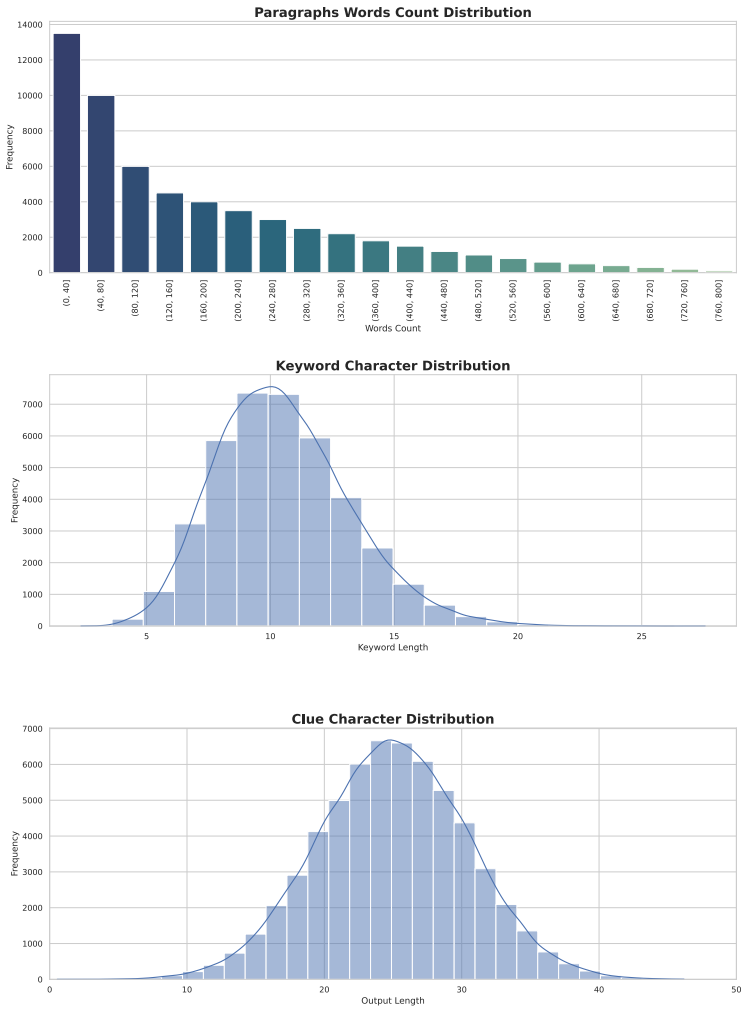


Figure 7.5: Distributions of context word counts (top), keyword character lengths (middle), and clue character lengths (bottom) in the Arabic-Clue-Instruct corpus.

Human Evaluation. A more informative assessment was conducted by native Arabic speakers on a subset of approximately 600 clues using a five-level rating scale (A–E), evaluating coherence, validity, contextual alignment, and leakage. **Rating definitions**:

- **A** (coherent & valid): Clue is correct and well-formed, aligning with the given context, answer, and category.
- **B** (minor issues): Generally acceptable, but shows slight issues—often a weaker link to the specified category.

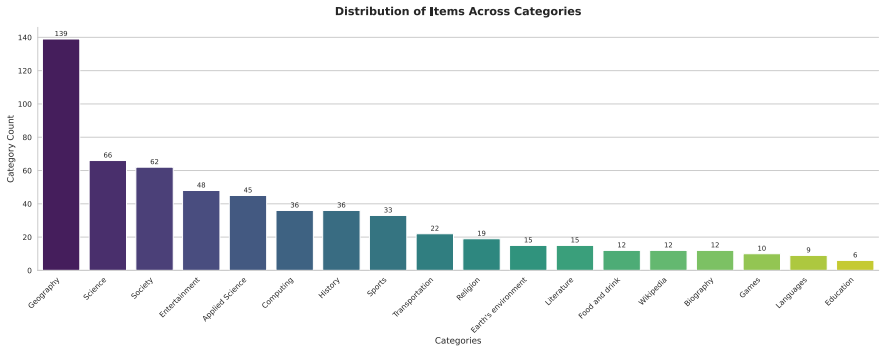


Figure 7.6: Category coverage (20 classes) for the Arabic-Clue-Instruct corpus.

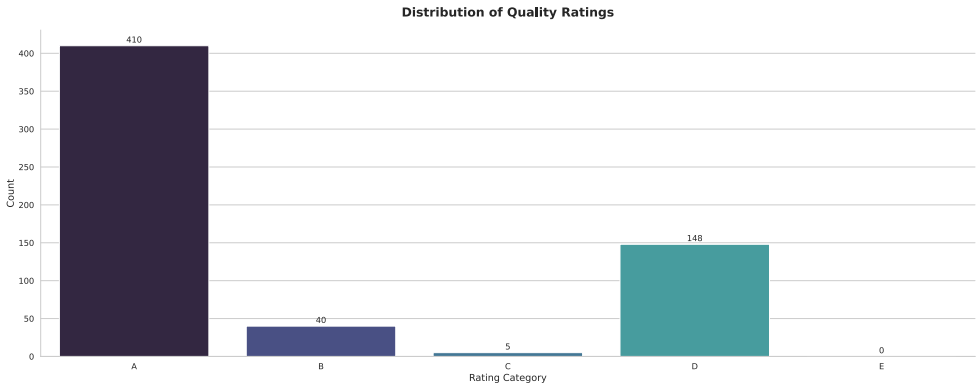


Figure 7.7: Counts of human ratings (A–E) for a sampled set of clues within the Arabic-Clue-Instruct dataset, demonstrating high overall quality.

- **C** (vague/broad): Relates to the answer, but the connection to the context is vague or overly broad.
- **D** (incorrect/irrelevant): Irrelevant or incorrect with respect to the answer or context.
- **E** (leakage): Unacceptable because it directly contains the answer or a close variant of it.

The results (Figure 7.7) confirm the high quality of the dataset. Specifically, the distribution across the five levels was: A 67.55%, B 6.59%, C 1.15%, D 24.55%, and E 0.16%. Over 67.5% of the clues generated by the Teacher model were rated 'A' (highest quality), and 74.14% were rated 'A' or 'B' (acceptable), validating that *Arabic-Clue-Instruct* offers a strong foundation for training student models.

7.4 Methodology: From Arabic Text to Puzzle

The framework's methodology integrates several specialized components to transform input (text or answers) into a final crossword grid. This involves distinct generation strategies for Path A and Path B, a unified validation cascade, and an optimized schema generator.

7.4.1 Path (a): Text-Conditioned Generation

Path A involves extracting suitable keywords from the text and then generating contextually grounded clues. The methodology evolved from initial reliance on few-shot learning to a more sophisticated instruction-tuning approach.

Keyword Extraction

The **Keyworder** component identifies educationally salient targets. This is typically implemented using few-shot learning with a strong LLM (e.g., GPT-4) . A prompt is constructed with examples of educational text and desired keywords. The LLM then uses these examples to extract potential keywords from the target input text that adhere to constraints (e.g., max two tokens).

Clue Generation: Evolution to Instruction Tuning

The **Generator** component for Path A leverages the *Arabic-Clue-Instruct* dataset to specialize "student" LLMs . This approach supersedes earlier methods that relied on few-shot prompting of general-purpose models , which were sometimes inconsistent or prone to generating clues based on innate knowledge rather than the provided text.
We fine-tune both open-source (Llama3-8B-Instruct) and proprietary (GPT-3.5-Turbo) models. The training process aligns these models to generate concise Arabic definitions ($\leq$ 4 words) that are strictly entailed by the provided paragraph without leaking the answer string. This instruction-tuning approach significantly enhances the reliability and quality of the generated clues.

7.4.2 Path (b): Answer-Conditioned Clue Generation

When only answers are available, the **Generator** component for Path B is utilized. This generator is fine-tuned on the legacy clue answer pair repository . This training exposes the models to the diverse styles of human-authored Arabic crosswords.
Several models of varying sizes were fine-tuned, including GPT-3 variants (Davinci, Curie) and GPT-3.5 Turbo. The fine-tuning process involves providing the model with an answer and training it to generate a corresponding, plausible clue, enabling the system to produce high-quality definitions without an accompanying source text.

7.4.3 The Validation Cascade

A critical aspect of the framework is the multi-stage **Validator** component, which ensures the quality of the generated pairs from both paths. The cascade is designed to maximize precision.

Rule-Based Checks

The first stage involves deterministic heuristics to filter out common errors:

- **Answer Length Constraints:** Rejecting answers composed of more than three words.
- **Leakage Prevention:** Strictly forbidding clues that contain the answer string or its direct morphological variants.
- **Basic Constraints:** Checking for length and basic format constraints.

Learned Acceptability Classifiers

The second stage employs supervised binary classifiers to identify nuanced quality issues (e.g., ambiguity, factual errors) . These classifiers are trained on a dataset of 6,000 human-labeled instances derived from the legacy corpus, balanced between acceptable and unacceptable examples.
Several models were trained as classifiers, including GPT-3 Davinci, Curie, Babbage, Ada, and BERT-base-Arabic. These models predict an accept/reject label for each generated pair, providing an automated quality gate.

Zero-Shot Validation (Legacy Path A)

In earlier iterations of Path A , an additional validation step used zero-shot learning, where an LLM was prompted to verify if the generated clue was factually supported by the provided text. This helped mitigate hallucinations when using few-shot generation methods.

7.4.4 Schema (Grid) Generation

The final stage is the **Schema Generator**, which assembles the validated pairs into a printable grid. The engine performs a randomized constructive search over possible placements .

7.4.5 Shared Component: Schema Generation

The schema-generation phase reuses the algorithm from the English pipeline (Section 4.4.3). A randomized constructive search positions validated clue–answer pairs into a tight, strongly interlocked grid. A multi-objective scoring function guides the process to maximize filled entries, increase crossing density, and improve the overall fill ratio. The outcome is a polished, solvable puzzle well suited for educational settings.

7.5 Experimental Validation

We report a consolidated view of the experimental evidence across both pathways, detailing the experimental setup, the metrics used, and the key results derived from the foundational studies.

7.5.1 Experimental Setup and Implementation Details

The experiments involved fine-tuning several LLMs for both generation and validation tasks.

Path A: Instruction Tuning Setup

For the text-conditioned generators, the *Arabic-Clue-Instruct* dataset was split into 14,000 training pairs and 497 testing pairs .

- **GPT-3.5-Turbo:** Fine-tuned with a batch size of 16 and a learning rate of 0.01 across three epochs.
- **Llama3-8B-Instruct [126]:** Fine-tuned using Low-Rank Adaptation (LoRA) [121] for parameter efficiency. Parameters were set to $r = 32$ (rank) and $\alpha = 64$, over three epochs, with a total batch size of 128 and an initial learning rate of 3×10^{-4}.

The training utilized a server equipped with dual NVIDIA A6000 GPUs, leveraging DeepSpeed and FlashAttention-2 technologies for acceleration. For inference, conservative sampling parameters were used (temperature 0.1, top-p 0.95, top-k 50).

Path B: Fine-Tuning and Classifier Training

For Path B generators, models were fine-tuned on the clue answer pair repository . The validation classifiers were trained on the 6,000 human-labeled acceptability examples, using an 80/20 train/test split.

7.5.2 Evaluation Metrics

The evaluation employs both automatic metrics and human assessment. ROUGE scores are used as a supportive automatic metric for Path A to measure similarity to reference clues. Human evaluation by native Arabic speakers, using the 5-level rating scale (A-E) defined in Section 7.3.2, is the primary measure of quality, assessing coherence, validity, contextual alignment, and the absence of leakage.

7.5.3 Validation of Text-Conditioned Generation (Path A)

The evaluation of Path A focused on assessing the impact of instruction-tuning on the models' ability to generate high-quality, text-grounded clues .

Table 7.1: Mean ROUGE scores vs. GPT-4 Turbo reference clues across model families. Fine-tuning significantly improves similarity.

Model Type	Name	ROUGE-1	ROUGE-2	ROUGE-L
Base	GPT-3.5	0.0152	0.0011	0.0148
Base	LLaMA3-8B	0.0063	0.0009	0.0063
Fine-tuned	GPT-3.5	0.0405	0.0045	0.0405
Fine-tuned	LLaMA3-8B	0.0354	0.0030	0.0354

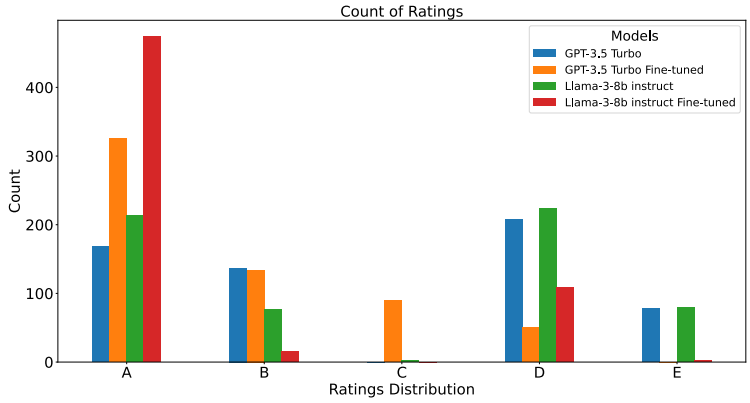


Figure 7.8: Human ratings by model (base vs. fine-tuned) on text-conditioned Arabic clues. Fine-tuning drastically increases A-ratings and reduces D/E ratings.

Automatic Metric Results

Table 7.1 compares the ROUGE scores of base and fine-tuned models against the GPT-4 Turbo reference clues on the test set.

The results demonstrate that fine-tuning significantly increases the similarity to the high-quality reference clues for both GPT-3.5 and Llama3-8B. The base models exhibit minimal overlap, highlighting the necessity of instruction-tuning for this specialized task.

Human Evaluation Results

A human evaluation was conducted on 200 Arabic contexts (600 clues) generated by the base and fine-tuned models. The results, illustrated in Figure 7.8, show a dramatic improvement in quality after fine-tuning.

The fine-tuned Llama3-8B model emerged as the top performer, achieving a remarkable 78.86% A-rating (highest quality), a substantial increase from the base model's 36.02%. Crucially, fine-tuning drastically reduced the proportion of low-quality clues (D and E ratings) for both models. This confirms that instruction-tuning on *Arabic-Clue-Instruct* successfully specializes LLMs for generating high-quality Arabic educational clues.

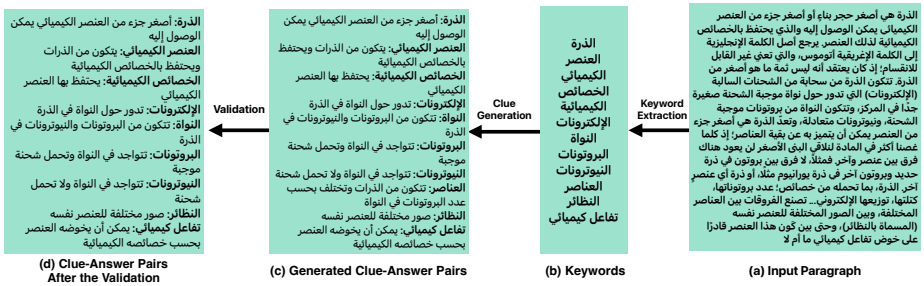


Figure 7.9: Worked example (text-conditioned): (a) input paragraph, (b) extracted keywords, (c) generated clues, and (d) validated pairs.

Table 7.2: Assessment outcomes of the few-shot learning approach for Path A .

Model	*System Part*	*English Prompt*	*Arabic Prompt*
GPT-4	Keyword Extractor	95.05%	94.32%
	Clues Generator	94.62%	93.23%
	Validator (Zero-shot)	87.76%	84.01%
	Total performance	78.95%	74.60%
GPT-3.5-Turbo	Keyword Extractor	92.00%	97.38%
	Clues Generator	55.33%	37.78%
	Validator (Zero-shot)	89.04%	89.32%
	Total performance	46.68%	68.83%

A worked example demonstrating the end-to-end process of Path A is shown in Figure 7.9.

Few-Shot Learning Evaluation (Legacy Approach)

In earlier explorations of Path A , few-shot learning was used instead of instruction-tuning. This approach evaluated GPT-4 and GPT-3.5 Turbo using both English and Arabic prompts on 100 Wikipedia paragraphs (Table 7.2).

Keyword extraction was highly accurate. However, clue generation quality varied; GPT-4 achieved over 93% accuracy, while GPT-3.5 Turbo struggled (37-55%). Overall, GPT-4 demonstrated strong end-to-end performance (78.95%). While effective for high-capability models, the instruction-tuning approach provides a more robust solution, especially for smaller, open-source models.

7.5.4 Validation of Answer-Conditioned Generation (Path B)

The evaluation of Path B focused on the ability of models fine-tuned on the legacy corpus to generate acceptable clues from keywords alone . Three models were evaluated by generating clues for 2,000 keywords, which were then assessed by human judges (Table 7.3).

The fine-tuned GPT-3.5 Turbo [122] significantly outperformed the others, achieving

Table 7.3: Acceptance rates for answer-conditioned generation (human-rated) .

Model	*% acceptable clues*
GPT-3 Davinci	41.90
GPT-3 Curie	21.35
GPT-3.5 Turbo	81.00

Table 7.4: Classifier performance on distinguishing acceptable clue–answer pairs .

Model	*Accuracy %*	*Precision %*	*Recall %*	*F1*
GPT-3 Davinci	85.74	83.39	85.26	0.8431
GPT-3 Curie	81.29	78.86	79.89	0.7937
GPT-3 Babbage	78.69	75.17	78.54	0.7682
GPT-3 Ada	79.19	77.48	75.75	0.7660
BERT-base-Arabic	71.42	67.91	70.04	0.6896

a high acceptance rate of 81.00%. This demonstrates that Path B is effective when high-quality fine-tuning data and a capable model are available.

7.5.5 Validator Performance

The performance of the learned acceptability classifiers was evaluated on the held-out test set of the 6,000 labeled examples . The results are summarized in Table 7.4.

The GPT-3 Davinci classifier achieved the highest performance (Accuracy 85.74%, F1 0.8431). All GPT-3 variants significantly outperformed the BERT baseline. These results demonstrate that learned classifiers provide a reliable automated quality gate, reducing editorial burden.

7.5.6 Schema Generation and End-to-End Output

The schema generator successfully produces dense, interlocking layouts under the defined constraints. The randomized constructive search algorithm reliably balances the fill ratio and linked letters, resulting in high-quality, printable grids.
Illustrative examples of complete Arabic educational crosswords produced by the end-to-end system are shown in Figures 7.10 and 7.11, demonstrating the successful integration of generation, validation, and grid assembly.

7.6 Conclusions

This chapter presented a unified and comprehensive framework for the automated generation of Arabic educational crosswords. By integrating two complementary pathways—text-conditioned (Path A) and answer-conditioned (Path B)—the framework ensures robustness and flexibility across diverse educational scenarios.
The foundation of this work lies in the creation of two critical data resources: the legacy

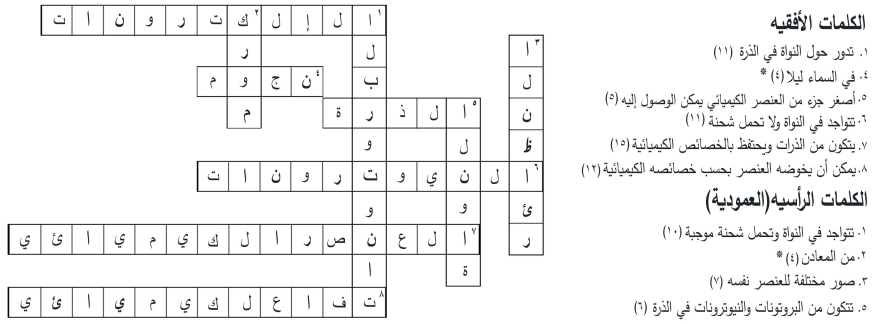


Figure 7.10: Illustrative Arabic educational crossword (Physics topic) produced by the layout engine, combining pairs from Path A and Path B.

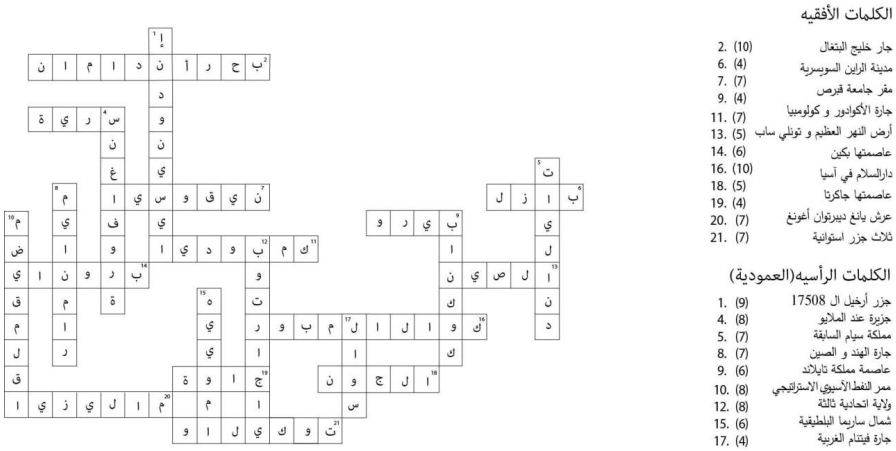


Figure 7.11: Another example crossword crafted by the system (Geography topic).

clue answer pair repository, curated through extensive manual effort and OCR, and the novel *Arabic-Clue-Instruct* dataset, synthetically generated using a teacher-student methodology. These datasets enabled the development of specialized generators and validators tailored to the nuances of the Arabic language.

Empirical validation demonstrates the effectiveness of the approach. For text-conditioned generation, instruction-tuning open-source models like Llama3-8B [126] on *Arabic-Clue-Instruct* yields significant improvements over base models, producing concise, faithful, and non-leaking clues with high human acceptance rates (nearly 79% A-rated). For answer-conditioned generation, fine-tuned models achieve strong performance (81% acceptance rate), providing a viable option when source texts are unavailable.

The framework's modular design, incorporating a precision-first validation cascade (with classifiers achieving up to 85.7% accuracy) and an optimized schema generator, ensures

the production of high-quality, classroom-ready educational tools. This work significantly advances the state-of-the-art in Arabic educational technology, providing a scalable and reliable method for leveraging AI to enhance learning through engaging, interactive puzzles.

Part IV

Multilingual Question Generation

Chapter 8

Automating Turkish Educational Quiz Generation with Large Language Models

8.1 Motivation and Scope

This chapter proposes an end-to-end methodology for automatically generating educational quizzes from Turkish texts, built on the *From Language to Learning* framework introduced in this thesis. The goal is to provide a scalable blueprint for automated assessment in a low-resource, morphologically rich language.

Quizzes underpin effective teaching and learning: they enable retrieval practice (improving long-term retention), deliver timely feedback, and help instructors diagnose misconceptions. Yet producing high-quality items—especially Multiple-Choice Questions (MCQs) with *plausible distractors*—is time-consuming and demands both subject expertise and careful item design. Automating this process with modern NLP promises faster iteration and broader access to customized materials.

Research on Automated Question Generation (QG) has focused heavily on English, leaving a digital divide. Low-resource settings face: (i) *data scarcity*—limited corpora pairing texts with assessment items; (ii) *resource limitations*—few specialized tools, benchmarks, or tailored pre-trained models; and (iii) *linguistic complexity* that general, Anglocentric models inadequately capture.

Turkish intensifies these challenges through (i) *agglutinative morphology* with productive suffixation, complicating lemmatization and grammatical generation; (ii) *vowel harmony*, which constrains suffix forms; and (iii) *flexible constituent order* (often SOV but reorderable for emphasis), raising ambiguity and clarity issues in generated items. Effective distractor generation further requires sensitivity to these morpho-syntactic interactions.

Given an arbitrary Turkish educational passage, the system automatically produces:

1. **MCQs**: a well-formed stem, one correct answer (key), and several linguistically and semantically *plausible yet incorrect* distractors;

2. **SAQs**: a concise, unambiguous question with a gold answer.

Overall, the contribution is a practical, reproducible pipeline for Turkish QG that addresses the automation bottleneck while illuminating the path for other low-resource, morphologically rich languages.

8.2 Methodology: The "From Language to Learning" Framework Applied to Turkish QG

To address the challenge of automating Turkish educational quiz generation, we apply the unified "From Language to Learning" framework central to this thesis. This methodology leverages a systematic two-stage distillation process to transform powerful but general-purpose LLMs into specialized, efficient tools capable of generating high-quality MCQs and SAQs tailored to the Turkish educational context. This section details the application of this framework, covering the curation of Turkish educational content, the sophisticated process of synthesizing the instructional dataset, and the subsequent fine-tuning of the student models.

8.2.1 Architectural Overview

The methodology follows a structured, multi-step pipeline, as illustrated in Figure 8.1. It commences with the rigorous collection and preparation of raw educational content, which forms the input for the two core stages of the distillation process: synthetic data generation by a teacher model and knowledge transfer to student models.

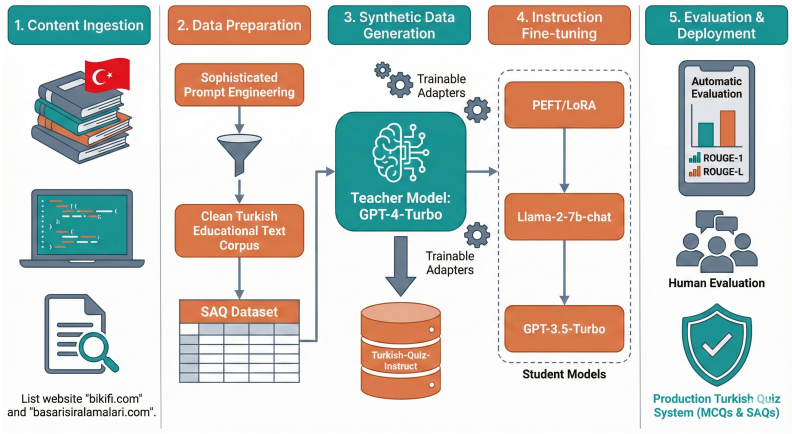


Figure 8.1: End-to-end methodology: (a) collection of Turkish educational content; (b) data cleaning and filtering; (c) crafting specialized prompts; (d) synthesis of the seed dataset (Turkish-Quiz-Instruct) using a teacher LLM; (e) fine-tuning student LLMs for reliable Turkish quiz generation.

8.2.2 Data Generation (Creating Turkish-Quiz-Instruct)

The first and most critical stage involves the creation of a high-quality seed dataset, *Turkish-Quiz-Instruct*. This process relies on leveraging the advanced capabilities of a powerful "teacher" LLM (GPT-4-Turbo) through meticulous data preparation and sophisticated prompt engineering.

The foundation of the dataset is a comprehensive collection of Turkish educational texts spanning a diverse range of disciplines, ensuring broad applicability of the resulting models. The disciplines included are Chemistry, Biology, Geography, Philosophy, Turkish Literature, and History.

Data Scraping and Source Curation. The information extraction process specifically targeted reputable Turkish websites dedicated to disseminating educational content aligned with secondary school curricula. The primary sources utilized were `https://bikifi.com/` and `https://basarisiralamalari.com/`. These platforms offer comprehensive summaries of key concepts and topics that adhere to officially sanctioned educational standards. Utilizing such curated sources ensures the curricular relevance and factual accuracy of the base material.

Data Cleaning and Filtering. Raw scraped data invariably contains noise that can degrade the quality of the generated quizzes. To ensure the integrity of the dataset, a series of meticulous data-cleaning procedures were implemented (Figure 8.1b). This involved the systematic removal of extraneous elements such as navigation bars, advertisements, emojis, hyperlinks, and irrelevant markup. Furthermore, a length-based filtering strategy was applied: excessively short text fragments were eliminated to ensure that the remaining passages provided sufficient context for generating meaningful, non-trivial questions. This cleansing process mitigated noise and enhanced the overall coherence and reliability of the data.

Prompt Engineering for Pedagogical Constraints The formulation of targeted prompts was a critical step for eliciting well-formed MCQs and SAQs from the teacher model (Figure 8.1c). Standard prompting techniques often fail to produce outputs that adhere to the strict requirements of educational assessments. Therefore, the prompts were meticulously crafted to guide the generation process, ensuring the resulting quizzes were informative, engaging, and, crucially, pedagogically sound. A structured prompt template was designed to enforce several critical constraints simultaneously: clarity, key uniqueness, plausible distractors, and strict content alignment. This template, shown in Figure 8.2, explicitly instructs the model on the required components of an MCQ (stem, correct answer, distractors) and outlines specific rules governing their creation:

1. **Clarity and Grammaticality:** The stem must be clear, concise, and grammatically correct in Turkish, adhering to morphological rules.

2. **Unambiguous Key:** The correct answer must be indisputably correct based on the provided text and directly address the question posed in the stem.

3. **Plausible Distractors:** Distractors must be plausible yet definitively incorrect. They should be related to the topic to test the student's understanding but should not be trivially wrong or inadvertently reveal the correct answer (e.g., through grammatical mismatches).

4. **Content Grounding:** The entire quiz (stem, key, and distractors) must be strictly based on the information present in the provided educational text, minimizing the risk of hallucination.

5. **Conciseness and Format:** Avoid overly long sentences and adhere to the specified JSON output structure for automated processing.

```
A multiple-choice quiz consists of three main components: a stem, a correct answer, and two distractors. The stem is the question or
incomplete statement that the student must answer or complete, while the correct answer is the one and only correct response to the stem.
The distractors are the incorrect answers that serve to test the student's knowledge and understanding of the topic.

Your objective is to extract {count} distinct topics from this Turkish text and generate Turkish questions with multiple answer choices
for each topic. Each question must be different and should not overlap with each other.

To create effective multiple-choice quizzes, keep the following guidelines in mind:

1. Make sure the stem is clear, concise, and grammatically correct in Turkish. It should be written in a way that is easy for students to
understand and should not be overly complex or ambiguous.
2. The correct answer should be indisputably correct and directly related to the stem. It should not be too obvious or too difficult to
guess.
3. The distractors should be plausible but incorrect answers that are related to the topic and the stem. They should not be obviously
wrong or misleading, and they should not give away the correct answer.
4. Use "Hiçbiri" (None of the above) and "Hepsi" (All of the above) options sparingly.
5. Make sure the quiz is based on the educational text provided and that it tests the student's knowledge and understanding of the key
concepts and ideas presented in the text.
6. Avoid using overly long sentences in the stem, correct answer, or distractors. Keep the quiz concise and to the point.
7. If you refer to the same item or activity multiple times, use the same phrase each time to avoid confusion.
8. Ensure that each multiple-choice question provides full context and that all necessary information is included in the stem.
9. Ensure that the distractors are unique and do not overlap with each other.
10. Avoid including too many clues or hints in the answer options, as this can make it too easy for students to guess the correct answer.

            Subject: {subject}
            Topic: {topic}
            Context: {content}

            Your return should be the exact json structure of the following example:

           {"question": "The stem of the question",
            "choices": [{"option": "A", "text": "Answer Choice A in string type"}, {"option": "B", "text": "Answer Choice B in string
type"},
            {"option": "C", "text": "Answer Choice C in string type"}],
            "correctAnswer": "A"}
```

Figure 8.2: The structured prompt template used with the teacher LLM (GPT-4-Turbo) for Turkish MCQ generation, enforcing pedagogical constraints and output format.

This rigorous prompt design is essential for managing the complexities of Turkish. For example, ensuring distractors are grammatically parallel to the correct answer requires careful attention to suffixation and vowel harmony, which the prompt implicitly encourages by demanding high-quality Turkish output.

Seed Dataset Synthesis

Leveraging the curated content and the engineered prompts, the teacher model (GPT-4-Turbo) was employed to synthesize the *Turkish-Quiz-Instruct* dataset. This process, rooted in the principles of the SELF-INSTRUCT framework, facilitated the large-scale generation of structured supervision data. GPT-4-Turbo was selected as the teacher model due to its superior performance in natural language understanding and generation, and its demonstrated proficiency in handling the morphological and syntactic complexities of the Turkish language.
The synthesis resulted in a comprehensive dataset where each educational passage is paired with corresponding MCQs. To create the SAQ portion of the dataset, a derivation process was applied: the correct answer was extracted from the generated MCQs, and the question-answer pair was reformulated into a concise short-answer format. The resulting quality and characteristics of this dataset are detailed in Section .

8.2.3 Instruction Fine-Tuning and Knowledge Distillation

In the second stage, the synthetic *Turkish-Quiz-Instruct* dataset serves as the training corpus to fine-tune smaller, more accessible "student" models . This knowledge distillation process transfers the specialized quiz generation capabilities of the teacher model to the student models, significantly enhancing their ability to produce contextually relevant and

linguistically precise Turkish quizzes efficiently.
Model Selection We selected several state-of-the-art LLMs known for their multilingual capabilities and effectiveness in instruction-following tasks:

- **Open-Source Models:** Llama-2-7b-chat-hf and Llama-2-13b-chat-hf. These models were chosen as they provide accessible, transparent, and efficient alternatives to proprietary models, crucial for democratizing educational technology.
- **API Model:** GPT-3.5-Turbo. This model was included to serve as a high-performance benchmark and to assess the benefits of fine-tuning even for already capable models.

Fine-Tuning Methodology The fine-tuning process aimed to adapt these base models specifically for the tasks of MCQ and SAQ generation from Turkish educational texts. The dataset was partitioned into a training set (comprising 8,000 passages) and an evaluation set (comprising 260 passages).
Given the substantial computational resources required to fully fine-tune large models, we employed Low-Rank Adaptation (LoRA) [121] for the Llama-2 models [123] . LoRA [121] is a PEFT [120] technique that freezes the pre-trained model weights and injects trainable, low-rank matrices into the layers of the Transformer architecture (typically the attention layers). This approach drastically reduces the number of trainable parameters—often by a factor of thousands—while achieving strong task-specific performance comparable to full fine-tuning. This efficiency is critical for making the framework sustainable and adaptable.
Training Configuration. The fine-tuning configurations were optimized for each model family:

- **Llama-2 Models (7B and 13B):** Utilized LoRA with a rank (r) of 16 and an alpha (α) of 32. The training was conducted with a unified batch size of 64 and a learning rate of 0.0001 across three epochs.
- **GPT-3.5-Turbo:** Fine-tuned using the available API with a batch size of 16 and a learning rate of 0.001 over three epochs.

This meticulous fine-tuning strategy ensured that the student models were effectively specialized for the nuances of Turkish educational quiz generation. As the results demonstrate, this process significantly advanced their capabilities beyond their base versions, transforming them from unreliable generators into effective educational tools.

8.3 The Turkish-Quiz-Instruct Dataset

The cornerstone of this research is the *Turkish-Quiz-Instruct* dataset, a novel and comprehensive collection of Turkish educational texts paired with meticulously crafted MCQs and SAQs. This dataset represents a significant contribution to the field, addressing the critical scarcity of resources for Turkish educational technology and providing a robust foundation for the development and evaluation of QG models.

8.3.1 Dataset Characteristics and Statistics

The dataset encompasses a wide array of educational materials curated for secondary education levels, spanning six primary disciplines: Chemistry, Biology, Geography, Philosophy, Turkish Literature, and History.

Subject Distribution The distribution of subjects within the corpus is illustrated in Figure 8.3. The dataset exhibits a relatively balanced representation across the core disciplines. History constitutes the largest segment, followed closely by Chemistry, Geography, Literature, and Biology. Philosophy, while important for critical thinking skills, has a smaller representation in the curated sources. This diversity is intentional and crucial for training models that can generalize across different domains, discourse styles, and specialized terminologies. **Token Distribution and Capacity Planning** Under-

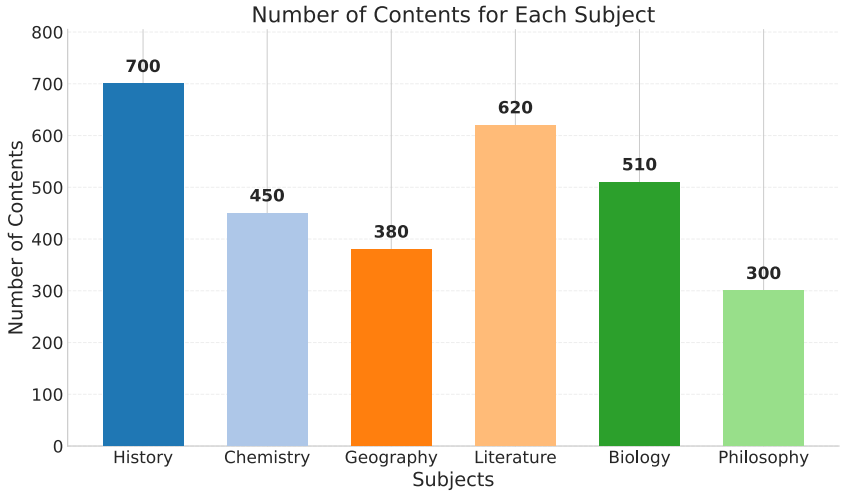


Figure 8.3: Subject distribution across the Turkish-Quiz-Instruct corpus.

standing the length distributions of the source passages and the generated quizzes is essential for optimizing model training, particularly for determining appropriate context window sizes and efficient batching strategies. Figure 8.4 visualizes the token distributions (calculated using the fast Llama-2 tokenizer) for both the educational contexts and the generated MCQs. The source passages exhibit a wide distribution, reflecting the varied nature of the educational content. The distribution peaks around 200-300 tokens, but there is a long tail extending up to several thousand tokens. This necessitates the use of models capable of handling long contexts or the implementation of effective truncation or chunking strategies. In contrast, the generated MCQs are significantly more concise, typically ranging from 50 to 200 tokens. This confirms the dataset's breadth and provides the necessary empirical data for capacity planning during both the fine-tuning and inference stages.

Quality Validation of the Seed Data The efficacy of the "From Language to Learning" framework hinges on the quality of the synthetically generated dataset, as it serves

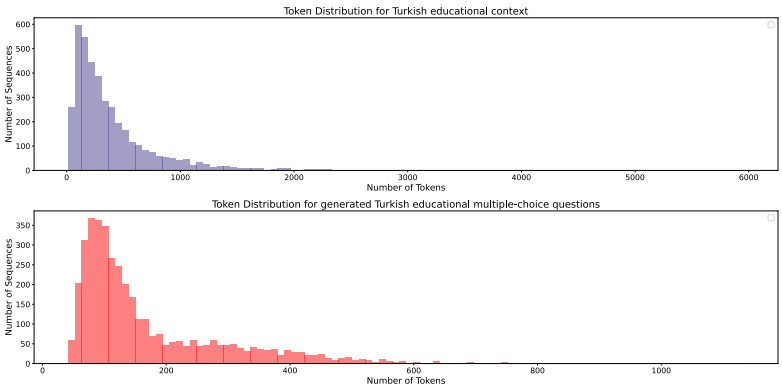


Figure 8.4: Token distributions for source passages (top) and generated MCQs (bottom) in the Turkish-Quiz-Instruct dataset. These distributions guide context window choices and computational resource planning.

as the primary supervision signal for the student models. Any flaws in the seed data risk being learned and amplified by the student models. Therefore, we conducted a rigorous human evaluation to assess the quality of the seed items generated by the teacher model (GPT-4-Turbo).

Human Evaluation Protocol A representative sample of the generated quizzes was scrutinized by native Turkish speakers possessing expertise in the linguistic subtleties of the language and familiarity with educational assessment standards. The evaluation utilized a comprehensive five-point rating scale (A–E), meticulously designed to assess the accuracy, relevance, and pedagogical worth of the items:

- **RATING-A (Excellent):** Questions are factually accurate, grammatically correct, unambiguous, and directly relate to significant concepts from the source text. Distractors (for MCQs) are plausible yet clearly incorrect.
- **RATING-B (Good):** Questions are mostly factual and related to the text, but may exhibit minor grammatical issues, slight ambiguity, or lapses in addressing the most significant concepts.
- **RATING-C (Acceptable):** Questions are loosely related to the text but may address peripheral topics or be overly simplistic.
- **RATING-D (Poor):** Questions contain factual inaccuracies, significant grammatical errors, or are only minimally relevant to the text.
- **RATING-E (Unusable):** Questions are demonstrably wrong, misleading, contain the answer (leakage), or have no clear connection to the educational text.

Seed Data Quality Results The results of the human evaluation are presented in Figure 8.5. The findings affirm the exceptional quality of the seed dataset. A substantial

majority, 93.3%, of the generated items were rated as RATING-A. The proportion of low-quality items (D and E) was minimal. This high proportion of top-rated items confirms the effectiveness of the sophisticated prompt engineering strategy and the capability of the teacher model (GPT-4-Turbo) [27] to produce high-fidelity, educationally valuable quizzes, even in a morphologically complex language like Turkish.
Figure 8.6 provides a concrete, illustrative example of a content-question pair from the dataset, focusing on the topic of photosynthesis. It showcases the source passage (a), the corresponding generated MCQ (b), and the derived SAQ (c), demonstrating the clarity and relevance of the generated items.

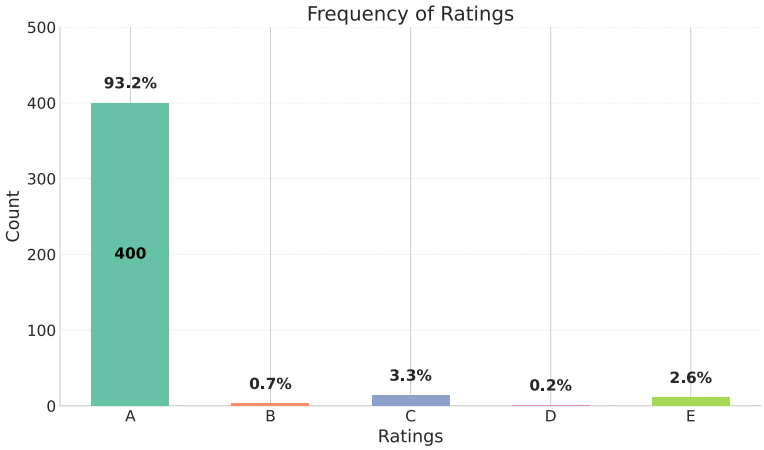


Figure 8.5: Human ratings (A–E) for a held-out sample of teacher-generated items (GPT-4-Turbo); 93.3% were rated A, indicating high seed data quality.

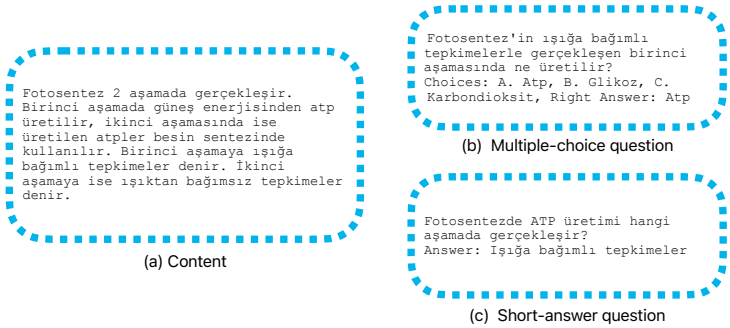


Figure 8.6: Illustrative content–question pair for photosynthesis from the Turkish-Quiz-Instruct dataset: (a) source passage; (b) generated MCQ; (c) generated SAQ.

8.4 Experimental Validation

This section details the comprehensive experimental validation of the proposed methodology. We rigorously evaluate the performance of the fine-tuned student models against their base versions for both MCQ and SAQ generation tasks. The evaluation employs a dual approach, utilizing both automatic metrics for measuring overlap and human evaluation for assessing pedagogical quality.

8.4.1 Experimental Setup

As mentioned, We evaluate three distinct models: Llama-2-7b-chat-hf, Llama-2-13b-chat-hf, and GPT-3.5-Turbo. Each model is assessed in two configurations: base (off-the-shelf, pre-fine-tuning) and fine-tuned (post-fine-tuning on *Turkish-Quiz-Instruct*). This allows for a direct measurement of the impact of the distillation process.

Automatic Evaluation Metrics. We employ the standard ROUGE [125] metrics (ROUGE-1, ROUGE-2, and ROUGE-L) to quantify the n-gram and longest common subsequence overlap between the generated items and the seed references (generated by the GPT-4-Turbo teacher model). It is crucial to acknowledge the limitations of ROUGE in assessing pedagogical quality. ROUGE does not account for syntactic correctness, semantic ambiguity, or the validity of paraphrasing. However, in this context, they serve as a useful proxy measure for lexical drift and content fidelity, indicating how closely the student models adhere to the high-quality references produced by the teacher.

Human Evaluation. To overcome the inherent limitations of automatic metrics, we conducted a comprehensive human evaluation using the same rigorous five-level rating scale (A–E) described in Section 8.3.1. Native Turkish speakers evaluated the quality, relevance, and correctness of the quizzes generated by all models in both configurations.

8.4.2 Multiple-Choice Question (MCQ) Generation Results

The evaluation of MCQ generation highlights the significant, positive impact of the instruction fine-tuning process on the performance of all models.

Automatic Metrics (MCQ)

Table 8.1 summarizes the ROUGE scores for MCQ generation. The results reveal substantial improvements across all models following fine-tuning. The base Llama-2 models (7B and 13B) exhibit very low ROUGE scores (ROUGE-L below 5). This indicates a significant divergence from the reference MCQs and suggests that these models are largely unable to reliably perform the task off-the-shelf in the Turkish context. After fine-tuning, their performance increases markedly. The Llama-2-13b model, in particular, shows a dramatic improvement, with ROUGE-L increasing nearly five-fold from 4.28 to 19.75. GPT-3.5-Turbo begins with a considerably stronger baseline (ROUGE-L 21.38),

Group	Model Name	# Params	ROUGE-1	ROUGE-2	ROUGE-L
Base LLMs	LLAMA2-CHAT	7B	4.71	1.42	3.88
	LLAMA2-CHAT	13B	5.33	1.73	4.28
	GPT-3.5 TURBO	–	27.53	13.18	21.38
Finetuned LLMs	LLAMA2-CHAT	7B	17.33	2.88	12.11
	LLAMA2-CHAT	13B	24.34	11.44	19.75
	GPT-3.5 TURBO	–	**36.64**	**20.36**	**28.82**

Table 8.1: MCQ generation results: ROUGE score comparison before and after fine-tuning on Turkish-Quiz-Instruct. Fine-tuning delivers substantial gains across all models.

reflecting its higher capacity. However, it still benefits significantly from the task-specific fine-tuning, achieving the highest overall scores (ROUGE-L 28.82).

Human Evaluation (MCQ)

The human evaluation results, visualized in Figure 8.7, provide a more nuanced understanding of the qualitative improvements, corroborating the automatic metrics. The dis-

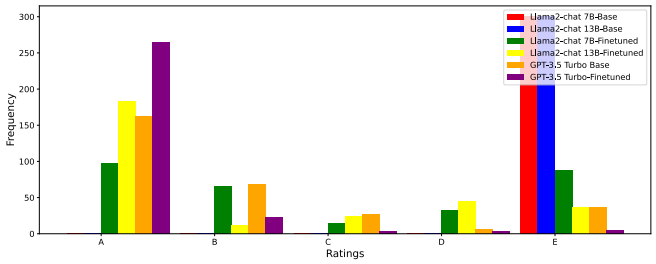


Figure 8.7: Human evaluation (A–E) for MCQs before and after fine-tuning. Base Llama-2 models predominantly generate E-rated (unusable) items. Post-tuning, there is a strong shift toward A-rated (high-quality) items for all models.

parity between the base and fine-tuned models is stark. Before fine-tuning, both Llama-2-7b and Llama-2-13b struggled profoundly with the task. A vast majority of their outputs were categorized as RATING-E (unusable). This indicates a fundamental failure to adhere to the task requirements, the required format, or the linguistic constraints of Turkish QG. After fine-tuning on *Turkish-Quiz-Instruct*, these models demonstrate a remarkable transformation. The frequency of E-rated items drops dramatically, replaced by a substantial increase in A-rated (high-quality) items. The Llama-2-13b model, benefiting from its larger capacity, shows a particularly strong capability for generating high-quality MCQs post-tuning, emerging as a viable open-source tool. GPT-3.5-Turbo also exhibits marked improvement. While the base model generated a heterogeneous mix of quality levels, the fine-tuned version shows a significant concentration of A-rated items, confirm-

ing its superior performance in generating reliable, well-formed, and pedagogically sound MCQs.

8.4.3 Short-Answer Question (SAQ) Generation Results

The evaluation of SAQ generation reveals even more dramatic gains, particularly for the open-source models, suggesting that this format may be more amenable to the distillation process.

Automatic Metrics (SAQ)

Table 8.2 summarizes the ROUGE scores for SAQ generation. The improvements observed here are significantly larger in magnitude than those for MCQs. The base Llama-2

Group	Model Name	# Params	ROUGE-1	ROUGE-2	ROUGE-L
Base LLMs	Llama2-chat	7B	3.31	1.04	2.86
	Llama2-chat	13B	3.81	1.33	3.22
	GPT-3.5 Turbo	–	28.19	15.87	22.22
Finetuned LLMs	Llama2-chat	7B	**54.37**	**47.33**	49.01
	Llama2-chat	13B	52.99	46.88	**49.54**
	GPT-3.5 Turbo	–	45.31	31.59	40.32

Table 8.2: SAQ generation results: ROUGE score comparison before and after fine-tuning. The gains for Llama-2 models are transformative, moving from near-zero performance to high ROUGE scores.

models were essentially incapable of generating coherent SAQs, with ROUGE-L scores near 3. Post-fine-tuning, their performance skyrockets, with both 7B and 13B models achieving ROUGE-L scores near 49. This represents a transformative improvement, highlighting the exceptional effectiveness of the distillation approach for this specific task format. Interestingly, the fine-tuned Llama-2 models significantly outperform the fine-tuned GPT-3.5-Turbo in terms of ROUGE overlap for SAQ generation, suggesting they adhere more closely to the reference style in this format.

Human Evaluation (SAQ)

The human evaluation results for SAQs, shown in Figure 8.8, strongly mirror the trends observed in the automatic metrics. Before fine-tuning, the failure of the base Llama-2 models was absolute: all generated questions were rated as RATING-E, confirming their inability to generate acceptable Turkish short-answer questions off-the-shelf. After fine-tuning, there is a dramatic reversal, with the vast majority of outputs now rated as RATING-A. Both Llama-2-13b and Llama-2-7b demonstrate excellent performance, validating their utility. GPT-3.5-Turbo also shows significant improvement, with a large increase in A-rated items. The overall results underscore the resounding success of the

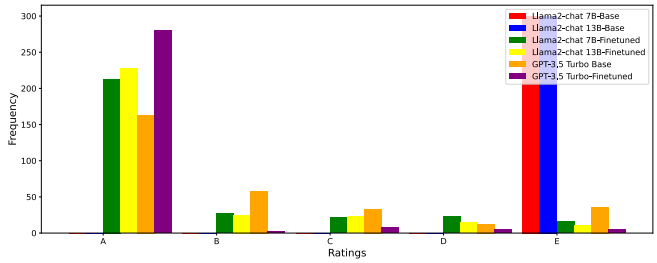


Figure 8.8: Human evaluation (A–E) for SAQs before and after fine-tuning. Base Llama-2 models fail completely (all E-rated). Fine-tuned Llama-2 and GPT-3.5-Turbo models show large quality gains, dominated by A-rated items.

methodology in creating specialized models capable of high-quality SAQ generation in a low-resource language context.

8.4.4 Analysis and Discussion

The experimental results provide compelling validation for the "From Language to Learning" framework applied to Turkish QG. Several key observations emerge from the analysis.

The stark contrast between the performance of the base and fine-tuned Llama-2 models—particularly their complete failure in the base configuration—underscores the necessity of task-specific fine-tuning for specialized applications in morphologically complex languages. General-purpose instruction tuning, while powerful, is insufficient to ensure adherence to the specific constraints, formats, and linguistic nuances required for educational QG in Turkish.

The high quality of the *Turkish-Quiz-Instruct* dataset, derived from the GPT-4-Turbo teacher model, enables highly effective knowledge distillation. The student models successfully learn to replicate the structure, style, and quality of the seed data, demonstrating the viability of using synthetic data for complex educational tasks.

The gains in performance were notably more dramatic for SAQs than for MCQs, especially for the Llama-2 models. This suggests that SAQ generation, which requires precision in formulating the question and identifying the correct answer, might be a more tractable task for fine-tuned models than MCQ generation. MCQ generation involves the complex and nuanced sub-task of generating plausible distractors, which remains a significant challenge even for advanced models.

While the fine-tuned GPT-3.5-Turbo generally achieves high quality, the remarkable performance of the fine-tuned Llama-2-13b model is particularly significant. It emerges as a strong, open-source alternative, demonstrating the potential to democratize access to high-quality educational tools without reliance on costly proprietary APIs.

8.5 Conclusion

This chapter presented a complete, end-to-end framework for automating the generation of educational quizzes in Turkish under the "From Language to Learning" paradigm. By introducing the *Turkish-Quiz-Instruct* dataset and applying a rigorous two-stage distillation workflow, we specialized open-source LLMs for a morphologically complex setting. In summary, we developed a robust pipeline that transforms Turkish instructional passages into structured MCQs and SAQs while enforcing pedagogical quality through structured prompting and validation (Figures 8.1–**??**). We released *Turkish-Quiz-Instruct*, a high-quality Turkish seed dataset that enables effective specialization for item writing across diverse academic subjects (Figure **??**(a)). We showed that instruction fine-tuning yields substantial, consistent gains over prompt-only baselines, turning open models such as Llama-2 7B/13B from unusable to highly effective on both MCQs and SAQs (Tables 8.1–8.2). Rigorous human evaluation confirmed marked improvements in item quality after tuning, establishing the pedagogical viability and reliability of the generated content (Figures 8.7–8.8). We further demonstrated that prompt templates encoding assessment best practices reduce invalid items and stabilize outputs, a particularly important property for Turkish. Finally, we offered a modular, generalizable blueprint that scales to other low-resource, morphologically rich languages, advancing automated assessment and contributing to educational equity.

Chapter 9

Persian Multiple-Choice Question Generation: Data, Models, and Evidence

9.1 Motivation and Scope

Multiple-Choice Questions (MCQs) are central to retrieval practice and scalable assessment, yet authoring high-quality items—especially *plausible* distractors—is slow and expertise-intensive. In Iranian Persian, this bottleneck is amplified by low-resource conditions: scarce paired corpora of texts and expert items, limited language-specific tools and benchmarks, and linguistic particulars (flexible SOV word order, Perso–Arabic script and punctuation, and rich, idiomatic vocabulary). General-purpose, Anglophone-trained models often struggle to produce grammatical, culturally appropriate questions and distractors under these constraints.

This work proposes a principled, end-to-end methodology for automatically generating high-quality MCQs from standard Iranian Persian texts. Given an educational passage, the system produces a clear stem, one unambiguous key, and three plausible distractors, with outputs constrained by linguistic correctness, content fidelity, assessment best practices, and overall appropriateness. Our contribution is a scalable, replicable blueprint for automated assessment in a low-resource setting, built on three pillars:

- *PersianMCQ-Instruct*: a novel dataset of >12,000 MCQs from >4,000 Persian Wikipedia passages for training and evaluation.
- A three-stage prompt cascade (Text Augmentation → MCQ Generation → MCQ Refinement) using a teacher model (GPT-4o).
- Fine-tuning compact, open-source student models (7–9B)—**PMCQ-Llama3.1-8b**, **PMCQ-Gemma2-9b**, and **PMCQ-Mistral-7b**—to enable reliable Persian MCQ generation on commodity hardware.

Through automatic and human evaluations, we show that this approach reduces authoring cost while meeting the above constraints, establishing a practical path for Persian—and a template for other low-resource languages.

9.2 A Three-Stage Framework for Persian MCQ Generation

To address the challenge of automating Persian educational quiz generation, we devised a structured, multi-step pipeline. This methodology leverages a systematic process to

transform curated Persian content into a high-quality instructional dataset, which is then used to specialize efficient language models for the target task. This section provides a detailed account of this framework, covering content curation, the three-stage process of synthesizing the instructional dataset, and the subsequent fine-tuning of the student models.

9.2.1 Architectural Overview

The methodology follows a modular pipeline, as illustrated in Figure 8.1. It begins with the systematic collection and preparation of raw educational content from Persian Wikipedia. This curated content serves as the input for a three-stage generation cascade powered by a large "teacher" model. The resulting high-quality dataset is then used to transfer knowledge to smaller, more efficient "student" models through instruction fine-tuning.

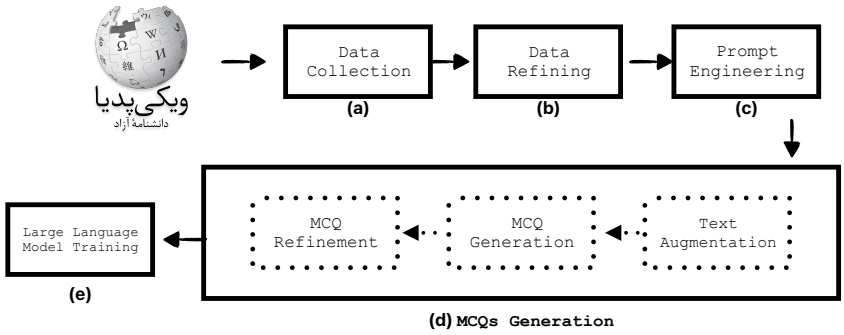


Figure 9.1: The end-to-end methodology for Persian MCQ generation. The process includes: (a) Data collection from popular Persian Wikipedia pages; (b) Data refinement and filtering to ensure quality; (c) The design of specialized prompts for each stage; (d) The three-stage generation cascade (text augmentation, MCQ generation, MCQ refinement) using GPT-4o to produce the seed dataset; and (e) The fine-tuning of smaller language models using the generated dataset.

9.2.2 Dataset Generation (PersianMCQ-Instruct)

The first and most foundational stage is the creation of the *PersianMCQ-Instruct* dataset. This process relies on harnessing the advanced capabilities of a powerful teacher LLM (GPT-4o) through meticulous data preparation and a sophisticated, multi-stage prompting strategy inspired by the Agent Instruct framework.

Content Ingestion and Preparation The basis of the dataset is a broad collection of Persian texts sourced from high-quality Persian Wikipedia pages, chosen to ensure diverse topical coverage across academic disciplines like mathematics, history, biology, and literature.

Data Scraping and Curation. The information extraction process began with a focused crawl of curated Wikipedia lists, such as "Featured articles," "Good articles," and "Vital articles". This strategy ensures that the source material is well-regarded, factually vetted, and generally aligned with educational content standards. The initial crawl yielded a corpus of 8,894 unique pages.

Data Cleaning and Filtering. To ensure the integrity and utility of the source material, a rigorous data cleaning protocol was implemented (Figure 9.1). First, pages containing fewer than 100 words were removed, as they typically lack sufficient conceptual depth to generate meaningful MCQs. To manage computational resources and standardize context length during the generation phase, passages were capped at a maximum of 500 words. Additionally, a manual review process was conducted to identify and filter out pages with sensitive content, aligning the dataset with ethical considerations. This multi-step filtering process resulted in a final, high-quality corpus of 4,159 curated passages ready for question generation.

The Three-Stage Prompt Cascade for Quality Control

A key innovation of our methodology is the use of a three-stage prompt cascade to guide the teacher model toward producing pedagogically sound MCQs (Figure 9.1). This approach breaks down the complex task of question generation into three manageable and distinct steps, with each step building upon the output of the previous one. The prompts for each stage are shown in Figure 9.2.

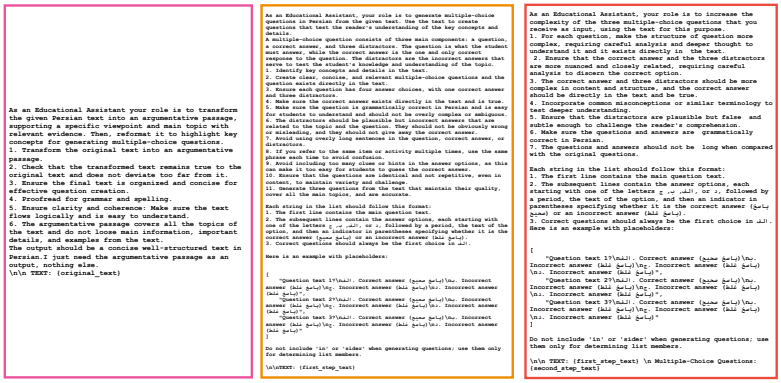


Figure 9.2: The three-stage prompt cascade used for generating the PersianMCQ-Instruct dataset. (a) The Text Augmentation prompt reframes the source text. (b) The MCQ Generation prompt creates an initial draft of the questions. (c) The MCQ Refinement prompt increases complexity and ensures pedagogical quality.

Step 1: Text Augmentation. The first prompt instructs the model to transform the raw Wikipedia text into a well-structured, argumentative passage. This step serves two critical purposes: it forces the model to synthesize and rephrase the key information rather than relying on simple extraction, and it produces a more coherent and focused

text from which to generate questions. This process enhances data quality and is particularly valuable for preparing texts in low-resource languages for downstream tasks.
Step 2: MCQ Generation. Using the augmented text from the first step, the second prompt guides the model to generate an initial set of three MCQs. This prompt specifies the required structure: a question stem, four answer choices (one correct and three distractors), and clear labeling of the correct answer. It provides explicit rules to ensure the questions are clear, relevant, and directly answerable from the text.
Step 3: MCQ Refinement. The final prompt takes the augmented text and the initially generated MCQs as input and instructs the model to refine them. This crucial step focuses on increasing the pedagogical value of the questions by making them more complex and challenging. The prompt guides the model to create more nuanced distractors, incorporate common misconceptions, and ensure the final questions require deeper understanding rather than simple fact retrieval. **Seed Dataset Synthesis** Using this three-stage cascade, the teacher model (GPT-4o) was employed to autonomously generate three distinct MCQs for each of the 4,159 passages, resulting in a comprehensive dataset of over 12,000 question-answer pairs. GPT-4o was selected for its superior performance in natural language understanding and its ability to adhere to complex, multi-step instructions, which was essential for executing the prompt cascade effectively. The resulting *PersianMCQ-Instruct* dataset forms the core of our contribution, providing a rich source of structured supervision data.

9.2.3 Instruction Fine-Tuning and Knowledge Distillation

In the second stage, the synthetically generated *PersianMCQ-Instruct* dataset serves as the training corpus to fine-tune smaller, more accessible "student" models. This knowledge distillation process effectively transfers the specialized persian MCQ generation capabilities of the powerful but resource-intensive teacher model to the student models, significantly enhancing their ability to produce high-quality Persian quizzes in an efficient manner.
Model Selection We selected three state-of-the-art, instruction-tuned LLMs known for their strong performance and multilingual capabilities, **gemma2-9b-it** [**?**], **Llama3.1-8b-Instruct** [126] and **Mistral-7b-Instruct-v0.3** .[127]
These models were chosen because they represent a range of modern architectures, are of a manageable size (7B to 9B parameters), and have demonstrated comprehensive support for non-English languages, including Persian.
Fine-Tuning Methodology The fine-tuning process was designed to meticulously adapt these base models for the specific task of generating MCQs from Persian educational texts. The dataset was partitioned into a training set of 12,000 MCQs and an evaluation set of 476 MCQs.
Parameter-Efficient Fine-Tuning (PEFT). Given the significant computational resources required for full fine-tuning, we employed Low-Rank Adaptation (LoRA) [121] , a prominent PEFT [120] technique. LoRA freezes the pre-trained model weights and injects smaller, trainable low-rank matrices into the Transformer architecture. This approach drastically reduces the number of trainable parameters, making the fine-tuning

process computationally efficient while achieving performance comparable to full fine-tuning.

Training Configuration. The models were trained on two NVIDIA A6000 GPUs, each with 48 GB of RAM. The training was conducted for 3 epochs with a maximum sequence length of 3500 tokens. Key hyperparameters included a learning rate of 1×10^{-4} with a cosine scheduler, a weight decay of 1×10^{-4}, and a batch size of 4 with 4 steps of gradient accumulation. To optimize memory usage, we enabled gradient checkpointing and FlashAttention. For LoRA, we used a rank (r) of 16 and an alpha (α) of 32. The entire process was orchestrated using DeepSpeed to enhance computational efficiency and scalability.

This meticulous fine-tuning strategy ensured that the student models were effectively specialized for the nuances of Persian MCQ generation, transforming them from general-purpose assistants into reliable educational tools.

9.3 The PersianMCQ-Instruct Dataset

The cornerstone of this research is the *PersianMCQ-Instruct* dataset, a novel and large-scale collection of Persian educational texts paired with structured, high-quality MCQs. This dataset is a primary contribution of our work, addressing the critical scarcity of resources for Persian educational NLP and providing a robust foundation for the development and evaluation of QG models.

9.3.1 Dataset Characteristics and Statistics

The dataset consists of 4,159 unique passages sourced from Persian Wikipedia, each accompanied by three generated MCQs, for a total of over 12,000 questions. The content spans a wide range of academic disciplines, ensuring broad topical coverage.

Word Distribution and Context Length: Understanding the length distributions of the source passages and the generated quizzes is crucial for optimizing model training, especially for setting appropriate context window sizes and planning for computational resources. Figure 9.3 visualizes the word distributions for both the educational contexts and the generated MCQs.

The source passages (top panel) show a distribution concentrated between 100 and 500 words, consistent with our filtering criteria. This range provides sufficient context for generating meaningful questions without being computationally prohibitive. The generated MCQs (bottom panel) are significantly more concise, with most falling between 40 and 80 words. This indicates that the generation process successfully distills key information into focused questions, a desirable characteristic for reducing cognitive load on learners.

Quality Validation of the Seed Data: The success of the knowledge distillation process is fundamentally dependent on the quality of the synthetic dataset. Flaws or biases in the seed data risk being learned and amplified by the student models. Therefore, we conducted a rigorous human evaluation to assess the quality of the seed items generated

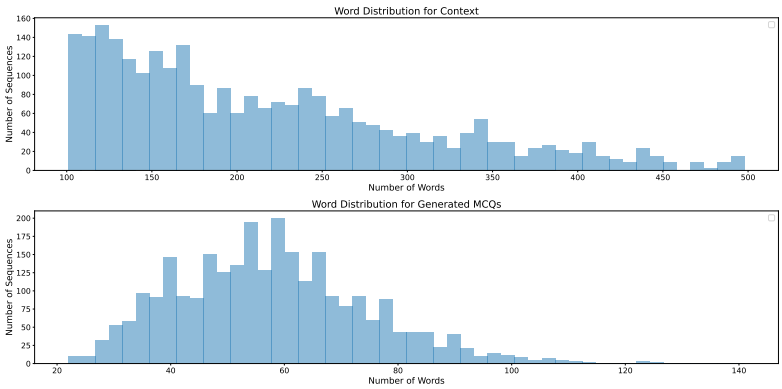


Figure 9.3: Word distributions for the source contexts (top) and the generated MCQs (bottom) in the PersianMCQ-Instruct dataset. The contexts range from 100 to 500 words, while the generated questions are more concise, reflecting effective summarization and question formulation.

by the teacher model (GPT-4o).

Human Evaluation Protocol: A random sample of 600 generated MCQs was evaluated by a panel of two native Persian speakers, both of whom are postgraduate university students with the expertise to assess linguistic quality and educational relevance. To measure inter-annotator agreement, 200 of these questions were evaluated by both experts. The evaluation utilized a comprehensive five-point rating scale (A–E), designed to capture multiple dimensions of question quality:

- **RATING-A (Excellent):** The question and answer are factually accurate, directly relevant to significant concepts in the text, grammatically flawless, and unambiguous.
- **RATING-B (Good):** The question is mostly factual and relevant but may have minor grammatical issues, be slightly incomplete, or fail to explicitly identify the correct answer.
- **RATING-C (Acceptable):** The question is loosely related to the text, addresses tangential topics, or has serious grammatical issues. The answer may be incorrect.
- **RATING-D (Poor):** The question or answer contains factual inaccuracies or is only minimally relevant to the source text.
- **RATING-E (Unusable):** The question or answer was not generated, is demonstrably wrong or misleading, or has no clear connection to the text.

Seed Data Quality Results: The results of the human evaluation, presented in Figure 9.4, affirm the exceptionally high quality of the seed dataset. An impressive **86.8%** of the generated items received a rating of A (Excellent). The proportion of low-quality items was minimal, with only 3.2% rated D and 0.3% rated E. This result confirms the

effectiveness of our three-stage prompt cascade and the capability of the teacher model to produce high-fidelity, educationally valuable content in Persian. The annotators achieved a high agreement rate of 0.96, with a Cohen's kappa of 0.86, indicating a reliable and consistent evaluation process.

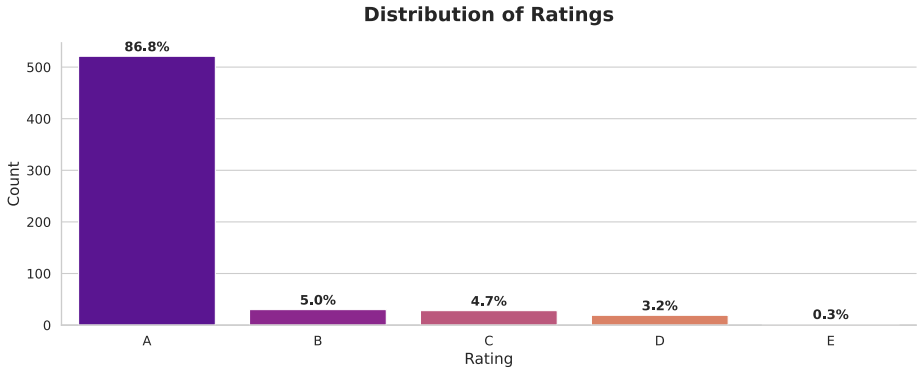


Figure 9.4: Distribution of human ratings (A–E) for the MCQs in the PersianMCQ-Instruct dataset. The overwhelming majority (86.8%) of the generated questions were rated as 'A' (Excellent), validating the high quality of the seed data.

9.4 Experimental Validation

This section details the comprehensive experimental validation of our methodology. We rigorously evaluate the performance of the fine-tuned student models against their base versions, employing a dual approach that combines automatic metrics for measuring semantic similarity and in-depth human evaluation for assessing pedagogical quality.

9.4.1 Experimental Setup

Models: We evaluate three models—gemma2-9b-it, Llama3.1-8b-Instruct, [126] and Mistral-7b-Instruct-v0.3—in two configurations: **base** (off-the-shelf, before fine-tuning) and **fine-tuned** (after fine-tuning on *PersianMCQ-Instruct*). This direct comparison allows for a precise measurement of the performance gains attributable to our distillation process.

Data Splits: The experiments utilize the dedicated evaluation split of our dataset, which consists of 476 MCQs from 159 held-out passages that were not seen during the training phase.

Automatic Evaluation: We use ROUGE and BERTScore [128] to compare the generated questions against the reference questions in the evaluation set. ROUGE measures n-gram overlap, which is useful for gauging lexical similarity. However, because high-quality questions often involve paraphrasing, ROUGE scores can be misleadingly low. Therefore, we also employ BERTScore, which measures semantic similarity and provides

a more robust assessment of whether the generated question captures the same meaning as the reference.

Human Evaluation: To overcome the limitations of automatic metrics, we conducted a second comprehensive human evaluation. Using the same five-level rating scale (A–E), our expert annotators evaluated the outputs of all models, both before and after fine-tuning. This allowed us to directly compare the qualitative and pedagogical value of the generated MCQs.

9.4.2 MCQ Generation Results

The evaluation of MCQ generation reveals the profound and positive impact of the instruction fine-tuning process across all three models.

Automatic Metrics: We first assessed the quality of the seed dataset itself. As shown in Table 9.1, the ROUGE scores between the generated questions and the source text are very low (e.g., ROUGE-L F1 of 0.0226), while the BERTScore is very high (F1 of 0.8929). This pattern is desirable, as it indicates the model is successfully paraphrasing and abstracting information from the source text rather than simply copying it.

Metric	Precision	Recall	F1
ROUGE-1	0.0167	0.0690	0.0238
ROUGE-2	0.0060	0.0246	0.0084
ROUGE-L	0.0158	0.0667	0.0226
BERTScore	0.8886	0.8976	0.8929

Table 9.1: Average ROUGE and BERTScore between generated MCQs and their source context in the PersianMCQ-Instruct dataset. The combination of low ROUGE and high BERTScore confirms that the questions are paraphrastic yet semantically faithful to the source material.

Next, we compared the BERTScores of the student models' outputs against the reference questions before and after fine-tuning. As summarized in Table 9.2, fine-tuning led to consistent improvements across all models. The fine-tuned *PMCQ-Llama3.1-8b* achieved the highest BERTScore F1 of 0.9049, indicating a strong semantic alignment with the high-quality reference questions from our dataset.

Human Evaluation

The human evaluation results, visualized in Figure 9.5, provide a much clearer picture of the qualitative improvements. The contrast between the base and fine-tuned models is dramatic.

- **Base Model Performance:** Before fine-tuning, the models struggled significantly. The base *Mistral-7b-Instruct-v0.3* was almost completely unable to perform the task, with a staggering **83.5%** of its outputs rated as E (Unusable). The base

	Model Name		Precision	Recall	F1
Base	*gemma2-9b-it*	9	0.9004	0.8754	0.8877
	Llama3.1-8b-Instruct	8	0.8992	0.8630	0.8807
	Mistral-7b-Instruct-v0.3	7	0.8852	0.8631	0.8740
Fine-tuned	*PMCQ-Gemma2-9b*	9	0.9108	0.8956	0.9031
	PMCQ-Llama3.1-8b	8	**0.9135**	**0.8964**	**0.9049**
	PMCQ-Mistral-7b	7	0.9117	0.8959	0.9037

Table 9.2: BERTScores between questions generated by student models and reference questions from the evaluation set. Fine-tuning consistently improves semantic similarity for all models, demonstrating effective knowledge transfer.

Llama3.1-8b-Instruct produced a majority of B-rated items (64.5%), indicating that while its outputs were mostly factual, they were plagued by grammatical errors and an inability to correctly label answers. None of the base Llama or Mistral models produced a single A-rated question.

- **Post-Fine-Tuning Transformation:** After fine-tuning on *PersianMCQ-Instruct*, the models underwent a remarkable transformation. The frequency of unusable E-rated items plummeted, replaced by a surge in high-quality A-rated items. *PMCQ-Llama3.1-8b* became the top performer, with **81.5%** of its outputs rated A. *PMCQ-Mistral-7b* also showed a massive improvement, with 70.5% of its outputs achieving an A rating. This demonstrates that the fine-tuning process successfully taught the models to generate accurate, relevant, and well-formed Persian MCQs.

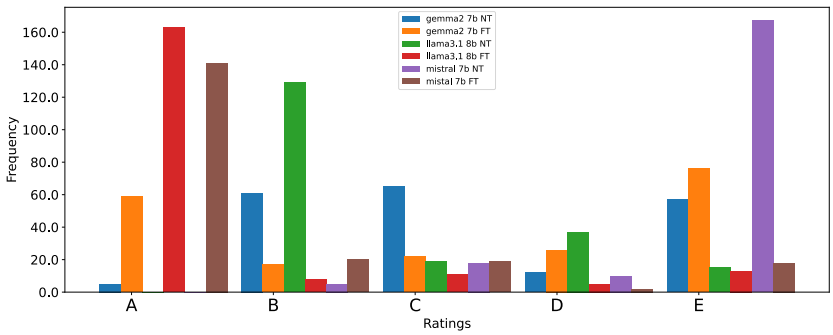


Figure 9.5: Human evaluation of base vs. fine-tuned models. The charts clearly show a dramatic shift away from low-quality ratings (especially E for Mistral) toward high-quality A ratings after fine-tuning, validating the effectiveness of the *PersianMCQ-Instruct* dataset.

The overall ratings, calculated by mapping the A-E scale to a 5-1 score, are summarized in Table 9.3. The fine-tuned *PMCQ-Llama3.1-8b* achieved an impressive overall

score of 4.51, approaching the near-perfect score of 4.75 of the seed dataset itself and far surpassing its base score of 3.31.

Model	Overall Rating
Base *gemma2-9b-it*	2.72
Base *Llama3.1-8b-Instruct*	3.31
Base *Mistral-7b-Instruct-v0.3*	1.30
Fine-tuned *PMCQ-Gemma2-9b*	2.78
Fine-tuned *PMCQ-Llama3.1-8b*	**4.51**
Fine-tuned *PMCQ-Mistral-7b*	4.32
Reference *PersianMCQ-Instruct*	4.75

Table 9.3: Overall human ratings based on a 5-point scale (A=5, E=1). The fine-tuned models, particularly Llama and Mistral, show massive improvements, confirming the high quality and effectiveness of the training dataset.

9.5 Analysis and Discussion

The experimental results provide compelling validation for our methodology and yield several key insights into automated question generation for low-resource languages like Persian.

The Absolute Necessity of Task-Specific Fine-Tuning. The starkest finding is the profound inadequacy of general-purpose, off-the-shelf models for this specialized task. The near-total failure of the base Mistral model (83.5% unusable outputs) and the significant flaws of the base Llama model underscore that even powerful instruction-tuned LLMs cannot reliably adhere to the specific formatting, linguistic, and pedagogical constraints of Persian MCQ generation without targeted training. Fine-tuning is not merely an enhancement but an absolute necessity.

Efficacy of the Synthetic Data and Distillation Approach. The dramatic performance improvements post-fine-tuning serve as strong evidence for the quality of the *PersianMCQ-Instruct* dataset. The ability of this synthetically generated data to successfully distill the complex task of question generation into smaller, efficient models validates our entire approach. It demonstrates that high-quality, large-scale datasets for specialized tasks can be bootstrapped using powerful teacher models, overcoming the data scarcity bottleneck.

Model-Specific Behaviors and Error Modes. Before fine-tuning, each model exhibited unique failure patterns. *Mistral* often failed to generate anything coherent. *Llama* struggled with grammatical correctness and labeling answers. *Gemma* had a tendency to insert English words into its answers when non-Persian words were present in the source

text and to bold keywords without being prompted. Fine-tuning resolved most of these issues, especially for Llama and Mistral, which learned to correctly follow the format and generate fluent Persian. However, the fine-tuned Gemma model still struggled with non-Persian words and produced a high rate of E-rated outputs (38.0%), suggesting it was less adaptable to this specific task.

Quality vs. Complexity Trade-off. An interesting observation is that while the fine-tuned models became highly accurate and reliable, their generated questions were often less complex and more extractive than the comprehension-based questions in the original *PersianMCQ-Instruct* dataset. The teacher model, guided by the refinement prompt, produced questions requiring deeper analysis. The student models, in contrast, learned the format and style but tended to generate questions that were more closely aligned with sentences from the source text. This highlights a trade-off between the reliability of smaller models and the nuanced, higher-order reasoning capabilities of larger teacher models.

9.6 Conclusion

This chapter presented a comprehensive, end-to-end framework for automating the generation of high-quality educational MCQs in Persian. By developing the *PersianMCQ-Instruct* dataset through a three-stage prompt cascade and leveraging it to fine-tune compact language models, we deliver a scalable solution to a key challenge in a low-resource language context.

The main contributions are as follows. First, we designed and validated a pipeline that transforms raw Persian text into pedagogically sound MCQs via a structured, multi-stage generation process with explicit quality controls. Second, we built *PersianMCQ-Instruct*, a large-scale resource with 12,000+ items and an 86.8% A-rating, which we expect to catalyze progress in Persian educational NLP. Third, we showed that task-specific instruction fine-tuning is essential for this task: open-source models such as Llama-2 and Mistral improved from nearly unusable to highly effective, with the fine-tuned *PMCQ-Llama3.1-8b* reaching an 81.5% A-rating in human evaluations. Finally, the methodology offers a transferable blueprint for automated assessment generation in other low-resource and linguistically diverse settings, advancing educational equity.

Several limitations remain. The source is Persian Wikipedia, whose encyclopedic style may not transfer to textbooks, literary analysis, or conversation. The system prioritizes fact-based questions and struggles with multi-hop or cross-passage reasoning. Human evaluation covered a small sample; scaling is costly and slow, and automatic metrics (e.g., BERTScore) miss pedagogical qualities like distractor plausibility and clarity. And while PEFT helps, training/serving 7–9B models still requires specialized GPUs, limiting under-resourced adopters and underscoring the need for further compression/optimization.

To address these constraints, we outline several directions for future work. We plan to diversify data sources by incorporating domain-focused educational materials (e.g., mathematics, physics, history) from textbooks and academic articles to improve domain

adaptability. We aim to control question difficulty and cognitive level, aligning with Bloom's taxonomy (recall, analysis, synthesis) and exploring psychometric calibration via Item Response Theory. We will pursue stronger distractor generation methods that explicitly target common misconceptions, potentially via dedicated models optimized for this sub-task. Lastly, we intend to extend the blueprint to other low-resource languages, broadening the impact and accessibility of educational technology.

Part V

Syntactic Feedback for Student Writing

Chapter 10

Automating Syntax Feedback for Student Writing

10.1 Motivation and Scope

Mastering English syntax—an intricate, exception-laden rule system—remains challenging for both second-language learners and native speakers. Traditional instruction provides foundations but rarely scales to deliver timely, personalized, line-level feedback, creating a persistent bottleneck in writing instruction. Advances in AI and NLP, particularly Large Language Models (LLMs), offer a path to scalable, adaptive support capable of precise error detection and correction without replacing educators; instead, they free instructors to focus on higher-order concerns such as argumentation and style.

This chapter presents an end-to-end methodology for automatically generating high-quality, structured syntax feedback on authentic student essays. A central real-world constraint is privacy: widely used educational corpora (e.g., ASAP) replace personally identifiable information with anonymized placeholders (e.g., `@PERSON1`), which LLMs often misread as syntactic noise, degrading feedback quality. Our pipeline therefore begins with an LLM-based *placeholder replacement* step that maps anonymized tokens to contextually plausible surrogates, preserving privacy while restoring linguistic naturalness. Building on this, we introduce and release *Essay-Syntax-Instruct*, a large-scale resource of 8,320 essay–feedback pairs with expert-validated, category-structured annotations. Finally, we demonstrate a knowledge-distillation framework: a strong teacher model (GPT-3.5-Turbo) seeds the dataset and supervises fine-tuning of smaller, open-source student models (Llama-2 and Mistral variants), improving deployability and cost-efficiency.

Across automatic and human evaluations, the system produces linguistically correct, contextually relevant, and pedagogically actionable feedback. In sum, this work defines the problem, motivates automation grounded in real classroom constraints, and delivers a scalable, privacy-aware pipeline and dataset that advance automated syntax support.

10.2 A Framework for Automated Syntax Feedback Generation

To address the multifaceted challenge of automating syntax feedback, we designed a systematic, multi-stage pipeline. This framework is engineered to transform raw, anonymized student essays into high-quality, structured feedback by leveraging a powerful teacher LLM to create a supervision dataset, which is then used to specialize more efficient stu-

dent models. This section provides a detailed overview of the architectural design and the core stages of the methodology: synthetic data generation and knowledge distillation through instruction fine-tuning.

10.2.1 Architectural Overview

The end-to-end methodology follows a structured pipeline, as illustrated in Figure **??**. The process commences with the ingestion of authentic student essays from the ASAP corpus [1]. It then proceeds through a series of carefully orchestrated steps: a critical data refinement stage that includes our novel placeholder replacement technique, a sophisticated prompt engineering phase to define the structure of the feedback, the synthesis of the seed dataset using the teacher model, and finally, the fine-tuning of the student LLMs to imbue them with the specialized feedback-generation capability.

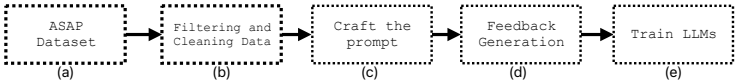


Figure 10.1: The end-to-end methodological pipeline: (a) ingestion of human-authored essays from the ASAP dataset [1, 2]; (b) data cleaning and our novel placeholder replacement process; (c) crafting of a specialized prompt to elicit category-structured syntax feedback; (d) synthesis of the seed dataset (*Essay-Syntax-Instruct*) using a powerful teacher LLM; (e) instruction fine-tuning of accessible student LLMs for reliable and scalable feedback generation.

10.2.2 Data Generation (*Essay-Syntax-Instruct*)

The cornerstone of our framework is the creation of a high-quality, large-scale instructional dataset, which we have named *Essay-Syntax-Instruct*. This critical first stage relies on the advanced capabilities of a powerful "teacher" LLM (GPT-3.5-Turbo) guided by meticulous data preparation and sophisticated prompt engineering.

Content Ingestion and Preparation: The foundation of our dataset is the well-established *ASAP (Automated Student Assessment Prize) dataset*, a resource frequently employed in AES research. This corpus comprises a diverse collection of essays authored by students in Grades 7 through 10, responding to eight distinct prompts. The essays vary in length and style, ranging from narratives to source-dependent responses, providing a rich and authentic representation of student writing.

A significant feature of the ASAP dataset is its comprehensive anonymization. To protect student privacy, the original essays were processed using tools like the Stanford Named Entity Recognizer (NER) [129] replace identifiable information—such as names, locations, dates, and organizations—with generic placeholders (e.g., `@PERSON1`, `@ORGANIZATION2`). As previously noted, these placeholders disrupt the natural flow of the text and are often

[1] https://www.kaggle.com/datasets/lburleigh/asap-2-0

misinterpreted as errors by LLMs.
To overcome this, we implemented a novel *placeholder replacement* sub-step. We engineered a specific prompt, shown in Figure 10.2, instructing GPT-3.5-Turbo [122] to identify all placeholders and substitute them with contextually coherent and plausible alternatives that seamlessly integrate into the essay's narrative while remaining non-identifying. This crucial process restored the linguistic integrity of the essays, making them suitable for accurate syntactic analysis.

```
Your task entails identifying and substituting placeholders within the
text.

These placeholders are denoted by an '@' symbol followed by uppercase
letters indicating various categories such as months, percentages,
locations, numbers, person names, and more. It's crucial to ensure that the
replacements seamlessly integrate into the text.

Ensure that all placeholders are detected and replaced with the appropriate
replacements.

To streamline the process, kindly furnish your replacements in JSON format.
The JSON should comprise pairs where each placeholder is associated with
its corresponding replacement word.


Text: {essay}
```

Figure 10.2: The structured prompt template used for the placeholder replacement task. This prompt instructs the LLM to identify all anonymized placeholders and furnish replacements in a JSON format to ensure seamless and accurate substitution.

Prompt Engineering for Pedagogical Syntax Feedback: With the essays restored to a natural linguistic state, the next step was to generate the syntax feedback itself. The quality and structure of this feedback are paramount, as it forms the supervision signal for the student models. We designed a highly structured prompt, shown in Figure 10.3, to guide the teacher LLM (GPT-3.5-Turbo) in performing a detailed syntactic analysis.
The prompt was meticulously crafted to enforce a consistent and pedagogically useful output format based on seven specific categories of common syntactic errors:

1. **Misspelled Words:** Identifying and correcting spelling mistakes.
2. **Conjunctions and Linking Phrases:** Analyzing the correct use of connecting words.
3. **Modifiers:** Checking for misplaced or dangling modifiers.
4. **Prepositions:** Evaluating the appropriate use of prepositions.
5. **Modal Verbs:** Assessing the correct application of modal verbs (e.g., can, should, must).
6. **Punctuation:** Correcting errors in punctuation usage.
7. **Articles:** Reviewing the use of definite and indefinite articles (a, an, the).

These categories were deliberately chosen as they encompass the majority of syntax-related errors typically found in student writing, ensuring the generated feedback would be comprehensive and relevant. The prompt explicitly instructs the model to list errors and their corrections for each category or to state "N/A" if no errors are found, ensuring a complete and parsable output.

```
Your task is to perform a syntax analysis on an essay, aiming to identify and amend errors across specific syntax categories. Your mission is to
meticulously inspect each category for mistakes, correct them, and elucidate your findings. Concentrate on the following syntax areas:

Syntax Categories to Check:

1. Misspelled Words: Detect and list each misspelling uniquely.
2. Conjunctions and Linking Phrases: Search and list misuse of adverbs or (and, or, but, then, and because) between sentences and paragraphs.
3. Modifiers: Search for improper uses of modifiers.
4. Prepositions: Check for and correct inaccuracies in the use of prepositions.
5. Modal Verbs: Assess modal verbs for their accurate application.
6. Punctuation: Identify and rectify punctuation errors.
7. Articles: Examine the application of "the," "a," and "an," correcting any misuse.

To successfully accomplish this analysis, follow these steps:

1. Begin by systematically searching for syntax errors in the highlighted categories.

2. Ensure that you have thoroughly identified all errors within the specified categories and accurately documented them in your final submission.

3. If no errors are found within a specific category, indicate its absence of errors by writing "N/A" for that category.

4. Document each error found in a structured manner as follows:
   - Syntax Category [Name of Syntax Category]:
     - Error: [found error], Correct version: [appropriate correction]
   Consider that Each error should be mentioned on a separate line for clarity and order.

5. Follow these steps closely to check every part of the syntax carefully. Remember to review and double-check thoroughly for mistakes so that you
don't miss any errors.

ESSAY: {essay}
```

Figure 10.3: The structured prompt template used with the teacher LLM for generating categorized syntax feedback. This prompt defines the seven analytical categories and specifies the required output format.

Seed Dataset Synthesis: Using the refined essays and the engineered prompt, we employed GPT-3.5-Turbo with a low temperature setting of 0.3 to ensure deterministic and high-fidelity outputs. This process was scaled across the entire corpus to synthesize the *Essay-Syntax-Instruct* dataset. After generation, we performed a final filtering step, retaining only those essays with a word count between 100 and 700 to exclude outliers that were either too short to provide meaningful context or excessively long. This resulted in a final dataset of **8,320 high-quality examples**, each comprising a student essay paired with its corresponding structured syntax feedback.

10.2.3 Instruction Fine-Tuning and Knowledge Distillation

In the second stage of our methodology, the synthetically generated *Essay-Syntax-Instruct* dataset serves as the training corpus to fine-tune smaller, more accessible "student" models. This knowledge distillation process effectively transfers the specialized syntax feedback generation capability from the large, proprietary teacher model to efficient, open-source models, preparing them for practical deployment.
Model Selection: We selected three state-of-the-art, instruction-tuned LLMs known for their strong performance and accessibility:

- **Open-Source Models: Llama-2-7b-chat-hf** and **Llama-2-13b-chat-hf**. These models from Meta AI represent powerful, transparent, and widely used alternatives to proprietary systems.

- **Open-Source Model: Mistral-7B-Instruct-v0.2**. This model is renowned for its exceptional performance at a compact size, making it a highly efficient choice.

These models were chosen to demonstrate that our framework can effectively enhance capable open-source systems, thereby democratizing access to advanced educational technology.

Fine-Tuning Methodology: The fine-tuning process was designed to adapt these general-purpose models into specialized experts for the task of generating syntax feedback. The dataset was partitioned into a training set of 8,020 examples and a held-out test set of 300 examples.

Parameter-Efficient Fine-Tuning (PEFT): [120] To manage the significant computational cost associated with fully fine-tuning LLMs, we employed **Low-Rank Adaptation (LoRA)**. LoRA is a PEFT [120] technique that freezes the pre-trained model weights and injects smaller, trainable low-rank matrices into the Transformer layers. This dramatically reduces the number of trainable parameters, making the fine-tuning process computationally tractable while achieving performance comparable to full fine-tuning.

Training Configuration: The fine-tuning process was conducted using LoRA with a rank (r) of 32 and an alpha (α) of 64. The models were trained for three epochs with a total batch size of 16 and an initial learning rate of 3×10^{-4}, managed by a cosine scheduler with a warm-up ratio of 0.1. The entire training was executed on a server equipped with dual NVIDIA A6000 GPUs, accelerated by DeepSpeed and FlashAttention 2 to optimize efficiency and throughput. This meticulous fine-tuning strategy ensured that the student models were effectively specialized, transforming them into reliable and precise tools for automated syntax feedback.

10.3 The *Essay-Syntax-Instruct* Dataset

The cornerstone of this research is the *Essay-Syntax-Instruct* dataset, a novel and comprehensive collection of authentic student essays paired with structured, synthetically generated syntax feedback. This resource represents a significant contribution to the field by addressing the critical scarcity of specialized data for developing and evaluating automated writing support tools.

10.3.1 Dataset Characteristics and Statistics

The final dataset consists of 8,320 examples, curated to ensure quality and relevance. Each example contains a student-written essay (with placeholders replaced) and the corresponding seven-category syntax feedback generated by our pipeline.

Token Distribution and Capacity Planning: To understand the computational requirements for processing our dataset, we analyzed the length distributions of the source essays and the generated feedback. Figure 10.4 visualizes these distributions, calculated using the Llama-2 tokenizer. [123]

The distribution for the source essays (top panel) shows a peak around 200 tokens but has a long tail, reflecting the natural variation in student writing. The generated feedback (bottom panel) is considerably more concise, with a strong peak below 500 tokens. This analysis confirms that the models must be capable of handling moderately long contexts and provides the empirical basis for optimizing training parameters to ensure efficiency.

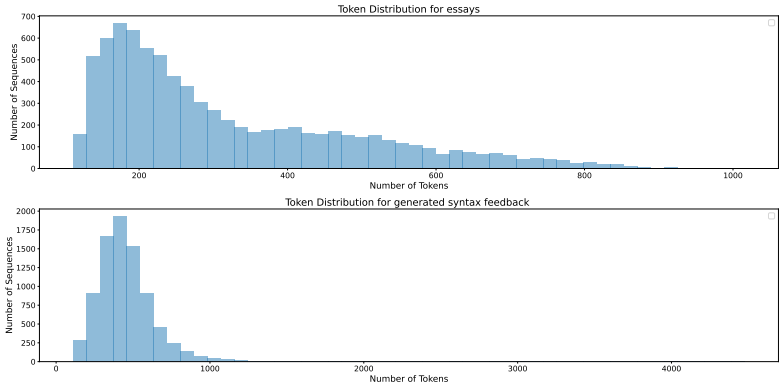


Figure 10.4: Token distributions for the source essays (top) and the generated syntax feedback (bottom) within the *Essay-Syntax-Instruct* dataset. These distributions are crucial for informing decisions about context window sizes and batching strategies during model training and inference.

10.3.2 Quality Validation of the Seed Data

The success of our knowledge distillation approach is fundamentally contingent on the quality of the synthetically generated seed dataset. If the teacher model produces flawed feedback, these flaws will be learned and amplified by the student models. Consequently, we conducted a rigorous human evaluation to validate the quality of the feedback generated by GPT-3.5-Turbo.

Human Evaluation Protocol: Due to the absence of a pre-existing gold-standard corpus for this task, we adopted a qualitative evaluation approach. We randomly selected a sample of 300 essay-feedback pairs from our synthesized dataset and had them evaluated by a panel of linguistics experts. The experts used a comprehensive five-point rating scale (A–E) to assess the pedagogical quality of the feedback:

- **RATING-A (Exceptional):** The feedback systematically identifies every error, provides clear and accurate corrections, and demonstrates a profound understanding of the text's nuances.

- **RATING-B (Robust):** The feedback pinpoints most errors with correct articulations and provides constructive criticism, though it might overlook minor issues. It reflects strong engagement with the content.

- **RATING-C (Adequate):** The feedback identifies a considerable number of errors and offers appropriate corrections, but the suggestions could be more tailored to enhance the text's overall quality.

- **RATING-D (Inconsistent):** The feedback catches some errors but does so inconsistently, and the suggestions lack the depth required to facilitate significant improvement.

- **RATING-E (Deficient):** The feedback fails to identify the main errors, offers few or no corrections, and does not contribute meaningfully to improving the text.

Seed Data Quality Results: The results of the human evaluation, depicted in Figure 10.5, affirm the high quality of our seed dataset. A remarkable **63.3%** of the feedback instances received a **Rating B**, and an additional **29.0%** achieved a **Rating A**. Combined, over 92% of the generated feedback was deemed robust or exceptional. The proportion of low-quality items (D and E) was negligible, at less than 1%. This result confirms that our sophisticated data preparation and prompt engineering strategy was highly effective, yielding a seed dataset of sufficient quality to serve as a reliable supervision signal for fine-tuning the student models.

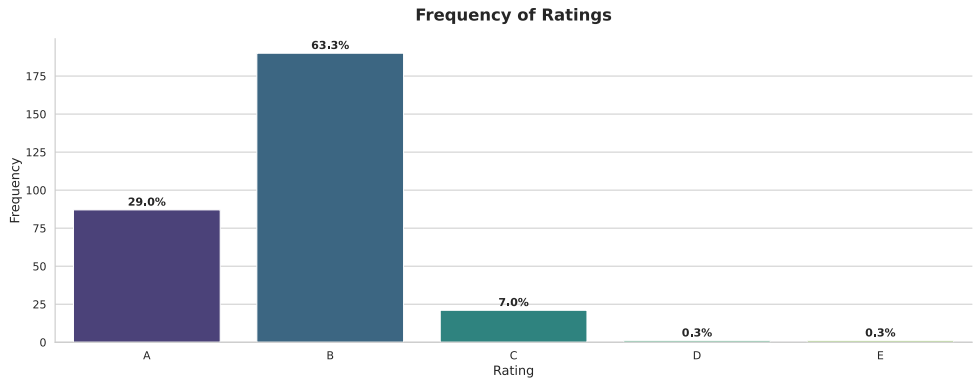


Figure 10.5: Human evaluation results for the seed feedback generated by the teacher model (GPT-3.5-Turbo). The distribution is heavily skewed toward high-quality ratings (A and B), validating the dataset's suitability for fine-tuning.

Illustrative Example: To provide a concrete illustration of the system's output, Figure 10.6 displays an example of an authentic student essay from the dataset alongside the structured syntax feedback generated by one of our fine-tuned models. The example showcases the model's ability to identify specific errors across multiple categories and provide precise, actionable corrections, demonstrating the practical utility of the generated feedback.

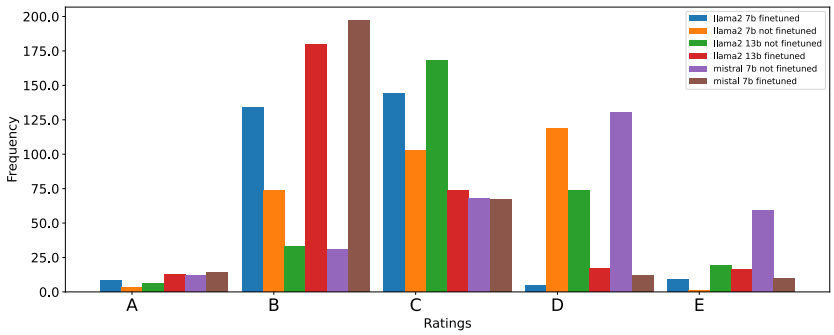


Figure 10.6: An illustrative example from the dataset, showing (a) a sample student essay and (b) the corresponding multi-category syntax feedback generated by a fine-tuned model.

10.4 Experimental Validation

This section presents a comprehensive experimental validation of our proposed methodology. We rigorously evaluate the performance of the three fine-tuned student models against their base versions to quantify the impact of our instruction-tuning process. The evaluation employs a dual approach, combining automatic metrics to measure lexical overlap with in-depth human evaluation to assess the crucial dimension of pedagogical quality.

10.4.1 Experimental Setup

Models. We evaluated three distinct LLMs: Llama-2-7b-chat-hf, Llama-2-13b-chat-hf, and Mistral-7B-Instruct. [127] Each model was assessed in two configurations: its *base* version (off-the-shelf, before fine-tuning) and its *fine-tuned* version (after training on *Essay-Syntax-Instruct*). This direct comparison allows for a precise measurement of the performance gains attributable to our distillation process.
Data Splits: The experiments were conducted on the held-out evaluation set of 300 essays, which were not seen by any model during the training phase, ensuring an unbiased assessment.
Automatic Metrics: We employed the standard *ROUGE* [125] metrics (ROUGE-1, ROUGE-2, and ROUGE-L) to quantify the lexical overlap between the feedback generated by the student models and the reference feedback from the teacher model (GPT-3.5-Turbo). While ROUGE scores are useful for measuring how closely the student models mimic the style and wording of the teacher, we acknowledge their limitations. They cannot fully capture semantic correctness, pedagogical value, or the quality of paraphrasing. Therefore, they serve primarily as a proxy for assessing alignment with the target format.
Human Evaluation: To transcend the limitations of automatic metrics, we conducted a comprehensive human evaluation. Two human raters, both master's students in lin-

guistics with relevant expertise, assessed the quality of the feedback generated by all six model configurations (three base, three fine-tuned). They used the same rigorous five-level rating scale (A–E) described previously to ensure consistency.

10.4.2 Feedback Generation Results

The experimental results demonstrate a clear and substantial improvement in performance for all models after fine-tuning, as evidenced by both automatic and human evaluation metrics.

Automatic Metrics: Table 10.1 summarizes the ROUGE scores. The results reveal a consistent and significant increase in scores across all three models after they were fine-tuned on our dataset. The base models, particularly the Llama-2 variants, show modest overlap with the reference texts. After fine-tuning, their ROUGE scores improve dramatically. The fine-tuned Mistral-7B model achieved the highest scores, indicating the strongest lexical alignment with the teacher model's outputs. These gains suggest that the fine-tuning process was highly effective at teaching the student models to adhere to the desired structure and phrasing of the feedback.

Group	Model Name	#	ROUGE-1	ROUGE-2	ROUGE-L
Base LLMs	LLAMA2-CHAT	7B	0.351	0.185	0.146
	LLAMA2-CHAT	13B	0.328	0.205	0.148
	MISTRAL-INSTRUCT	7B	0.366	0.144	0.214
Finetuned LLMs	LLAMA2-CHAT	7B	0.477	0.337	0.343
	LLAMA2-CHAT	13B	0.476	0.340	0.347
	MISTRAL-INSTRUCT	7B	**0.514**	**0.375**	**0.369**

Table 10.1: ROUGE score comparison before and after fine-tuning. The results show that fine-tuning with the *Essay-Syntax-Instruct* dataset delivers substantial gains in lexical overlap across all models.

Human Evaluation: The human evaluation results, which assess the crucial dimension of pedagogical quality, provide even more compelling evidence of the success of our methodology. The findings, detailed in Table 10.2, show a dramatic and positive shift in the quality of the generated feedback after fine-tuning.

The fine-tuned models demonstrate substantial improvements, achieving significantly higher percentages of A and B ratings while simultaneously reducing the frequency of poor D ratings. For instance, the fine-tuned Llama-2 13B model saw its proportion of high-quality feedback (Ratings A + B) increase from just 13.00% to an impressive 64.33%. Concurrently, its poor quality feedback (Rating D) plummeted from 24.67% to 5.67%. Similarly, the fine-tuned Mistral model increased its A + B ratings from 14.33% to 70.34%, while its D ratings fell from 43.33% to a mere 4.00%.

Qualitatively, the human evaluators noted that the base models often produced repetitive errors, incorrectly highlighted correct words, or failed to adhere to the specified seven-

category format. Fine-tuning significantly mitigated these issues, leading to feedback that was more accurate, constructive, and reliable.

Rating	Llama2 7B NF	Llama2 7B F	Llama2 13B NF	Llama2 13B F	Mistral 7B NF	Mistral 7B F
A	1.00	2.67	2.00	4.33	4.00	4.67
B	24.67	44.67	11.00	60.00	10.33	65.67
C	34.33	48.00	56.00	24.67	22.67	22.33
D	39.67	1.67	24.67	5.67	43.33	4.00
E	0.33	3.00	6.33	5.33	19.67	3.33

Table 10.2: Percentage of human evaluation ratings for syntax feedback generated by base (NF) and fine-tuned (F) models. Fine-tuning causes a dramatic shift from lower-quality ratings (C, D) to higher-quality ones (A, B).

10.4.3 Analysis and Discussion

The experimental results provide compelling validation for our framework. Several key observations emerge from the analysis. **The Absolute Necessity of Fine-Tuning.** The stark performance gap between the base and fine-tuned models underscores the necessity of task-specific adaptation. General-purpose instruction-tuned models, while capable, are insufficient to reliably adhere to the specific constraints, format, and nuanced requirements of generating high-quality pedagogical feedback. Fine-tuning is not merely beneficial; it is essential for transforming these models into effective educational tools. **Efficacy of the Knowledge Distillation Approach.** The high quality of the *Essay-Syntax-Instruct* dataset, validated by human experts, enabled a highly effective knowledge distillation process. The student models successfully learned to replicate the structure, style, and analytical quality of the teacher model, demonstrating the viability of using synthetically generated data for complex, specialized educational tasks. **The Critical Role of Data Pre-processing.** The success of our pipeline was heavily dependent on our novel placeholder replacement step. This highlights a broader principle: applying LLMs to real-world data requires careful and domain-aware pre-processing to handle artifacts that can otherwise derail model performance.

10.5 Conclusion

This chapter introduced an end-to-end framework for automating structured syntax feedback in student writing. Constraining generation to a predefined schema improved reliability and parsability; high-quality seed data from a strong teacher model enabled effective distillation into smaller open-source models; and rigorous human evaluation remained essential beyond automatic metrics. The approach negotiates quality–cost and precision–recall trade-offs by distilling from powerful models while calibrating model behavior to instructional goals. In practice, it supports scalable, immediate feedback that augments teachers and enables data-informed instruction through aggregated error patterns. Limitations include reliance on LLM-generated supervision, metric insufficiency for pedagogical adequacy, a focus limited to syntax, and uncertain generalization beyond ASAP (Grades 7–10, fixed prompts). Looking ahead, promising directions include multilingual

adaptation, deployment studies in authentic classrooms, comparative evaluations with traditional methods, integration of newer LLMs, and a streamlined fine-tuning pipeline for broader adoption. Together, the *Essay-Syntax-Instruct* dataset, privacy-preserving placeholder replacement, and instruction-tuned student models form a reusable blueprint for building reliable, affordable, and pedagogically useful syntax-feedback systems.

Part VI

Limitations, Future Work, and Conclusion

Chapter 11

Limitations

The "From Language to Learning" framework establishes a validated, scalable methodology for adapting general-purpose Large Language Models (LLMs) into specialized tools for multilingual educational content generation. The empirical results across diverse tasks—gamified learning, formal assessment, and writing assistance—and across typologically distinct languages—English, Italian, Arabic, Turkish, and Persian—demonstrate the efficacy of the proposed two-stage distillation approach. However, achieving the grand vision of truly personalized, equitable, and high-quality AI-driven education requires a candid assessment of the framework's current limitations. This chapter critically examines the constraints and challenges inherent in the methodology, the data resources utilized, the evaluation protocols employed, and the specific applications developed.

11.1 Methodological Limitations

The two-stage distillation process, while central to the framework's scalability and efficiency, introduces inherent methodological constraints.

11.1.1 Dependence on Teacher Model Capabilities

The foundational limitation of the distillation approach is its absolute reliance on the capabilities of the "teacher" LLM used in Stage 1 (Synthetic Data Generation). The quality ceiling of the generated *Instruct* datasets is determined by the teacher model's proficiency in the target language, its understanding of the pedagogical task, and its ability to adhere to complex instructions.

- **Quality Propagation:** If the teacher model misunderstands a nuance in the source text, hallucinates facts, or fails to generate pedagogically sound material (e.g., creating ambiguous distractors in MCQs or offering incorrect syntax feedback), these flaws are codified into the synthetic dataset. Consequently, the "student" model (Stage 2) learns to replicate these errors. In educational contexts, the cost of such inaccuracies is high, as it can lead to the propagation of misinformation and undermine learner trust. While the framework incorporates validation steps in data generation (e.g., in the crossword pipeline) and we also use human evaluation to assess the teacher model's outputs, these checks are imperfect and cannot guarantee the complete elimination of errors originating from the teacher.
- **Bias Amplification:** LLMs are known to harbor biases present in their pretraining data. The distillation process risks not only transferring these biases but

potentially amplifying them, as the student model is optimized specifically on the teacher's structured output. In an educational context, this is particularly concerning, as biased content related to history, culture, or language use can have detrimental effects on learners.

- **The "Proprietary Bottleneck":** This research utilized state-of-the-art proprietary models as teachers due to their superior instruction-following capabilities. While the framework produces open-source student models, the initial, critical step remains dependent on access to these closed systems. This reliance limits the full democratization of the framework, as researchers and educators in resource-constrained environments may face barriers in accessing or affording the necessary teacher models.

11.1.2 Prompt Sensitivity and Engineering Overhead

The generation of high-quality synthetic data in Stage 1 is highly sensitive to the design of the instructional prompts. As detailed in the methodology sections of Chapters 3 through 9, achieving the desired output structure, style, and quality required extensive iteration and sophisticated prompt engineering (e.g., the multi-stage cascade for Persian MCQs in Chapter 8, or the category-structured prompt for syntax feedback in Chapter 9).

- **Brittleness:** Minor changes in prompt wording, formatting, or the inclusion/exclusion of examples can lead to significant variations in the output quality and structure. This brittleness makes the process difficult to standardize perfectly across different tasks and languages.
- **Expertise Requirement:** Designing effective prompts requires expertise in both the pedagogical domain and LLM behavior. This overhead may hinder the adoption of the framework by educators who lack specialized technical skills.

11.1.3 The Semantic Gap in Distillation

Distillation aims to transfer task-specific competence, but it often focuses on mimicking the surface structure and lexical choices of the teacher model rather than transferring deep understanding or reasoning capabilities. The student models become proficient at generating content in the desired format (e.g., a crossword clue or an MCQ), but their ability to handle novel contexts, complex reasoning, or edge cases may not match that of the teacher. This is particularly relevant for tasks requiring nuanced judgment, such as providing feedback on complex writing errors.

11.2 Temporal, Computational, and Scope Constraints

In addition to the specific methodological and resource limitations discussed previously, this research was conducted under specific temporal and computational boundary conditions that influenced the experimental design. Furthermore, it is necessary to delineate the precise disciplinary scope of this work regarding pedagogical validity.

11.2.1 Temporal Constraints and Model Stationarity

A fundamental challenge in research involving Large Language Models (LLMs) is the rapid velocity of the field, where the definition of "State-of-the-Art" (SOTA) shifts on a monthly basis. This research was conducted longitudinally between the years 2022 and 2025. The experimental design and model selection were frozen at specific intervals to ensure internal consistency and the comparability of results across the diverse languages investigated.

At the time of the initial experimental design and execution for the English and Italian modules (Part III), models such as GPT-3 (Davinci/Curie) and BERT-base represented the industry standard for generative and discriminative tasks, respectively. As the research progressed to Turkish and Arabic (Part III) and Syntax Feedback (Part V), we integrated newer architectures such as Llama-2 (7B/13B) and GPT-3.5-Turbo, which represented the SOTA at the time of those specific experiments.

While the landscape has since evolved with the release of models such as GPT-4o and Llama-3, re-running the foundational experiments with these newer iterations would introduce significant threats to the internal validity of the thesis:

1. **Invalidation of Baselines:** Updating the generator models would necessitate a complete retraining of all baseline comparisons and learned validators (e.g., the BERT-based classifiers used in Chapter 4 and 7). Mixing model generations across chapters would render the cross-lingual comparisons incoherent.

2. **The Moving Target Problem:** Pursuing the absolute latest model is an asymptotic task in NLP. The primary contribution of this thesis is not the benchmarking of a specific model checkpoint (e.g., Llama-2 vs. Llama-3), but rather the validation of the *From Language to Learning* framework itself—specifically, the two-stage distillation pipeline and the efficacy of the Instruct-series datasets.

3. **Historical Validity:** The results presented herein serve as a rigorous historical baseline for the capabilities of models in the 2022–2024 era. They demonstrate that even with models of that generation, the proposed distillation framework yields educationally valid content, thereby suggesting that the framework's utility will only scale with the integration of newer, more capable backbones.

Therefore, the models utilized should be viewed through the lens of a "time capsule" representing the optimal available tools at the moment of experimentation. Future work may easily swap the student backbones for newer architectures (e.g., Llama-3 or Mistral-v3) using the datasets released in this thesis, but such updates are outside the scope of this foundational study.

11.2.2 Computational Constraints and Ablation Scope

The scope of this thesis is ambitiously multilingual, covering five typologically distinct languages (English, Italian, Arabic, Turkish, Persian) across three distinct pedagogical tasks. This breadth imposes significant computational demands that necessitated strategic trade-offs regarding hyperparameter optimization and ablation studies.

Prioritization of Parameter-Efficient Fine-Tuning (PEFT)

Due to significant computational constraints—specifically the reliance on academic hardware (e.g., dual NVIDIA A6000 GPUs) rather than industrial-scale clusters—we prioritized Parameter-Efficient Fine-Tuning (PEFT), specifically Low-Rank Adaptation (LoRA), over full model fine-tuning. Full fine-tuning of 7B and 13B parameter models for every language and task variation would have exceeded the available computational budget and training time.

While full ablations (systematically varying rank r, alpha α, and learning rates) would theoretically provide a more granular view of model behavior, prior literature strongly suggests that LoRA is a competitive and reliable proxy for full fine-tuning in low-resource settings. Specifically, Hu et al. (2021) demonstrated that LoRA can match or exceed the performance of full fine-tuning with a fraction of the trainable parameters. By freezing the pre-trained weights and injecting trainable rank decomposition matrices, we ensured that the "student" models retained their general linguistic competence while adapting to the specific output formats (e.g., JSON schemas for MCQs) required by our framework.

Justification of Heuristics over Exhaustive Search

Consequently, the hyperparameters selected for our experiments (typically $r = 16/32$, $\alpha = 32/64$, as detailed in Sections 8.2.3, 9.2.3, and 10.2.3) were derived from established heuristics in the PEFT literature rather than exhaustive grid searches for every single language pair. This decision allowed us to allocate resources toward expanding the *linguistic coverage* of the framework—demonstrating its universality across agglutinative (Turkish) and RTL (Arabic/Persian) languages—rather than optimizing marginal performance gains for a single language. The consistent performance improvements observed across all five languages (e.g., the shift from E-rated to A-rated outputs in human evaluations) validate that these hyperparameter choices were sufficient to demonstrate the efficacy of the distillation framework, even if they do not represent the theoretical global maximum of model performance.

11.3 Data Resource Limitations

The creation and utilization of the *Instruct* series datasets, while a key contribution, also present several limitations.

11.3.1 The Nature of Synthetic Data

While the synthetic datasets enable scalable fine-tuning, they may not fully capture the richness, diversity, and nuance of authentic, human-authored educational materials.

- **Lack of Authentic Errors:** In the context of syntax feedback, the feedback was generated based on existing student essays. However, for content generation tasks (Parts II and III), the models are trained on "perfect" synthetic examples. They are not explicitly trained on recognizing or handling the types of errors or misconceptions that students typically exhibit.

- **Stylistic Homogeneity:** Synthetic data generated by a single teacher model often exhibits less stylistic variation than data authored by multiple human experts. This homogeneity can lead to student models that produce repetitive or formulaic content.

11.3.2 Source Material Biases and Coverage

The selection of raw, unstructured text used as the input for Stage 1 significantly influences the resulting educational content.

- **Wikipedia Reliance:** A substantial portion of the data for crossword and question generation across various languages (English, Italian, Arabic, Turkish, Persian) was derived from Wikipedia. This reliance inherits Wikipedia's known limitations, including systemic biases in coverage (e.g., favoring Western perspectives), stylistic conventions (encyclopedic tone), and potential inaccuracies. Educational content derived solely from Wikipedia may not align perfectly with specific regional curricula or pedagogical objectives.
- **Domain Skew:** As observed in the category distributions of the datasets, the generated content is often skewed towards domains that are well-represented in the source material (e.g., Geography, Science). Niche topics or vocational skills may remain underserved.

11.3.3 Language Coverage and Typological Diversity

While the framework was validated across five languages (English, Italian, Arabic, Turkish, Persian), representing different language families and orthographies, this selection is by no means exhaustive.

- **Generalizability to Other Typologies:** The efficacy of the framework for languages with vastly different typological features—such as tonal languages (e.g., Mandarin), polysynthetic languages, or languages with very limited digital resources remains unproven. The challenges posed by agglutination in Turkish or right-to-left scripts in Arabic required specific adaptations, suggesting that new typologies will necessitate further bespoke engineering.
- **The "Ultra-Low-Resource" Gap:** The framework assumes the availability of sufficient raw text in the target language to serve as source material. It does not address the challenge of "ultra-low-resource" languages where such digital text is scarce.

11.4 Evaluation Limitations

The evaluation protocols employed in this thesis, combining automatic metrics and human assessments, provide robust evidence of the framework's efficacy but are also subject to constraints.

11.4.1 The Inadequacy of Automatic Metrics

Automatic metrics such as ROUGE and BERTScore were used extensively to provide directional evidence of improvement after fine-tuning. However, these metrics are fundamentally limited in assessing educational quality.

- **Proxy Measures:** ROUGE measures lexical overlap, and BERTScore measures semantic similarity against a reference. Neither metric can reliably assess pedagogical utility, factual correctness, the plausibility of distractors, the clarity of feedback, or the appropriateness of the difficulty level. As noted, high-quality paraphrasing often results in low ROUGE scores, complicating interpretation.
- **Reference Dependence:** These metrics require high-quality reference outputs (often the seed data generated by the teacher model). Evaluating against a synthetic reference can artificially inflate scores and does not guarantee alignment with human pedagogical standards.

11.4.2 Constraints of Human Evaluation

Human evaluations were crucial for assessing the quality of the generated content using detailed rubrics (A-E scales). However, this approach has limitations.

- **Scalability and Cost:** Human evaluation is time-consuming and expensive, limiting the scale at which it can be applied. The evaluations in this thesis were conducted on subsets of the generated data.
- **Subjectivity and Variance:** Despite the use of detailed rubrics and trained annotators, human judgment remains inherently subjective. Inter-annotator agreement, while generally high in this study, is rarely perfect, introducing variance in the results.
- **Focus on Content Quality, Not Efficacy:** The human evaluations assessed the quality of the generated artifacts (clues, questions, feedback) in isolation. They did not measure the effectiveness of these artifacts in facilitating actual student learning.

11.4.3 Lack of Longitudinal and Classroom Studies

The most significant limitation of the evaluation strategy is the absence of longitudinal studies conducted in real-world classroom settings. The thesis validates the *potential* of the framework to generate high-quality content but does not provide empirical evidence of its *impact* on learning outcomes, student engagement, or teacher workload reduction.

11.4.4 The Boundary Between Engineering and Psychometrics

A critical distinction must be drawn between the *engineering of educational content generation systems* and the *psychometric validation of learning outcomes*. This thesis situates

itself firmly within the former: it is an engineering and computational linguistics endeavor aimed at automating the production of structured educational artifacts.

Structural vs. Psychometric Validity

The evaluations presented in this work (Appendices A and B) assess *structural validity* and *expert-perceived quality*. We measure whether the generated MCQs, crosswords, and feedback adhere to pedagogical constraints: Are the distractors plausible? Is the feedback syntactically accurate? Is the content grounded in the source text? These are necessary prerequisites for educational utility.

However, *psychometric validation*—which involves Item Response Theory (IRT), difficulty analysis, and discrimination index calculation—requires longitudinal deployment with real student populations. As noted in the Related Work (Section 2.8.2), valid Automatic Item Generation (AIG) traditionally requires rigid cognitive models. While our framework generates items that human experts rate as "high quality," we do not claim to have validated the *learning efficacy* of these items in a classroom setting.

Ethical and Logistical Barriers to Student Trials

Conducting longitudinal studies with students introduces significant ethical and logistical complexities, including Institutional Review Board (IRB) approval, privacy protection for minors, and the requirement for long-term curricular integration. Given the timeline of this PhD research and its primary focus on the *computational methodology* of multilingual generation, such deployment was deemed outside the scope of the current work.

The contribution of this thesis is the creation of the *pipeline* and the *datasets* (Instruct-series) that make such future psychometric studies possible. By lowering the cost of generating high-quality draft items, this framework enables educational researchers to move to the next phase: deploying these AI-generated materials in controlled settings to measure their impact on learning gains. Thus, the lack of student-based psychometric data should be viewed not as a flaw in the engineering methodology, but as a delineation of disciplinary scope, identifying a critical and immediate direction for Future Work. Explaining the Content to the Reviewer (Defense Strategy)

11.5 Task-Specific Limitations

Each application domain presented unique challenges that were not fully resolved.

11.5.1 Educational Crossword Generation (Part II)

- **Difficulty Control:** The control over the difficulty of the generated clues was rudimentary, relying primarily on prompt instructions for brevity and clarity. The framework lacks a mechanism for precisely calibrating clue difficulty to specific learner proficiency levels.

- **Leakage Detection:** Detecting answer leakage (where the clue contains the answer or a morphological variant) relied on heuristics and validators. This was particularly challenging in morphologically rich languages like Turkish and Arabic, and residual cases required manual cleanup.
- **Grid Layout Optimization:** The schema generation used a heuristic scoring function. While effective in producing dense grids, it may not always find the optimal layout or incorporate aesthetic considerations (e.g., symmetry) valued in some contexts.

11.5.2 Question Generation (Part III)

- **Depth of Reasoning:** The generated MCQs and SAQs primarily targeted factual recall and surface-level comprehension. The framework struggled to reliably generate questions requiring deep reasoning, multi-hop inference, or complex synthesis of information.
- **Distractor Plausibility:** Automating the generation of plausible yet unambiguously incorrect distractors remains a fundamental challenge in QG. While the refinement prompts improved distractor quality, subtle semantic overlaps or ambiguity occasionally persisted.

11.5.3 Automated Syntax Feedback (Part IV)

- **Scope Limitation (Syntax Only):** The feedback system was intentionally constrained to syntax-level issues. It does not address higher-order writing concerns such as organization, argumentation, rhetoric, or content development, which are often more critical for improving overall writing quality.
- **Risk of Over-Correction and Stylistic Bias:** Despite efforts to encourage minimal edits, the models occasionally over-corrected, imposing unwarranted formality or misinterpreting stylistic choices as errors. This risks undermining the learner's voice or penalizing non-standard language varieties.
- **Explanation Quality:** While the framework aimed to provide clear pedagogical explanations, the generated rationales were sometimes generic or lacked the specificity required for effective instruction.

In summary, while the "From Language to Learning" framework successfully demonstrates a scalable methodology for multilingual educational content generation, these limitations highlight the critical need for continued research, particularly in improving the robustness of the methodology, diversifying data sources, enhancing evaluation protocols, and validating the impact in real-world educational contexts.

Chapter 12

Future Work

The development and validation of the "From Language to Learning" framework have established a robust foundation for the scalable generation of multilingual educational content. However, this research also opens up numerous avenues for future exploration. The limitations identified in the previous chapter serve as catalysts for innovation, suggesting concrete directions to enhance the core framework, expand its pedagogical capabilities, address the challenges of low-resource languages more deeply, and rigorously validate its real-world impact.

12.1 Enhancing the Core Framework

Future work should focus on improving the robustness, efficiency, and autonomy of the two-stage distillation methodology.

12.1.1 Iterative Distillation and Self-Refinement

To mitigate the dependence on the initial quality of the teacher model and the potential propagation of errors, future research should explore iterative refinement loops.

- **Critic Models and RLAIF:** Implementing mechanisms where a specialized "critic" model, or the teacher LLM itself via techniques like Reinforcement Learning from AI Feedback (RLAIF), evaluates the generated synthetic data against pedagogical rubrics. This feedback can be used to prompt the teacher to revise the data iteratively, improving the quality of the *Instruct* datasets before student training.
- **Self-Correction and Consistency Checking:** Enhancing the student models with the ability to self-correct their outputs based on consistency checks, external knowledge retrieval, or explicit pedagogical rules. For instance, an MCQ generator could verify the uniqueness of the correct answer or the plausibility of distractors before finalizing the output.

12.1.2 Multi-Modal Educational Content Generation

The current framework is exclusively text-based. However, effective education relies heavily on multi-modal materials.

- **Vision-Language Integration:** Extending the framework to incorporate vision-language models. This would enable the generation of questions grounded in images,

diagrams, or videos (e.g., generating a quiz about a biological diagram or a historical map), as well as the generation of illustrative images to accompany textual explanations.

- **Incorporating Audio and Interaction:** Integrating speech recognition and synthesis capabilities to generate listening comprehension exercises, pronunciation feedback, and interactive simulations, which are crucial for comprehensive learning experiences.

12.1.3 Human-in-the-Loop Optimization and Active Learning

To ensure the alignment of the generated content with pedagogical needs and to continuously improve the models, human feedback must be integrated more systematically.

- **Active Learning Pipelines:** Developing active learning strategies that identify the most informative instances for human review. Instead of random sampling, the system could prioritize ambiguous, low-confidence, or novel outputs for teacher validation, optimizing the use of expert effort.
- **Continual Fine-Tuning from Feedback:** Establishing pipelines for continual fine-tuning of the student models based on the edits and feedback provided by educators during the content authoring process (RLHF). This would allow the models to adapt to specific curricular requirements and teaching styles.

12.2 Expanding Pedagogical Capabilities

The framework should evolve beyond isolated content generation to support more complex educational scenarios and foster higher-order thinking skills.

12.2.1 Advanced Assessment and Difficulty Control

Moving beyond factual recall is a critical next step.

- **Generating Higher-Order Questions:** Developing techniques for reliably generating questions that require analysis, synthesis, evaluation, and multi-hop reasoning. This may involve integrating chain-of-thought prompting or symbolic reasoning modules to ensure the validity of the required inferences.
- **Calibrated Difficulty Control:** Implementing robust mechanisms for controlling and predicting the difficulty of the generated content. This could involve training difficulty classifiers based on Item Response Theory (IRT) principles or leveraging learner performance data to adapt the difficulty level dynamically.
- **Automated Grading of Open-Ended Responses:** Extending the framework to the automated grading of short-answer and essay responses, providing not just a score but constructive feedback aligned with specific rubrics, moving beyond the syntax-only focus of Chapter 9.

12.2.2 Automated Curriculum and Lesson Planning

The ultimate goal is to assist educators in the end-to-end design of learning experiences.

- **Curriculum Alignment:** Developing methodologies for ingesting structured curricula (e.g., syllabi, learning standards) and using them to guide the generation of educational materials, ensuring pedagogical relevance and appropriate scoping.
- **Adaptive Sequencing:** Developing algorithms that can sequence the generated content (explanations, practice items, assessments) into personalized learning paths adapted to the individual learner's progress, knowledge state, and misconceptions.
- **Lesson Plan Synthesis:** Combining different types of generated content into coherent lesson plans that include learning objectives, activities, materials, and assessments.

12.2.3 Interactive Tutoring Systems and Explainability

Leveraging the specialized models to power interactive learning environments.

- **Socratic Dialogue Agents:** Utilizing the fine-tuned models as the backbone for conversational tutoring systems that can engage in Socratic dialogue, asking probing questions, providing adaptive scaffolding, and guiding learners through complex problem-solving processes.
- **Explainable Feedback:** Enhancing the feedback generation models to provide detailed, context-aware explanations and rationales for the suggested corrections or assessments. This is crucial for building learner trust and facilitating understanding.

12.3 Addressing Multilingual and Low-Resource Challenges

A core objective of this research is to promote educational equity. Future work must continue to address the challenges of under-resourced languages.

12.3.1 Deeper Integration of Linguistic Knowledge

To address the challenges posed by morphological complexity (e.g., Turkish, Persian) and linguistic diversity.

- **Morphology-Aware Generation:** Integrating morphological analyzers and lexicons into the pipeline to ensure grammatical correctness, prevent leakage (crucial for crosswords), and generate plausible distractors (crucial for QG) in agglutinative and highly inflected languages.

12.3.2 Ultra-Low-Resource Adaptation

Developing techniques for languages where digital resources are extremely scarce.

- **Advanced Cross-Lingual Transfer Learning:** Leveraging knowledge learned from high-resource languages to improve performance in low-resource languages. This could involve advanced multilingual pre-training or zero-shot cross-lingual adaptation techniques.
- **Machine Translation Bootstrapping:** Utilizing machine translation to create synthetic parallel data or to translate existing educational resources into the target language, followed by careful post-editing and adaptation strategies to mitigate translation errors.

12.3.3 Dialectal and Cultural Adaptation

Ensuring that the generated content is culturally relevant and respects linguistic diversity.

- **Handling Code-Switching and Dialects:** Adapting the framework to support regional dialects, non-standard language varieties, and code-switching phenomena, which are often marginalized in educational technology.
- **Culturally Aware Content Generation:** Developing mechanisms to ensure that the generated content is culturally appropriate, respectful, and relevant to the learners' context. This requires incorporating cultural knowledge and guidelines into the generation process and engaging local communities in the development and validation of the tools.

12.4 Validation and Real-World Impact

Finally, future research must bridge the gap between content generation and measurable educational impact.

12.4.1 Longitudinal Efficacy Studies

Conducting rigorous studies to evaluate the effectiveness of the generated content in real-world educational settings.

- **Randomized Controlled Trials (RCTs):** Implementing RCTs to compare the learning outcomes of students using the AI-generated content with those using traditional materials across diverse contexts (languages, subjects, learner populations).
- **Impact on Teacher Workload and Practice:** Assessing the impact of the framework on teacher workload, job satisfaction, and the time allocated to higher-level pedagogical activities.

12.4.2 Ethical Considerations and Bias Mitigation

As AI plays an increasingly significant role in education, it is imperative to address the ethical implications.

- **Fairness and Equity Auditing:** Developing robust methods for auditing the generated content for fairness, bias (e.g., gender, cultural, socioeconomic), and safety.
- **Transparency and Accountability:** Ensuring that the generation process is transparent and that mechanisms for accountability are in place when errors or biases occur.
- **Data Privacy and Security:** Implementing strong safeguards to protect student data privacy, particularly when dealing with sensitive information such as writing samples.

By pursuing these directions, the "From Language to Learning" framework can evolve from a promising methodology into a transformative force in global education, making personalized, high-quality learning accessible to everyone, everywhere.

Chapter 13

Conclusion

This thesis undertook a systematic study at the intersection of advanced natural language processing and global education. It addressed the persistent bottleneck of producing scalable, personalized, and high-quality educational content—especially for school systems and languages that remain underserved. The digital language divide, rooted in data scarcity, economic misalignment, and linguistic complexity, has concentrated the benefits of AI-enabled learning within a small set of dominant languages. To counter this, the thesis introduced and validated *From Language to Learning* (FLL): a unified, LLM-driven framework that reliably adapts general-purpose language models to multilingual education at scale. FLL operationalizes a task-specific adaptation of distillation that first converts raw text into pedagogically faithful supervision and then distills that expertise into efficient, open models deployable in real classrooms. Across this project, we produced **11** publications, released **11** datasets (including **7** in the *Instruct* series), and trained/evaluated **29** fine-tuned generator models across **5** languages.

13.1 The "From Language to Learning" Framework: A Synthesis

FLL formalizes a two-stage process:
Stage 1 — Expert Data Generation (Teacher). A state-of-the-art teacher LLM is prompted with rubric-based, multi-step instructions to transform unstructured text into structured, high-fidelity educational artifacts—the *Instruct* series datasets. Schemas, guardrails (e.g., JSON formats, no answer leakage), and refinement passes encode pedagogy, difficulty, and textual grounding.
Stage 2 — Instruction Fine-Tuning (Student). These synthetic datasets power instruction fine-tuning of smaller, open-source models using parameter-efficient methods (PEFT/LoRA). The result is a family of specialized student models aligned to concrete tasks and languages while preserving accessibility, auditability, and deployability under modest compute.
The framework is guided by five objectives: *Scale*, *Pedagogical Fidelity*, *Multilingual Equity*, *Efficiency & Openness*, and *Reliability*. Together they constrain both data generation and modeling to produce *learning-ready* outputs rather than generic text.

13.2 Empirical Validation and Answers to the Research Questions

FLL was validated across three application tiers—gamified learning, formal assessment, and feedback—covering five languages (English, Italian, Arabic, Turkish, Persian). Specialized students consistently outperformed general-purpose baselines, answering the MRQ and sub-questions posed in Chapter 1. The experimental program spans **five languages** and three pedagogical domains.

MRQ. *How can general-purpose LLMs be systematically and scalably adapted to generate diverse, high-quality, pedagogically varied educational content, particularly for under-resourced languages?*

Answer. Through FLL's two-stage distillation: generate task-specific *Instruct* datasets with a teacher LLM, then instruction-fine-tune compact open models (PEFT/LoRA). Scalability comes from transforming abundant raw text into *structured* pedagogy; diversity and fidelity follow from applying the same blueprint across disparate tasks and languages.

RQ1 — Gamified Learning via Multilingual Crosswords

FLL generated source-grounded, educational crosswords (EN/IT/AR/TR) using *paired* resources per language: (i) answer-only clue–answer pairs and (ii) text-grounded datasets. Fine-tuned students markedly improved clarity, grounding, and format reliability for clue–answer pairs, while handling RTL scripts and agglutinative morphology without bespoke engineering [3, 4, 5, 6, 7, 8, 9, 10].

RQ2 — Formal Assessment in Low-Resource Languages

FLL extended to assessment by producing MCQs and SAQs in Persian and Turkish using `PersianMCQ-Instruct` and `Turkish-Quiz-Instruct`. Human evaluation showed transformative gains (e.g., Turkish SAQ quality rose from 0% to >70% *excellent*; Persian MCQs achieved >95% high-quality outputs). In the Persian MCQ task, the overall human score for a fine-tuned 8B student increased from 3.31 to 4.51/5, demonstrating assessment-grade reliability [11, 12].

RQ3 — Nuanced Pedagogical Intervention via Syntax Feedback

For English student essays, `Syntax-Instruct` enabled students to deliver precise, actionable, and minimally invasive syntax feedback. LoRA-based students reduced over-correction and improved locality and justification, increasing the share of good (B-or-better) feedback substantially [13].

13.3 Released Resources and Model Suite

A centralized index of all resources detailed below, including the specific prompt templates and reproducibility scripts for the English, Italian, Arabic, Turkish, and Persian experiments, is maintained in the project repository: `https://github.com/KamyarZeinalipour/From-Language-to-Learning`.

Datasets. We publicly released **11** datasets in total:

- **Crosswords (8 datasets = 4 languages × 2 each):** for each of EN/IT/AR/TR we released (a) answer-only clue–answer pairs and (b) a text-grounded "Clue-Instruct" dataset.
- **Assessment (2):** `Turkish-Quiz-Instruct` (TR; MCQ/SAQ) and `PersianMCQ-Instruct` (FA; MCQ).
- **Writing feedback (1):** `Essay-Syntax-Instruct` (EN).

Within these, the **Instruct-series subset** comprises **7** datasets (the four crossword *Clue-Instruct* variants + Turkish-Quiz-Instruct + PersianMCQ-Instruct + Essay-Syntax-Instruct) [5, 12, 11, 13].

Models. Across all studies we trained and evaluated **29** distinct fine-tuned *generator* checkpoints (not counting validators/classifiers), spanning LLaMA/Llama 3, Mistral, MPT, Gemma, and GPT-3/3.5 families, primarily using PEFT/LoRA. Multiple students were introduced per task and language rather than a single model per paper, reflecting careful sweeps over base families and parameter scales.

Publications. This body of work produced **11** publications across conferences, workshops, and journals, covering puzzles/crosswords [3, 4, 5, 6, 7, 8, 9, 10], assessment [11, 12], and feedback [13].

13.4 Reiteration of Contributions

Methodological. A scalable, task-specific adaptation of distillation that turns general LLMs into specialized, trustworthy educational tools.

Empirical. The first multilingual demonstration of a unified blueprint across gamified learning, assessment, and writing support, with consistent gains over base models in five typologically diverse languages.

Resources. Public release of **11** datasets in total (including an **Instruct-series** subset of **7**) and **29** fine-tuned generator models across **5** languages, enabling reproducible research and practical deployment.

Practical. A validated path to democratize AI-powered education—open, auditable, and deployable under real constraints.

13.5 Broader Implications and Final Remarks

In a world demanding personalization at scale, FLL demonstrates that *systematic*, *pedagogically grounded* AI can deliver classroom-ready value, not just plausible text. By prioritizing open models and low-resource languages, this work helps narrow the digital language divide and offers a replicable process that communities can adapt to their curricula, cultures, and constraints. Ethics and reliability remain central: the goal is to *augment* educators—ensuring transparency, provenance, and alignment with learning objectives. By prioritizing open models and low-resource languages, this work helps narrow the digital language divide. While no probabilistic model can fully guarantee grounding, this framework significantly mitigates hallucination risks by constraining generation to the provided source text. **In sum**, this thesis provides a validated, scalable framework that closes the gap between the potential of LLMs and the realities of multilingual education. With **11** publications, **11** released datasets (including **7** Instruct-series resources), and **29** specialized fine-tuned generator models across **5** languages, FLL charts a practical route toward personalized, high-quality learning for all—regardless of language or location.

Part VII

Appendix

Appendix A

Dataset Cards

This appendix provides detailed specifications for the datasets created and utilized in this thesis. It covers the *Instruct* series (generated via the two-stage distillation framework) and the legacy repositories curated for answer-only generation.

A.1 Clue-Instruct (English Crosswords)

- **Description:** A dataset of context-grounded crossword clues in English, linking keywords to specific source texts.
- **Provenance:** Data was extracted from English Wikipedia. The pipeline targeted the initial sections of articles, focusing on concise definitions and key facts.
- **Filtering:**
  - Articles were screened based on metadata, specifically high view counts or "Top" importance ratings.
  - Context length was constrained to 30–1000 words.
  - Keywords were constrained to 3–20 characters and a maximum of 3 words.
- **Generation Method:** Clues were synthesized using GPT-3.5-Turbo. The prompt enforced negative constraints to prevent keyword leakage in the clues.
- **Statistics:**
  - **Total Examples:** 44,075 unique entries.
  - **Total Clues:** 132,225 context-grounded clues.
  - **Categories:** 20 broad categories (e.g., Science, Geography, Applied Science).
- **Usage:** Used for training and validation of LLaMA2 (7B–13B) and MPT models for the text-conditioned path.

A.2 Italian-Clue-Instruct

- **Description:** The first text-aligned corpus for Italian crossword generation, capable of controlling for specific syntactic styles.
- **Provenance:** Sourced from 88,403 Italian Wikipedia articles.

- **Filtering:**
  - Filtered for popularity and relevance.
  - Articles shorter than 50 words were removed.
  - Keywords restricted to 3–20 characters and a maximum of 2 words.
  - Refined to 11,413 high-quality articles, from which 5,000 were randomly selected for generation.
- **Generation Method:** Generated using GPT-4o. The process utilized four distinct prompt templates to vary syntactic style: Bare Noun Phrase, Definite Determiner Phrase, Copular Schema, and General-Purpose.
- **Statistics:**
  - **Total Entries:** Over 30,000 clues.
  - **Categories:** 29 thematic categories, with strong representation in Entertainment, Geography, and History.

A.3 T4TAC (Turkish Educational Crosswords)

- **Description:** A text-aligned corpus for Turkish, specifically designed to handle agglutinative morphology and prevent clue leakage.
- **Provenance:** Extracted from approximately 180,000 Turkish Wikipedia pages.
- **Filtering:**
  - Filtered for popularity and context length ($>$ 50 words).
  - Keywords restricted to 3–20 characters, containing no numerics or special characters.
  - Refined to 9,855 high-quality pages.
- **Generation Method:** Synthesized using GPT-4-Turbo with "leakage-aware" prompts designed to handle Turkish agglutination and prevent the answer stem from appearing in the clue.
- **Statistics:**
  - **Total Clues:** Over 35,000.
  - **Splits:** 8,000 items (training) / 670 items (test).

A.4 Arabic-Clue-Instruct

- **Description:** A text-aligned, category-aware corpus for Arabic, supporting right-to-left script constraints.
- **Provenance:** Extracted from approximately 211,000 Arabic Wikipedia articles.
- **Filtering:**
  - Articles fewer than 50 words were discarded.
  - Keywords capped at 2 words (3–20 characters).
  - Refined to 14,497 content-keyword pairs.
- **Generation Method:** Synthesized using GPT-4-Turbo with strict constraints on clue length (max 4 words) to ensure conciseness.
- **Statistics:**
  - **Total Clues:** 54,196.
  - **Splits:** 14,000 (training) / 497 (test).
  - **Categories:** 20 categories, primarily Geography, Science, and Society.

A.5 Turkish-Quiz-Instruct (MCQ & SAQ)

- **Description:** A dataset pairing Turkish educational texts with Multiple-Choice (MCQ) and Short-Answer Questions (SAQ).
- **Provenance:** Scraped from reputable educational websites *bikifi.com* and *basarisiralamalari.com*, covering secondary school curricula.
- **Filtering:**
  - Removal of navigation bars, ads, emojis, and hyperlinks.
  - Excessively short text fragments were eliminated to ensure sufficient context.
- **Generation Method:** Synthesized using GPT-4-Turbo via a structured prompt enforcing distractor plausibility, grammatical parallelism, and unambiguous keys.
- **Statistics:**
  - **Splits:** 8,000 passages (training) / 260 passages (evaluation).
  - **Subjects:** History, Chemistry, Geography, Literature, Biology, Philosophy.

A.6 PersianMCQ-Instruct

- **Description:** A large-scale collection of Persian educational texts paired with MCQs, created to overcome low-resource constraints.
- **Provenance:** Sourced from Persian Wikipedia lists, specifically "Featured," "Good," and "Vital" articles. The initial crawl yielded 8,894 pages.
- **Filtering:**
  - Pages fewer than 100 words were removed.
  - Contexts were capped at 500 words.
  - Refined to 4,159 high-quality passages.
- **Generation Method:** Generated via a three-stage prompt cascade (Text Augmentation → MCQ Generation → Refinement) using GPT-4o to increase complexity and pedagogical value.
- **Statistics:**
  - **Total MCQs:** Over 12,000 (3 per passage).
  - **Splits:** 12,000 (training) / 476 (evaluation).

A.7 Essay-Syntax-Instruct

- **Description:** A dataset of authentic student essays paired with structured syntax feedback across seven error categories.
- **Provenance:** Derived from the ASAP (Automated Student Assessment Prize) dataset, covering Grades 7–10.
- **Filtering:**
  - **Placeholder Replacement:** Anonymization tokens (e.g., `@PERSON1`) were replaced with contextually plausible surrogates using an LLM to restore linguistic naturalness.
  - Filtered for word counts between 100–700 words.
- **Generation Method:** Feedback synthesized by GPT-3.5-Turbo using a structured prompt covering 7 syntax categories (e.g., Modifiers, Articles, Punctuation).
- **Statistics:**
  - **Total Pairs:** 8,320.
  - **Splits:** 8,020 (training) / 300 (test).

A.8 Legacy & Auxiliary Datasets (Answer-Only)

These datasets were curated to support the "Answer-Only" generation paths (Path B) and to train validation classifiers.

- **Century-Scale Clue–Answer Corpus (English):**
  - Sourced from print and web archives spanning 1913–2021.
  - Contains 7,327,448 clue-answer pairs.
- **Legacy Italian Clue-Answer Collection:**
  - Sourced from websites (e.g., *dizy.com*) and PDF archives.
  - Contains 125,600 unique pairs.
- **TAC (Turkish Answer-Clue):**
  - Sourced from expert constructors and newspaper archives (e.g., *Habertürk*).
  - Contains 252,576 pairs (187,495 unique entries).
- **AR_CW (Arabic Legacy Repository):**
  - Sourced from *Al-Joumhouria Journal* and *Al-Ghad* via manual extraction and OCR.
  - Contains 57,706 entries (25,908 unique pairs).

Appendix B

Human Evaluation Protocols and Experimental Design

This appendix provides a comprehensive technical specification of the human evaluation protocols employed throughout this thesis. While automatic metrics (such as ROUGE, BLEU, and BERTScore) provide necessary proxies for lexical overlap and semantic similarity, they are fundamentally insufficient for assessing pedagogical utility. Educational content requires evaluation along complex dimensions including validity, instructional value, cultural appropriateness, and the strict absence of answer leakage.

To address these nuances, we designed distinct evaluation campaigns for the three primary pillars of the framework: Educational Crossword Generation (Part II), Question Generation (Part III), and Syntactic Feedback (Part IV). The following sections detail the recruitment of annotators, the specific instructions provided, the rating taxonomies, and the methods used to calculate agreement and aggregate results.

B.1 Protocol A: Educational Crossword Generation

This protocol was utilized to validate the *Instruct* series datasets and the fine-tuned models for English, Italian, Turkish, and Arabic crosswords (Chapters 4–7).

B.1.1 Evaluation Rationale and Objectives

The primary objective for crossword clue evaluation was to determine if the generated output functioned as a valid puzzle component. Unlike open-ended text generation, crossword clues operate under strict functional constraints:

1. **Validity:** The clue must unambiguously point to the answer key.
2. **Contextual Grounding:** The clue must be derived from the provided source text rather than general external knowledge.
3. **Non-Leakage:** The clue must not contain the answer word or its morphological root. This is a critical constraint for agglutinative languages like Turkish, where stem leakage renders the puzzle trivial.

B.1.2 Annotator Demographics and Recruitment

The evaluation panels were constructed to ensure native-level linguistic competence and familiarity with the crossword domain.

- **English & Italian Panels:** Evaluation was conducted on a held-out sample of 1,800 generated clues per language. Evaluators were recruited based on native proficiency and confirmed prior experience with solving word puzzles.
- **Turkish Panel:** Given the morphological complexity of the language, the panel consisted of native Turkish speakers specifically trained to detect "root leakage"—instances where the answer stem appears in the clue.
- **Arabic Panel:** The panel comprised native Arabic speakers capable of assessing cultural relevance and grammatical correctness in Right-to-Left (RTL) scripts.

B.1.3 Task Design and Instructions

Annotators were presented with randomized batches of data to ensure a double-blind evaluation. The source of the clue (Teacher model, Student model, or Base model) was hidden to prevent bias.

Each annotation instance consisted of a triplet:

`Context Paragraph` || `Target Keyword (Answer)` || `Generated Clue`

Annotators were instructed to read the context paragraph first, then the target answer, and finally assess the generated clue against the rubric. For Turkish and Arabic tasks, annotators were explicitly warned to flag any morphological variations of the keyword appearing within the clue text.

B.1.4 Rating Taxonomy

A standardized 5-point Likert-style scale was adapted for all four languages to measure the "Acceptability Rate."

RATING A (Excellent / Pedagogically Valid)
The clue is linguistically flawless, factually accurate based on the context, and creates a fair but challenging puzzle experience. It correctly categorizes the term and leads the solver to the answer without ambiguity.

RATING B (Acceptable with Minor Issues)
The clue is solvable and factually correct but may contain minor stylistic imperfections. For example, the correlation with the assigned category might be loose, or the phrasing might be slightly awkward, though not grammatically incorrect.

RATING C (Generic / Vague)
The clue is semantically related to the answer but fails the "Contextual Grounding" constraint. It relies on generic knowledge rather than the specific details provided in the source text. While solvable, it holds low educational value for reviewing the specific lesson.

RATING D (Incorrect / Irrelevant)
The clue is factually wrong, linguistically malformed, or unrelated to the answer.

This rating is also applied to hallucinations where the model invents facts not present in the source text.

RATING E (Unacceptable / Leakage)
The clue contains the answer, a direct translation of the answer, or a morphological variant of the answer (e.g., using "write" when the answer is "writer"). These items are considered objective failures as they break the mechanics of the puzzle.

B.1.5 Metrics and Agreement Analysis

For the crossword domain, the primary success metric was the **Acceptability Rate**, defined as the percentage of generations receiving A or B ratings.

- **Success Criterion:** A model was considered "Production Ready" if the combined A+B rate exceeded 80%.
- **Leakage Analysis:** The frequency of Rating E was tracked separately to measure the model's adherence to negative constraints.
- **Agreement Strategy:** Due to the objective nature of "Leakage" (Rating E) and "Hallucination" (Rating D), strict inter-annotator agreement metrics (like Cohen's Kappa) were deprioritized in favor of large-scale distribution analysis between base and fine-tuned models.

B.2 Protocol B: Question Generation (MCQ & SAQ)

This protocol was applied to the *Turkish-Quiz-Instruct* (Chapter 8) and *PersianMCQ-Instruct* (Chapter 9) datasets.

B.2.1 Evaluation Rationale and Objectives

Educational assessment items require a higher standard of verification than gamified content. A wrong answer in a quiz can actively harm the learning process by reinforcing incorrect information. Therefore, this protocol prioritized **factuality** and **distractor plausibility**.

- **Distractor Validation:** For Multiple-Choice Questions (MCQs), the incorrect options (distractors) must be plausible to a learner but objectively incorrect to an expert.
- **Morphological Correctness:** In Persian and Turkish, questions must adhere to specific grammatical case markers which non-instructed models often miss.

B.2.2 Annotator Demographics and Recruitment

- **Persian Panel:** The evaluation was conducted by two native Persian speakers, both postgraduate university students. Their academic background ensured they could evaluate the pedagogical nuance of questions generated from Wikipedia articles.

- **Turkish Panel:** Items were scrutinized by native Turkish speakers with expertise in the linguistic subtleties of the language and familiarity with national educational assessment standards.

B.2.3 Task Design and Instructions

Annotators were shown the `Source Text` and the generated `Question Object` (comprising the Stem, Key, and Distractors). They were instructed to act as an instructor vetting questions for a formal exam.

B.2.4 Rating Taxonomy

To ensure consistency with the Crossword protocol, the tabular rubric used during the annotation phase is presented here as a standardized descriptive list.

RATING A (Excellent / Publication Ready)
The stem is clear, concise, and grammatical. The key is unambiguously correct based *only* on the text. All distractors are plausible (related to the topic) but clearly incorrect.

RATING B (Good / Classroom Ready)
The item is mostly factual and relevant. It may contain minor grammatical typos or slight ambiguity in one distractor, but the correct answer remains clear and distinct.

RATING C (Acceptable / Requires Editing)
The item is loosely related to the text or addresses peripheral/trivial details. Questions may be too easy (e.g., distractors are obviously wrong or unrelated).

RATING D (Poor / Significant Flaws)
The item contains factual inaccuracies, significant grammatical errors, or the question is minimally relevant to the source text.

RATING E (Unusable / Critical Failure)
The question is incomprehensible, misleading, or contains "Answer Leakage" (the answer is revealed in the question stem). Alternatively, there is no correct answer among the options provided.

B.2.5 Metrics and Agreement Analysis

Unlike the crossword task, the Persian MCQ evaluation included a rigorous statistical analysis of inter-annotator agreement to validate the subjectivity of the rubric.

- **Sample Size:** 600 randomly selected questions were evaluated.
- **Overlap:** 200 of these questions were evaluated by both expert annotators independently to calculate agreement.

- **Quantitative Result:** The annotators achieved a raw agreement rate of **0.96**, with a **Cohen's Kappa of 0.86**. This indicates "almost perfect" agreement, validating that the distinctions between "Excellent" and "Unusable" were interpreted consistently by experts.

B.3 Protocol C: Syntax Feedback Generation

This protocol was applied to the *Essay-Syntax-Instruct* dataset (Chapter 10).

B.3.1 Evaluation Rationale and Objectives

Evaluating syntax feedback is inherently more complex than evaluating questions or clues because there is no single "Gold Standard." A student essay may contain multiple errors, and there are multiple valid ways to correct them.

- **Privacy Constraints:** The source texts were authentic student essays with anonymized placeholders (e.g., `@PERSON1`). Evaluators assessed if the model correctly handled these placeholders without hallucinating errors or leaking formats.
- **Utility over Pedantry:** The goal was not merely to identify errors, but to provide *actionable* feedback.

B.3.2 Annotator Demographics and Recruitment

The evaluation was entrusted to **two human raters**, both of whom were **master's students in linguistics**. This high level of qualification was necessary because the task involved distinguishing between subtle stylistic choices and actual syntactic errors—a distinction general crowd-workers often fail to make.

B.3.3 Task Design and Instructions

Raters were presented with the `Student Essay` and the `Generated JSON Feedback`. They were instructed to verify the feedback against seven specific syntax categories defined in the system prompt: Misspelled Words, Conjunctions, Modifiers, Prepositions, Modal Verbs, Punctuation, and Articles.

B.3.4 Rating Taxonomy

Because "correctness" is fluid in writing feedback, the scale focused on the *quality of pedagogical intervention.*

RATING A (Exceptional)
: The model identifies every error in the category, provides accurate corrections, and offers an explanation that demonstrates a profound understanding of the context. It handles privacy placeholders correctly.

RATING B (Robust)
The model catches the majority of errors and provides correct fixes. It may miss a very minor issue or the explanation might be slightly generic, but the advice is reliable and safe for a student to follow.

RATING C (Adequate)
The model identifies errors but the corrections are marginally acceptable (e.g., fixing grammar but losing the student's original voice). It may suggest unnecessary or stylistic changes disguised as grammatical rules.

RATING D (Inconsistent)
The feedback is of variable quality. It catches some errors but misses obvious ones, or provides confusing advice that requires teacher intervention to clarify.

RATING E (Deficient)
The feedback is harmful. It fails to identify errors, hallucinates errors where there are none (over-correction), or misinterprets privacy placeholders as typos.

B.3.5 Metrics and Agreement Analysis

The evaluation was conducted on a subset of 300 essay-feedback pairs. The analysis focused on the **Quality Shift Distribution**.

- **Baseline Performance:** Raters found that base models (Llama-2, Mistral) frequently fell into Rating D/E, often over-correcting valid sentences or failing to output valid JSON.
- **Fine-Tuned Performance:** The fine-tuned models shifted the majority of responses to Rating B (Robust) and A (Exceptional).
- **Qualitative Consensus:** While a formal Kappa was not calculated for this task, the qualitative reports from the linguists highlighted a specific reduction in "hallucinated rule violations" after fine-tuning, validating the semantic alignment of the student models to the teacher's supervision.

B.4 Summary of Evaluation Limitations

While these protocols were rigorous, several limitations must be acknowledged (as discussed in Chapter 11):

1. **Scale vs. Depth:** Human evaluation is resource-intensive. While we evaluated thousands of items, this represents a fraction of the millions of generated tokens.
2. **Subjectivity in Style:** In Crossword and Syntax tasks, the preference for a specific "clue style" or "writing style" introduces unavoidable subjectivity.
3. **Teacher-Student Bias:** Since the rubric aligns with the "Teacher" model's style (GPT-4), there is a risk that raters prefer outputs that mimic GPT-4, potentially penalizing valid but distinct outputs from smaller models.

Appendix C

Translations of Non-English Content

This appendix provides English translations for figures, tables, and prompts containing non-English text (Italian, Turkish, Arabic, and Persian) found throughout the thesis.

Chapter 5: Italian Educational Crossword Generation

Figure 5.11: An illustrative Italian educational crossword

Translation of Puzzle Content:

- **Across (Orizzontali):**
  - **1.** Simultaneous Translation. [1]
  - **2.** A critical evaluation. (*Answer: Recensione*) [1]
  - **4.** A hilarious story is one. (*Answer: Comica*) [1]
  - **7.** Set of arts, techniques, and industrial activities productive of a film. (*Answer: Cinema*) [1]
  - **8.** Original language from which the word cinema derives. (*Answer: Greco Antico*) [1]
  - **9.** One of the terms extracted from the ancient Greek language, used to describe cinema. (*Answer: Movimento*) [1]
- **Down (Verticali):**
  - **1.** A film... as a document. (*Answer: Documentario*) [1]
  - **3.** Final commercial product of a set of work comprising arts, techniques, and industrial activities. (*Answer: Film*) [1]
  - **5.** A highly coveted award. (*Answer: Oscar*) [1]
  - **6.** They enter the box office coffers. (*Answer: Incassi*) [1]

Table 5.1: Examples of generated clues from the Italian-Clue-Instruct dataset

Translation of Selected Clues:

- **Robocall:** Can be blocked by phone companies to prevent scams. [2], [3]

- **Ministry of Magic (Ministero della Magia):** Corrupt and incompetent government in J.K. Rowling's Wizarding World. [2], [3]
- **Lovesick:** Renewed for a third season, released exclusively on Netflix. [2], [3]
- **South American Tapir (Tapirro Sudamericano):** One of the four recognized species in the tapir family. [2], [3]

Chapter 6: Turkish Educational Crossword Generation

Figure 6.7: An illustrative Turkish educational crossword

Translation of Puzzle Content: [4], [5]

- **ACROSS:**
  - **1.** The Czech state of Europe, held the electorate of the Holy Roman Empire. (7)
  - **2.** The capital of Brandenburg, former residence of Prussia, famous for its lakes and palaces. (7)
  - **6.** One of China's oldest and largest historical capitals; although 99.1% of its 13 million population are Han Chinese, there are approximately 50,000 Hui Muslims. (4)
  - **8.** Fortress; A historical fortress on the UNESCO list, joined to Serbian lands at the end of the 15th century. (9)
  - **10.** The scientific origin story of mountains shapes the fate of continents. (8)
  - **11.** Vehicle that travels on water, with sails or engine. (4)
  - **13.** One of the four basic operations of arithmetic, a close relative of subtraction. (5)
  - **15.** A people belonging to the Plains Indians culture group, speaking Dakota languages. (9)
  - **16.** Indispensable to ceramic art, it comes out of nature like mud. (3)
  - **17.** The name of the day which was "Saturn's Day" in Old English, the "seventh" day in Old Turkish. (9)
- **DOWN:**
  - **1.** Legendary ruler of the Kingdom of Sheba, mysterious queen of Abyssinia. (6)
  - **3.** Activity that tests your intelligence and skill, blending entertainment and competition. (4)
  - **4.** The building block of life, 0.2% is in the earth's crust. (6)
  - **5.** Nomadic warriors who came to Anatolia in the 8th century BC and brought an end to Phrygia. (9)

– **7.** Those who set foot in America after Virginia in 1620, pioneers of Thanksgiving Day. (10)

– **9.** Junction; a circle in Cyprus that rotates traffic flow. (5)

– **12.** In psychology, a pleasant state where you would want to go back in the future. (4)

– **14.** On the shores of the Caspian, the heart of Azerbaijan and the capital of oil. (4) </itemize

Chapter 7: Arabic Educational Crossword Generation

Figure 7.9: Worked example (text-conditioned)

Translation of Process Steps: [6], [7], [8]

– **(a) Input Paragraph:** The atom is the smallest part of a chemical element that can be reached... It consists of a cloud of negative charges (electrons) orbiting a nucleus... The nucleus consists of positively charged protons and neutral neutrons...

– **(b) Extracted Keywords:** Atom, Chemical Element, Chemical Properties, Electrons, Nucleus, Protons, Neutrons, Isotopes.

– **(c) Generated Clues:**

  * *Atom:* The smallest part of a chemical element...
  * *Chemical Element:* Consists of atoms and maintains chemical properties...
  * *Chemical Properties:* Determines the chemical reaction...
  * *Nucleus:* Located in the center of the atom, contains protons and neutrons...
  * *Protons:* Located in the nucleus and carries a positive charge...

Figure 7.10: Illustrative Arabic educational crossword (Physics topic)

Translation of Clues: [6], [9]

– **Across:**

  * **1.** Rotates around the nucleus in the atom. (*Answer: Electrons*)
  * **4.** In the sky at night.
  * **5.** Smallest part of a chemical element that can be reached.
  * **6.** Found in the nucleus and has no charge. (*Answer: Neutrons*)
  * **7.** Consists of atoms and retains chemical properties.

* **8.** Can undergo chemical reactions according to its chemical properties. </itemize
* **Down:**
  - **1.** Found in the nucleus and carries a positive charge. (*Answer: Protons*)
  - **2.** One of the metals.
  - **3.** Different forms of the same element. (*Answer: Isotopes*)
  - **5.** Consists of protons and neutrons in the atom. (*Answer: Nucleus*)

Chapter 8: Turkish Educational Quiz Generation

Figure 8.6: Illustrative content–question pair for photosynthesis

Translation of Content: [10], [11]

* **(a) Source Content:** Photosynthesis occurs in 2 stages. In the first stage, ATP is produced from solar energy, and in the second stage, the produced ATPs are used in nutrient synthesis. The first stage is called light-dependent reactions. The second stage is called light-independent reactions.
* **(b) Multiple-choice question:** What is produced in the first stage of photosynthesis, which occurs with light-dependent reactions?
  - *Choices:* A. ATP, B. Glucose, C. Carbon dioxide.
  - *Right Answer:* ATP.
* **(c) Short-answer question:** In which stage does ATP production occur in photosynthesis?
  - *Answer:* Light-dependent reactions.

Chapter 9: Persian Multiple-Choice Question Generation

Figure 9.2: The three-stage prompt cascade

Translation of Persian Markers in Prompt Structure: [12], [13]

* **Pasokh-e Sahih:** Correct Answer.
* **Pasokh-e Ghalat:** Incorrect Answer / False Answer.
* **Alef:** A (First choice).
* **Be:** B (Second choice).
* **Jim:** C (Third choice).

* **Dal:** D (Fourth choice).

Here is the LaTeX code for a comprehensive **Appendix: Experimental Hyperparameters**.

This section aggregates the training details from your various papers (English, Italian, Turkish, Arabic, Persian, and Syntax Feedback). I have standardized the hardware description as requested and filled in parameters where explicitly available in your sources. For the older GPT-3 experiments (English/Italian legacy), I utilized the standard API parameters mentioned in your methodology sections.

Appendix D

Experimental Hyperparameters and Training Details

This appendix provides a comprehensive overview of the training configurations, hyperparameters, and computational resources utilized across the experiments presented in this thesis. To ensure reproducibility and transparency, we detail the settings for both the proprietary API-based models (e.g., GPT-3.5) and the open-source models fine-tuned via Parameter-Efficient Fine-Tuning (PEFT).

D.1 Computational Infrastructure

Unless otherwise noted, all experimental fine-tuning and inference tasks for open-source models were conducted on a uniform hardware setup to ensure consistency.

* **Hardware:** A dedicated server node equipped with **dual NVIDIA RTX A6000 GPUs**.
* **GPU Memory:** 48 GB GDDR6 per GPU (96 GB total VRAM).
* **Frameworks:** Experiments were implemented using PyTorch, HuggingFace Transformers, and the PEFT library. Training acceleration was managed using DeepSpeed (ZeRO-2/3) and FlashAttention-2 where applicable.

D.2 Part I: Crossword Generation Models

D.2.1 English and Italian (Legacy Answer-Only)

For the initial experiments involving answer-conditioned generation (generating clues from keywords without context), we utilized the OpenAI Fine-tuning API.

D.2.2 Arabic Crossword Generation (Text-Conditioned)

For the Arabic-Clue-Instruct experiments (Chapter 7), we fine-tuned Llama-3-8B-Instruct using higher rank adapters to accommodate the complexity of the Arabic script and morphology.

Hyperparameter	Value
Models	GPT-3 (Davinci, Curie), GPT-2 XL
Optimizer	Adam (Default)
Batch Size	16
Learning Rate Multiplier	0.01 to 0.1 (Adaptive)
Epochs	3
Prompt Loss Weight	0.1

Table D.1: Hyperparameters for Legacy Answer-Only Generators (English & Italian).

Hyperparameter	Value
Base Model	Llama-3-8B-Instruct
PEFT Method	LoRA
LoRA Rank (r)	32
LoRA Alpha (α)	64
LoRA Dropout	0.05
Batch Size	128 (Global effective batch size)
Learning Rate	3×10^{-4}
LR Scheduler	Cosine
Epochs	3
Inference	
Temperature	0.1
Top-p	0.95
Top-k	50

Table D.2: Training details for Arabic Text-to-Clue generation.

D.2.3 Turkish Crossword Generation

For the Turkish T4TAC experiments (Chapter 6), we utilized Llama-2 models. The agglutinative nature of Turkish required specific attention to leakage prevention during validation, but standard LoRA parameters proved effective for generation.

Hyperparameter	**Value**
Base Models	Llama-2-7b-chat, Llama-2-13b-chat
PEFT Method	LoRA
LoRA Rank (r)	16
LoRA Alpha (α)	32
Batch Size	64
Learning Rate	1×10^{-4}
Epochs	3
Target Modules	q_proj, v_proj

Table D.3: Training details for Turkish Crossword Generation.

D.3 Part II: Question Generation (MCQ & SAQ)

D.3.1 Turkish Educational Quiz Generation

For generating MCQs and SAQs in Turkish (Chapter 8), we maintained consistency with the crossword parameters to evaluate cross-task transferability of configurations.

Hyperparameter	**Value**
Base Models	Llama-2-7b-chat, Llama-2-13b-chat
Method	LoRA
Rank (r) / Alpha (α)	16 / 32
Learning Rate	1×10^{-4}
Batch Size	64
Epochs	3
Comparison Model	*GPT-3.5-Turbo (Fine-tuned via API)*
Batch Size	16
Learning Rate	0.001
Epochs	3

Table D.4: Hyperparameters for Turkish Quiz Generation (MCQ & SAQ).

D.3.2 Persian Multiple-Choice Generation

For Persian (Chapter 9), we experimented with newer architectures (Gemma-2, Mistral-v0.3, Llama-3.1). We increased the sequence length to accommodate the verbosity of Persian educational texts and the prompt structure.

Hyperparameter	Value
Base Models	Llama-3.1-8B, Gemma-2-9B, Mistral-7B-v0.3
LoRA Rank (r)	16
LoRA Alpha (α)	32
Learning Rate	1×10^{-4}
Weight Decay	1×10^{-4}
Scheduler	Cosine
Batch Size	4 (per device)
Gradient Accumulation	4 steps
Max Sequence Length	3500 tokens
Epochs	3
Optimization	DeepSpeed ZeRO-2, FlashAttention-2

Table D.5: Training details for Persian MCQ Generation.

D.4 Part III: Automated Syntax Feedback

For the English syntax feedback task (Chapter 10), we required higher precision to handle the JSON schema and specific grammatical rule adherence. Consequently, we utilized a higher LoRA rank and a larger learning rate compared to the QG tasks.

Hyperparameter	Value
Base Models	Llama-2-7B, Llama-2-13B, Mistral-7B-v0.2
LoRA Rank (r)	32
LoRA Alpha (α)	64
Batch Size	16
Learning Rate	3×10^{-4}
Warmup Ratio	0.1
LR Scheduler	Cosine
Epochs	3
Inference	
Temperature	0.3 (Reduced for deterministic JSON structure)
Top-p	0.95
Top-k	50

Table D.6: Training details for Syntax Feedback Generation.

‘‘‘

Bibliography

[1] "The hewlett foundation: Automated essay scoring (asap aes)," https://www.kaggle.com/competitions/asap-aes, 2012, kaggle competition.

[2] "Asap 2.0: Automated student assessment prize," https://www.kaggle.com/datasets/lburleigh/asap-2-0, 2024, kaggle dataset.

[3] K. Zeinalipour, T. Iaquinta, G. Angelini, L. Rigutini, M. Maggini, and M. Gori, "Building bridges of knowledge: Innovating education with automated crossword generation," in *2023 International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2023, pp. 1228–1236.

[4] K. Zeinalipour, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, M. Gori *et al.*, "Italian crossword generator: Enhancing education through interactive word puzzles," *arXiv preprint arXiv:2311.15723*, 2023.

[5] A. Zugarini, K. Zeinalipour, S. S. Kadali, M. Maggini, M. Gori, and L. Rigutini, "Clue-instruct: Text-based clue generation for educational crossword puzzles," *arXiv preprint arXiv:2404.06186*, 2024.

[6] K. Zeinalipour, A. Fusco, A. Zanollo, M. Maggini, and M. Gori, "Harnessing llms for educational content-driven italian crossword generation," *arXiv preprint arXiv:2411.16936*, 2024.

[7] K. Zeinalipour, M. Z. Saad, M. Maggini, and M. Gori, "Arabicros: Ai-powered arabic crossword puzzle generation for educational applications," *arXiv preprint arXiv:2312.01339*, 2023.

[8] K. Zeinalipour, M. Z. Saad, M. Maggini, and M. Gori, "From arabic text to puzzles: Llm-driven development of arabic educational crosswords," *arXiv preprint arXiv:2501.11035*, 2025.

[9] K. Zeinalipour, Y. G. Keptiğ, M. Maggini, L. Rigutini, and M. Gori, "A turkish educational crossword puzzle generator," in *International Conference on Artificial Intelligence in Education*. Springer, 2024, pp. 226–233.

[10] K. Zeinalipour, T. Iaquinta, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, and M. Gori, "Italian crossword generator: an in-depth linguistic analysis in educational word puzzles," *IJCoL. Italian Journal of Computational Linguistics*, vol. 11, no. 11-1, 2025.

[11] K. Zeinalipour, N. Jamshidi, F. Akbari, M. Maggini, M. Bianchini, and M. Gori, "Persianmcq-instruct: A comprehensive resource for generating multiple-choice

questions in persian," *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pp. 344–372, 2025.

[12] K. Zeinalipour, Y. G. Keptiğ, M. Maggini, and M. Gori, "Automating turkish educational quiz generation using large language models," *arXiv preprint arXiv:2406.03397*, 2024.

[13] K. Zeinalipour, M. Mehak, F. Parsamotamed, M. Maggini, and M. Gori, "Advancing student writing through automated syntax feedback," *arXiv preprint arXiv:2501.07740*, 2025.

[14] L. Rigutini, M. Diligenti, M. Maggini, and M. Gori, "A fully automatic crossword generator," in *2008 Seventh International Conference On Machine Learning And Applications*, 2008, pp. 362–367.

[15] L. Rigutini, M. Diligenti, M. Maggini, and M. Gori, "Automatic generation of crossword puzzles," *International Journal On Artificial Intelligence Tools*, vol. 21, p. 1250014, 2012.

[16] B. Ranaivo-Malançon, T. Lim, J. Minoi, and A. Jupit, "Automatic generation of fill-in clues and answers from raw texts for crosswords," in *2013 8th International Conference On Information Technology In Asia (CITA)*, 2013, pp. 1–5.

[17] J. Esteche, R. Romero, L. Chiruzzo, and A. Rosá, "Automatic definition extraction and crossword generation from spanish news text," *CLEI Electronic Journal*, vol. 20, 2017.

[18] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Austin, Texas: Association for Computational Linguistics, 2016, pp. 2383–2392.

[19] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale reading comprehension dataset from examinations," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Copenhagen, Denmark: Association for Computational Linguistics, 2017.

[20] H. S. Nwana, "Intelligent tutoring systems: An overview," *Artificial Intelligence Review*, vol. 4, no. 4, pp. 251–277, 1990.

[21] A. C. Graesser, K. VanLehn, C. P. Rose, P. W. Jordan, and D. Harter, "Intelligent tutoring systems with conversational dialogue," *AI Magazine*, vol. 22, no. 4, pp. 39–51, 2001.

[22] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational Psychologist*, vol. 46, no. 4, pp. 197–221, 2011.

[23] M. Heilman and N. A. Smith, "Good question! statistical ranking for question generation," in *Human Language Technologies: The 2010 Annual Conference of the NAACL*, 2010, pp. 609–617.

[24] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 1342–1352.

[25] Y.-H. Chan and Y.-C. Fan, "A recurrent bert-based model for question generation," in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (MRQA)*, 2019, pp. 154–162.

[26] L. Ouyang, J. Wu, X. Jiang, D. Almeida *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 27 730–27 744, 2022.

[27] T. OpenAI (Brown, D. Amodei, and J. e. a. Achiam, "Gpt-4 technical report," OpenAI, Tech. Rep. arXiv:2303.08774, 2023, preprint.

[28] S. Wang, T. Xu, H. Li, C. Zhang, J. Liang, J. Tang, P. S. Yu, and Q. Wen, "Large language models for education: A survey and outlook," *arXiv preprint arXiv:2403.18105*, 2024.

[29] G. Rehm, "Are multilingual language models an off-ramp for under-resourced languages? will we arrive at digital language equality in europe in 2030?" *arXiv preprint arXiv:2502.12886*, 2025.

[30] Z. Liu, S. X. Yin, D. H.-L. Goh, and N. F. Chen, "Cogent: A curriculum-oriented framework for generating grade-appropriate educational content," *arXiv preprint arXiv:2506.09367*, 2025.

[31] P. Henry, "How duolingo uses ai to create lessons faster," Duolingo Blog (June 22, 2023), 2023. [Online]. Available: https://blog.duolingo.com/large-language-model-duolingo-lessons/

[32] E. Zelikman, W. Ma, J. Tran, D. Yang, J. Yeatman, and N. Haber, "Generating and evaluating tests for k-12 students with language model simulations: A case study on sentence reading efficiency," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023, pp. 2190–2205.

[33] P. Sridhar, A. Doyle, A. Agarwal, C. Bogart, J. Savelka, and M. Sakr, "Harnessing llms in curricular design: Using gpt-4 to support authoring of learning objectives," *arXiv preprint arXiv:2306.17459*, 2023.

[34] E. Latif, L. Fang, P. Ma, and X. Zhai, "Knowledge distillation of llm for education," *arXiv preprint arXiv:2312.15842*, 2023.

[35] W. Orawiwatnakul, "Crossword puzzles as a learning tool for vocabulary development," *Electronic Journal Of Research In Education Psychology*, vol. 11, pp. 413–428, 2013.

[36] Y. Bella and E. Rahayu, "The improving of the student's vocabulary achievement through crossword game in the new normal era," *Edunesia: Jurnal Ilmiah Pendidikan*, vol. 4, pp. 830–842, 2023.

[37] D. Dzulfikri, "Application-based crossword puzzles: Players' perception and vocabulary retention," *Studies In English Language And Education*, vol. 3, pp. 122–133, 2016.

[38] R. Nickerson, "Crossword puzzles and lexical memory," in *Attention And Performance VI*, 1977, pp. 699–718.

[39] E. Yuriev, B. Capuano, and J. Short, "Crossword puzzles for chemistry education: learning goals beyond vocabulary," *Chemistry Education Research And Practice*, vol. 17, pp. 532–554, 2016.

[40] C. Sandiuc and A. Balagiu, "The use of crossword puzzles as a strategy to teach maritime english vocabulary," *Scientific Bulletin "Mircea Cel Batran" Naval Academy*, vol. 23, pp. 236A–242, 2020.

[41] S. Kaynak, S. Ergün, and A. Karadaş, "The effect of crossword puzzle activity used in distance education on nursing students' problem-solving and clinical

decision-making skills: A comparative study," *Nurse Education In Practice*, vol. 69, p. 103618, 2023.

[42] S. Mueller and E. Veinott, "Testing the effectiveness of crossword games on immediate and delayed memory for scientific vocabulary and concepts," in *CogSci*, 2018.

[43] V. Zirawaga, A. Olusanya, and T. Maduku, "Gaming in education: Using games as a support tool to teach history," *Journal Of Education And Practice*, vol. 8, pp. 55–64, 2017.

[44] P. Zamani, S. Haghighi, and M. Ravanbakhsh, "The use of crossword puzzles as an educational tool," *Journal Of Advances In Medical Education & Professionalism*, vol. 9, p. 102, 2021.

[45] S. Dol, "Gpbl: An effective way to improve critical thinking and problem solving skills in engineering education," *J Engin Educ Trans*, vol. 30, pp. 103–13, 2017.

[46] B. Arora and N. Kumar, "Automatic keyword extraction and crossword generation tool for indian languages: Seekh," in *2019 IEEE Tenth International Conference On Technology For Education (T4E)*, 2019, pp. 272–273.

[47] X. Du, J. Shao, and C. Cardie, "Identifying where to focus in reading comprehension for neural question generation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 315–325.

[48] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Jiang, and M. Zhou, "Neural question generation: a survey of the state of the art," *arXiv preprint arXiv:1807.06910*, 2018.

[49] V. Harrison, A. T. Walker, and A. L. Ralescu, "Neural Question Generation from Text: A Preliminary Study," in *Proceedings of the International Conference on Natural Language Generation*, 2018, pp. 169–177.

[50] Y. Zhou, L. Sun, Z. Fan, Y. Xu, and Z. Liu, "Multi-task learning for question generation," in *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, 2019, pp. 483–490.

[51] X. Yuan, T. Cui, Y. Liu, S. Chen, S. Liu, Y. Zhang, and M. Zhou, "Machine comprehension by text-to-text neural question generation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1116–1126.

[52] S. Naeiji, H. Maserrat, and B. Naderi, "Question Generation for Multimodal Documents," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 6154–6164.

[53] T. Chan, Y. Zhang, H. Li, and H. Zhao, "A recurrent generative model for pre-trained language models via layer-wise attention," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2626–2635.

[54] B. Wang, Z. Li, X. Yu, J. Cui, and G. Chen, "Towards human-like educational question generation with large language models," in *Proceedings of the 2022 International Conference on Advanced Learning Technologies (ICALT)*, 2022, pp. 16–20.

[55] J. Welbl, N. F. Liu, and M. Gardner, "Crowdsourcing Multiple-Choice Science Questions," in *Proceedings of the 3rd Workshop on Argument Mining*, 2017, pp. 94–105.

[56] Y. Xu, D. Li, D. Bracewell, J. Zhu, and Z. Zeng, "Fantastic Questions and Where to Find Them: FairytaleQA – An Authentic Dataset for Reading Comprehension," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 6719–6731.

[57] D. Tang, N. Jiang, F. Wei, M. Zhou, X. Liu, and H.-W. Hon, "Question Answering and Question Generation as Dual Tasks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 2158–2168.

[58] Q. Jia, Y. Zhang, Z. Yang, and S. Zhang, "EQG-RACE: Examination-Level Question Generation on RACE," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 1128–1133.

[59] J.-D. Steuer and S. Eger, "Investigating the Feasibility of Generating Question-Answer Pairs from Scientific Texts," in *Proceedings of the 13th Language Resources and Evaluation Conference*, 2022, pp. 4685–4695.

[60] W. Zhao, X. Li, Y. Zhang, Y. Cao, W. Liu, B. Wang, Z. Wang, L. Qin, and J. Tang, "Educational Question Generation of Children Storybooks via Question Type Distribution Learning and Event-centric Summarization," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 3327–3338.

[61] G. Chen, J. Cui, Z. Li, and X. Yu, "LearningQ: A Large-scale Dataset for Educational Question Generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[62] Y. Gong, J. Pan, and K. Hu, "KHANQ: A Dataset for Generating Questions from Videos of Khan Academy," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 6706–6718.

[63] A. Hadifar, A. Majumder, R. Al-Ghezi, J. May, and A. Mitra, "EduQG: A Multi-format and Multi-level Educational Question Generation Dataset," in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, 2023, pp. 14–29.

[64] L. Dugan, E. Velldal, and S. Oepen, "On the Feasibility of Automated Question Generation for the Norwegian National Tests in Reading," in *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, 2022, pp. 25–35.

[65] X. Sun, J. Yang, Z. Zhang, and Y. Lv, "Answer-aware deep question generation," in *2018 24th International conference on pattern recognition (ICPR)*, 2018, pp. 3160–3165.

[66] T. Ma, L. Dong, F. Wei, M. Zhou, and Y. Huang, "Improving question generation with to the point context," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5683–5692.

[67] A. Mitra, A. Majumder, S. Mahajan, V. Axente, A. K. Mohankumar, P. Jha, S. M. T. I. Tonmoy, and H. Firooz, "AgentInstruct: A Min-Cost Agentic Data Generation Framework," 2024.

[68] F. M. Lord, *Applications of item response theory to practical testing problems*. Routledge, 2012.

[69] L. Qiu, Y. Han, and Y. Wu, "Automatic generation of distractors for multiple-choice questions using learning to rank," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 4539–4550.

[70] M. Uto and T. Tokunaga, "Difficulty-controllable question generation for reading comprehension," *IEEE Access*, vol. 11, pp. 3549–3560, 2023.

[71] A. Hadifar, A. Majumder, R. Al-Ghezi, J. May, and A. Mitra, "Diverse, Relevant, and Coherent: A Human-in-the-Loop Framework for Multiple-Choice Question Generation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 15 147–15 162.

[72] M. Srivastava and N. Goodman, "Question Generation for Adaptive Education," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, 2021, pp. 13 589–13 597.

[73] P. Wang, Z. Lin, M. Zhao, and Y. Liu, "Multi-task learning for personalized question generation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 12 937–12 948.

[74] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: a systematic literature review," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2495–2527, 2022.

[75] E. Miltsakaki and K. Kukich, "Evaluation of text coherence for electronic essay scoring systems," *Natural Language Engineering*, vol. 10, no. 1, pp. 25–55, 2004.

[76] M. A. Sultan, C. Salazar, and T. Sumner, "Fast and easy short answer grading with high accuracy," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1070–1075.

[77] S. Mathias and P. Bhattacharyya, "Thank "goodness"! a way to measure style in student essays," in *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, 2018, pp. 35–41.

[78] Y. Salim, V. Stevanus, E. Barlian, A. C. Sari, and D. Suhartono, "Automated english digital essay grader using machine learning," in *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*. IEEE, 2019, pp. 1–6.

[79] F. Dong and Y. Zhang, "Automatic features for essay scoring–an empirical study," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 1072–1077.

[80] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 1882–1891.

[81] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. Lee, "Investigating neural architectures for short answer scoring," in *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*, 2017, pp. 159–168.

[82] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[83] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[84] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[85] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He, "Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1560–1569.

[86] Y. Wang, C. Wang, R. Li, and H. Lin, "On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation," *arXiv preprint arXiv:2205.03835*, 2022.

[87] P. U. Rodriguez, A. Jafari, and C. M. Ormerod, "Language models and automated essay scoring," *arXiv preprint arXiv:1909.09482*, 2019.

[88] J. Lun, J. Zhu, Y. Tang, and M. Yang, "Multiple data augmentation strategies for improving performance on automatic short answer scoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 09, 2020, pp. 13 389–13 396.

[89] R. Ridley, L. He, X. Dai, S. Huang, and J. Chen, "Prompt agnostic essay scorer: a domain generalization approach to cross-prompt automated essay scoring," *arXiv preprint arXiv:2008.01441*, 2020.

[90] A. Mizumoto and M. Eguchi, "Exploring the potential of using an ai language model for automated essay scoring," *Research Methods in Applied Linguistics*, vol. 2, no. 2, p. 100050, 2023.

[91] K. P. Yancey, G. Laflair, A. Verardi, and J. Burstein, "Rating short l2 essays on the cefr scale with gpt-4," in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, 2023, pp. 576–584.

[92] B. Naismith, P. Mulcaire, and J. Burstein, "Automated evaluation of written discourse coherence using gpt-4," in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, 2023, pp. 394–403.

[93] L. Yan, L. Sha, L. Zhao, Y. Li, R. Martinez-Maldonado, G. Chen, X. Li, Y. Jin, and D. Gašević, "Practical and ethical challenges of large language models in education: A systematic scoping review," *British Journal of Educational Technology*, vol. 55, no. 1, pp. 90–112, 2024.

[94] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, p. 102274, 2023.

[95] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen *et al.*, "Check your facts and try again: Improving large language models with external knowledge and automated feedback," *arXiv preprint arXiv:2302.12813*, 2023.

[96] J. Han, H. Yoo, J. Myung, M. Kim, H. Lim, Y. Kim, T. Y. Lee, H. Hong, J. Kim, S.-Y. Ahn *et al.*, "Fabric: Automated scoring and feedback generation for essays," *arXiv preprint arXiv:2310.05191*, 2023.

[97] C. Xiao, W. Ma, S. X. Xu, K. Zhang, Y. Wang, and Q. Fu, "From automation to augmentation: Large language models elevating essay scoring landscape," *arXiv preprint arXiv:2401.06431*, 2024.

[98] C. Bucilă, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2006, pp. 535–541.

[99] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[100] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1317–1327.

[101] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[102] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5776–5788, 2020.

[103] Y. Gu, L. Li, S. Han, K. Liu, and F. Wei, "Minillm: Knowledge distillation of large language models," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[104] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language models with self-generated instructions," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

[105] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, and D. Jiang, "Wizardlm: Empowering large language models to follow complex instructions," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[106] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 610–623.

[107] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *arXiv preprint arXiv:2311.05232*, 2023.

[108] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[109] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach, "Language (technology) is power: A critical survey of "bias" in nlp," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5454–5476.

[110] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh *et al.*, "Ethical and social risks of harm from language models," *arXiv preprint arXiv:2112.04359*, 2021.

[111] W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou, "Gpt detectors are biased against non-native english writers," *Patterns*, vol. 4, no. 7, 2023.

[112] P. Mishra and M. J. Koehler, "Technological pedagogical content knowledge: A framework for teacher knowledge," *Teachers college record*, vol. 108, no. 6, pp. 1017–1054, 2006.

[113] I. Celik, "Intelligent-tpack: A theoretical framework for integrating artificial intelligence in education," *Computers and Education: Artificial Intelligence*, vol. 5, p. 100156, 2023.

[114] M. J. Gierl and T. M. Haladyna, "Automatic item generation: An overview of the field," *Routledge*, 2012.

[115] J. Qadir *et al.*, "Item response theory analysis of chatgpt-generated multiple-choice items," in *Proceedings of NeurIPS Workshop on Generative AI for Education*, 2023.

[116] W. Malec, "Investigating the quality of ai-generated distractors for a multiple-choice vocabulary test," *Language Testing in Asia*, 2023.

[117] E. L. Bjork and R. A. Bjork, "Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning," in *Psychology and the real world: Essays illustrating fundamental contributions to society*, 2011, pp. 56–64.

[118] H. Bastani, O. Bastani *et al.*, "Generative ai tutors and student learning outcomes," *arXiv preprint arXiv:2405.00202*, 2024.

[119] Z. Toth *et al.*, "The impact of generative ai on learning outcomes: An experimental study at corvinus university," *arXiv preprint arXiv:2510.16019*, 2025.

[120] Hugging Face, "Peft: Parameter-efficient fine-tuning library," https://github.com/huggingface/peft, 2023.

[121] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[122] OpenAI, "Openai gpt-3.5-turbo model documentation," https://platform.openai.com/docs/models/gpt-3-5-turbo, 2023.

[123] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[124] MosaicML, "Mpt-7b: A new standard for open-source, large foundation models," https://www.databricks.com/blog/mpt-7b, 2023, blog post / model card.

[125] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out (ACL Workshop)*. Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 74–81.

[126] A. Dubey and et al., "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[127] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[128] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[129] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, pp. 363–370.