UNIVERSITÀ DEGLI STUDI DI SIENA

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

UNIVERSITÀ
DI SIENA
1240

Generative Artificial Intelligence in Education

Tommaso Iaquinta

PhD in Information Engineering and Science

Supervisor

Prof. Marco Maggini

Examination Commitee

Prof. Franco Scarselli

Prof. Simone Marinai

Prof. Riccardo Lazzeretti

Thesis reviewers

Prof. Fabrizio Silvestri

Prof. Simone Marinai

Siena, 20/10/2025

Firmato digitalmente da
IAQUINTA TOMMASO
C=IT

Contents

List of Figures

List of Tables

Acknowledgements

I wish you were here ...

Part I

Foundation and Context

Chapter 1

Introduction

The integration of Artificial Intelligence (AI) in education has become a prominent topic of discourse, particularly with the rise of generative AI technologies such as Large Language Models (LLMs). While much attention has been devoted to the potential of these tools to transform traditional teaching and learning paradigms, this chapter provides a general introduction grounded in educational theory and empirical findings. The focus will be on primary and secondary education, with a pragmatic stance: AI presents opportunities and challenges, but not necessarily a revolution.

1.1 Artificial Intelligence and Its Educational Relevance

Artificial Intelligence, as a field, has long aspired to emulate human cognitive processes through machines capable of learning, reasoning, and decision-making. Over the decades, AI has transitioned from laboratory prototypes to widespread applications in everyday life [1]. The educational sector has not remained untouched. AI's incursion into the classroom spans from intelligent tutoring systems to automated feedback, and more recently, generative models capable of producing natural language responses on demand.

Among these, generative AI (GenAI) tools like LLMs have drawn considerable interest due to their ability to simulate conversation, generate coherent essays, and support both students and teachers in a variety of tasks. However, as several scholars suggest, including Monib et al. [2], the use of GenAI in education demands a critical evaluation that balances potential benefits with ethical, cognitive, and pedagogical implications.

1.2 The Myth of the Educational Revolution

Historically, claims that new technologies would revolutionize education—ranging from the motion picture to television and MOOCs—have largely failed to materialize. As discussed in recent pedagogical reflections, including the public talk used as a framing reference here, these proclamations often underestimate the systemic complexity of educational change. The pattern suggests that rather than transforming education, technologies are incorporated into existing practices as tools, with varying degrees of success.

This holds true for AI. Although technologies like LLMs can perform seemingly magical tasks, they do not replace the fundamental processes of learning, which involve effort, scaffolding, and social context. As [3] notes, the school remains a central institu-

tion for mediating technology's role, emphasizing both critical literacy and pedagogical intentionality.

1.3 AI in school: Use Cases and Evidence

The practical applications of AI in primary and secondary education have been diverse. Intelligent Tutoring Systems (ITS) have a long history of deployment, offering adaptive feedback and personalized instruction. Koedinger et al. [1] provided early evidence of ITS effectiveness in urban school settings. More recently, [4] reviewed multiple ITS studies and confirmed modest but positive impacts on student performance, particularly when systems incorporated immediate feedback and personalization features.

An example is ASSISTments, an ITS widely used for middle school science and reading comprehension tasks. The platform presents students with scaffolded practice questions and provides step-by-step hints tailored to the specific error a learner makes. For instance, when working on a reading comprehension passage, the system can highlight the relevant section of the text and prompt the learner to reconsider key vocabulary, rather than simply marking the answer wrong. Teachers, in turn, receive detailed analytics that reveal patterns of misunderstanding across the class. Studies have shown that such systems not only improve students' accuracy in practice but also enhance long-term retention and allow teachers to adapt their instruction more effectively [5].

Generative AI adds a new layer to these systems. LLMs have shown promise in providing support for writing, problem-solving, and even content generation. A recent meta-analysis [6] synthesized 51 (quasi-)experimental studies (Nov 2022–Feb 2025) involving thousands of students across different ages and subjects. The key finding was a large positive effect of ChatGPT on students' learning performance, with an overall Hedges' $g = 0.867$ [6]. Hedges' g is a standardized effect size (similar to Cohen's d) used in meta-analyses; a value of 0.867 indicates that, on average, students who used ChatGPT scored about 0.87 standard deviations higher on learning assessments than those who did not. In practical terms, this is a substantial improvement – roughly equivalent to moving a student from the middle of the pack to about the 80th percentile of performance. In educational outcomes, such a nearly one–standard deviation gain is considered statistically and educationally significant.

Zhang and Tur [7] focus on ChatGPT in K-12 education specifically, noting its potential to support personalized learning and improve engagement. Nonetheless, they also underscore the necessity of human oversight and teacher facilitation to avoid over-reliance and ensure accurate understanding.

1.4 Pedagogical and Ethical Considerations

Introducing AI into schools is not merely a technological matter; it is also pedagogical and ethical. AI can enter classrooms in multiple modalities, including intelligent tutoring systems, automated grading tools, adaptive learning platforms, and AI-assisted writing or language support applications. Each of these reshapes the relationship between teachers,

students, and knowledge, which is why teacher readiness, curriculum alignment, and student autonomy remain essential. Ranieri [3] proposes a framework for AI literacy that integrates cognitive, operational, critical, and ethical-social dimensions—crucial for fostering critical digital citizenship.

At the same time, ethical issues such as bias, transparency, and data protection are non-trivial, as emphasized by [2] and[8]. A well-known case is the 2020 UK A-level grading algorithm, which downgraded many students from disadvantaged schools because it relied heavily on historical school performance data, thereby reproducing social inequalities [9]. This episode illustrates how AI, if not carefully designed and monitored, can unintentionally perpetuate inequities in education.

1.5 Toward a Constructive Integration of AI

Rather than aspiring to overturn existing educational models, the integration of artificial intelligence should be approached as a constructive enhancement. Grounded in theories such as constructive alignment and cognitive load theory, AI can function as a pedagogical scaffold—providing adaptive support, personalized pathways, and timely feedback that align with each learner's needs and zone of proximal development. The objective is not to substitute educators, but to reinforce their role in offering differentiated, meaningful instruction [2, 3].

A thoughtful and deliberate implementation of generative AI requires acknowledging both its potential and its limits. This includes mitigating risks such as over-reliance or ethical misuse, and embedding AI tools within purposeful and evidence-based learning practices. AI should support, not automate, the deeper cognitive engagement necessary for mastery.

Among the many applications of AI in education, one promising avenue is the design of intelligent educational games and exercises. In particular, the automatic generation of crosswords has emerged as an effective strategy to reinforce vocabulary acquisition and subject-specific knowledge in a playful and interactive format. Recent research has demonstrated that AI systems based on large language models can autonomously generate and validate high-quality, pedagogically aligned crossword clues and grid layouts, enabling teachers to create customized puzzles with minimal effort [10, 11, 12].

The remainder of this thesis will investigate selected applications of AI in educational settings, with a specific focus on the development and evaluation of an AI-assisted system for the generation of crosswords tailored to primary and secondary level learners. This original contribution is intended to promote engagement, conceptual reinforcement, and personalized learning through gamified, language-rich experiences.

Effective implementation requires a balanced approach: one that acknowledges the capabilities of generative AI, mitigates its risks, and situates it within meaningful educational practices. The remainder of this thesis will explore various applications of AI in education, culminating in the author's original contribution: the development of a system for automatic crossword generation, designed to support vocabulary acquisition and contextual learning.

Contributions

Throughout this doctoral work, I have pursued research goals that integrate computational linguistics, generative artificial intelligence, and pedagogical design principles to foster innovation in digital education. The research led to theoretical, methodological, and applied results, advancing the field of AI in education and yielding concrete tools for adaptive and gamified learning. The main contributions of this thesis are summarized as follows:

- **Comprehensive architecture for automatic educational crossword generation:** design of a hybrid pipeline with two complementary pathways (from text and from keyword lists), including automatic validation, human revision, and crossword grid construction through a dedicated algorithm and scoring function.
- **Large-scale linguistic resource:** creation and typological analysis of a dataset of **125,600** Italian *clue–answer* pairs, with a syntactic taxonomy of over twenty categories supporting both clue generation and evaluation.
- **Definition generation:**
  - *From text:* zero-shot keyword extraction and few-shot definition generation aligned with educational content; quality verified through an automatic zero-shot validator.
  - *From keywords:* fine-tuning of LLMs (of different sizes) to produce crossword-style clues, combined with classifier-based automatic filtering of outputs.
- **End-to-end empirical evaluation:** detailed assessment of all components — keyword extraction (≈80% suitability), definition generation (≈70–77% acceptable from text; ≈60% from keywords with the largest model), and validation (≈70% of flawed clues detected automatically).
- **Surprisal-based difficulty modeling:** proposal and validation of an information-theoretic metric for clue difficulty (estimated via LLMs) that **correlates with human performance** (e.g., $r \approx -0.57$ with accuracy, stronger for nominal clues), supported by mixed-effects analysis and implications for adaptive educational game design.
- **Interdisciplinary contribution:** integration of NLP/LLMs, gamification theory, and instructional design to create a system that supports alignment with learning objectives and personalization of educational activities.

Publications

The publications listed below report research conducted during the doctoral program. For papers with multiple authors, the candidate contributed substantially to the conceptualization, implementation, and experimental evaluation of the proposed systems. In

particular, the candidate was responsible for the construction and preprocessing of the large-scale dataset of Italian crossword clue–answer pairs, the design and implementation of the automated crossword generation pipeline and system architecture, and the development of the end-to-end generation framework presented in the ICMLA 2023 and IJCOL 2025 publications. The candidate also contributed to the linguistic taxonomy analysis through the implementation of the Python-based processing and categorization tools. The work on surprisal-based difficulty modeling was primarily conducted by the candidate.

Journal Articles

- *Italian Crossword Generator: An In-depth Linguistic Analysis in Educational Word Puzzles* — **IJCOL – Italian Journal of Computational Linguistics**, Vol. 11(1), pp. 47–72, January 2025.
  Authors: Kamyar Zeinalipour, **Tommaso Iaquinta**, Asya Zanollo, Giulia Angelini, Leonardo Rigutini, Marco Maggini, Marco Gori.
  DOI: `10.17454/IJCOL111.03` — IRIS: `hdl:11365/1297476`.

Conference Proceedings

1. *Building Bridges of Knowledge: Innovating Education with Automated Crossword Generation* — **ICMLA 2023**, New York (USA), December 15–17, 2023, pp. 1228–1236.
   Authors: Kamyar Zeinalipour, **Tommaso Iaquinta**, Giulia Angelini, Leonardo Rigutini, Marco Maggini, Marco Gori.
   DOI: `10.1109/ICMLA58977.2023.00185` — IRIS: `hdl:11365/1255436`.

2. *Italian Crossword Generator: Enhancing Education through Interactive Word Puzzles* — **CLiC-it 2023**, Venice (Italy), November 30 – December 2, 2023.
   Authors: Kamyar Zeinalipour, **Tommaso Iaquinta**, Asya Zanollo, Giulia Angelini, Leonardo Rigutini, Marco Maggini, Marco Gori.
   Editor: CEUR-WS — IRIS: `hdl:11365/1253156`.

3. *The WebCrow French Crossword Solver* — **EAI INTETAIN 2023**, Lucca (Italy), November 27, 2023.
   Authors: Giulia Angelini, Massimiliano Ernandes, **Tommaso Iaquinta**, Charles Stehlé, Filipe Simões, Kamyar Zeinalipour, Alessandro Zugarini, Marco Gori.
   IRIS: `hdl:11365/1298821`.

4. *Surprisal and Crossword Clues Difficulty: Evaluating Linguistic Processing between LLMs and Humans* — **CLiC-it 2025** (accepted), Cagliari (Italy), September 24–26, 2025.
   Authors: **Tommaso Iaquinta**, Asya Zanollo, Achille Fusco, Kamyar Zeinalipour, Cristiano Chesi.

Chapter 2

Theoretical Foundations

2.1 Historical Evolution of Educational Technology

Technological innovations have long promised to transform education, yet history shows a pattern of exaggerated expectations followed by gradual integration. As early as 1922, Thomas Edison proclaimed that motion pictures would revolutionize schooling and eventually replace textbooks, claiming they could achieve "100 percent efficiency" compared to the mere 2 percent he ascribed to books [13].

Similar optimism greeted each new medium: radio in the 1930s, instructional television in the 1950s, personal computers in the 1980s, and online learning in the 2000s. In each case, bold predictions of a pedagogical revolution were tempered by the reality of classroom practice and systemic complexity. For example, the introduction of educational television was largely driven by policymakers and technology enthusiasts rather than teachers, and its classroom use remained limited [14].

Early studies found no substantial differences in learning outcomes between televised lessons and traditional teacher-led instruction [14], undercutting the notion that the new medium was inherently superior. Several recurring factors help explain why education has evolved rather than been "revolutionized" by technology. Often the infrastructure and training needed for effective use lag behind; for instance, mid-century surveys found that lack of equipment, scheduling difficulties, and misalignment with curriculum hampered the adoption of radio and TV in schools [14]. Teachers tend to embrace tools that genuinely solve problems or enhance what they already do—for example, interactive whiteboards, which extend traditional chalkboard teaching with digital annotation and multimedia support—while remaining skeptical of those that disrupt established classroom dynamics. Notably, interpersonal interaction and guidance are viewed as central to learning, so technologies that sideline the teacher's role often face resistance [14].

Thus, new devices and software have typically been assimilated as supplements to traditional practices (e.g. language labs, computer labs, smartboards), rather than wholesale replacements for them. This historical perspective informs current discussions about Artificial Intelligence in education. Like earlier innovations, AI is sometimes advertised as a game-changer that could automate or personalize learning at scale. Yet past experience suggests that meaningful change comes from thoughtful integration, not overnight transformation. The enduring lesson is that educational systems are complex social organizations: lasting improvements require aligning technology with pedagogical principles, teacher readiness, and the needs of learners. In the following sections, we focus on one such alignment – the use of gamified learning activities, including quizzes and crosswords

– as an avenue where technology (and AI) can constructively enhance engagement and learning without claiming to "revolutionize" the classroom.

2.2 Gamification and Game-Based Learning in Education

In recent years, gamification has emerged as a popular strategy to increase student motivation and engagement. Gamification is commonly defined as the use of game design elements in non-game contexts [15].

In an educational setting, this means incorporating features like points, levels, badges, leaderboards, challenges, and instant feedback into learning activities in order to stimulate the kind of interest and perseverance that games elicit. Gamification is related to, but distinct from, game-based learning: the latter involves using actual games or simulations to achieve learning outcomes, whereas gamification often adds game-like structures onto traditional tasks [16]. In practice, the two approaches overlap – for example, a quiz turned into a competitive point-scoring exercise could be seen as both a gamified activity and a simple educational game. The theoretical rationale behind gamification draws on several well-established ideas in psychology and education. One is motivation: by introducing clear goals, feedback loops, and a sense of progress, gamified systems can satisfy learners' needs for achievement and recognition. Research guided by self-determination theory suggests that specific game elements can enhance intrinsic motivation by supporting autonomy, competence, and relatedness (e.g. students feel in control, see their skills improving, and enjoy a sense of community through friendly competition) [17].

Another relevant concept is flow – the state of deep focus and enjoyment that games often induce. Well-designed gamified tasks adjust difficulty and provide continuous feedback, helping learners enter a flow state which is hypothesized to improve their learning experience. Empirically, a growing body of evidence indicates that gamification can yield positive, though moderate, improvements in learning outcomes.

In quantitative research, such improvements are typically expressed using effect sizes—standardized statistical measures that indicate the magnitude of a treatment's impact, independent of sample size. Values around 0.2 are generally considered small, around 0.5 moderate, and around 0.8 large according to conventional benchmarks.

Two recent meta-analyses encompassing diverse educational contexts found that applying gamification techniques led to small-to-medium improvements in learning outcomes. Specifically, the reported effect sizes ranged from approximately 0.25 to 0.56 [18, 19]. In statistical terms, an effect size around 0.25 represents a small but meaningful gain, while values closer to 0.56 indicate a moderate improvement. These numbers capture the overall magnitude of gamification's impact across cognitive outcomes (e.g., improved test performance and knowledge retention), motivational outcomes (e.g., increased engagement and persistence), and behavioral outcomes (e.g., more consistent participation in class activities). Taken together, the evidence suggests that while gamification is not a "revolutionary" intervention, it produces reliable, positive effects that can significantly enrich traditional teaching when well-aligned with pedagogical goals.

These benefits include higher academic performance in some studies, as well as increased participation and persistence. For example, Sailer and Homner's meta-analysis showed that gamified learning usually outperforms traditional instruction by a modest margin, while also noting that effectiveness varies depending on the context and implementation details [18].

Similarly, Yıldırım and Şen found that gamification had a positive impact on students' academic achievement on average, though not uniformly so for all types of students or subjects [19]. It is important to note that not all gamification is equally effective. The literature suggests that the design elements used and the way they align with educational objectives matter greatly [19, 17].

Simply adding points or badges as an afterthought may have little lasting impact, or could even be counterproductive if it shifts focus away from deep learning toward winning the game. On the other hand, gamification that is thoughtfully integrated – for example, using narrative themes to connect challenges to curricular content, or using progress mechanics that adapt to a student's level – tends to yield better results [17].

In practice, many educators opt for lightweight gamification – adding game elements to quizzes, homework, or class activities – as a feasible way to boost motivation without needing to develop complex games from scratch. In summary, gamification, when aligned with sound pedagogy, serves as a motivational scaffold. It harnesses students' natural enjoyment of play and competition to encourage time on task, repeated practice, and resilience in the face of challenges. Rather than replacing traditional instruction, it augments it: points and leaderboards might turn rote drills into friendly competitions; leveling systems can encourage students to progress through content at their own pace; and game-like storytelling can contextualize problems in ways that make learning more meaningful. The key is to use these elements not as superficial rewards, but as catalysts for the behaviors and mindset that underpin effective learning (such as curiosity, effort, and collaboration). The next sections examine two specific gamified approaches – interactive quizzes and crossword puzzles – which exemplify how game elements can be applied in educational contexts, and review the evidence for their effectiveness.

2.3 Interactive Quizzes and Educational Challenge Platforms

One widespread application of gamification in the classroom is the use of interactive quizzes and challenge platforms. These tools, exemplified by systems like Kahoot!, Quizizz, Quizlet, and others, turn question-and-answer drills into fast-paced games. In a typical scenario, students answer multiple-choice questions on their own device while a leaderboard updates in real time, rewarding accuracy and speed. Such platforms often incorporate music, colorful visuals, and avatars to create a playful atmosphere, transforming assessment into a form of competition or self-challenge. The popularity of quiz games in both primary and secondary education speaks to their perceived benefits. From a pedagogical standpoint, frequent quizzing can serve as formative assessment (giving teachers immediate feedback on student understanding) and as practice for students through the

testing effect (retrieving information in quizzes has been shown to strengthen memory). The gamified aspect adds an element of excitement that can drive participation even for topics that might otherwise seem dry. Classrooms using tools like Kahoot! commonly report heightened engagement: students become eager to join, pay close attention to questions, and feel a sense of friendly rivalry that can energize review sessions. Empirical research largely supports these observations. A literature review covering 93 studies on the effects of using Kahoot! concluded that this type of gamified quizzing has a broadly positive impact on learning. Their analysis found improvements in learning performance, better classroom dynamics, and more positive student attitudes when such tools were used, as well as reductions in anxiety in test preparation settings [20].

In other words, students not only performed better on knowledge assessments, but also reported enjoying the process more and feeling less nervous about quizzes. Other studies have noted that the immediate feedback provided by interactive quizzes (students instantly see the correct answer and their standing) helps keep learners on track and can clarify misconceptions on the spot. Additionally, features like points and badges in these systems tap into extrinsic motivation, which can be especially useful for engaging students who might not be self-motivated in a given subject. Over time, this increased engagement can translate to more time spent on task and, ultimately, improved mastery of the material [20].

It is worth acknowledging potential pitfalls as well. Critics of gamified quizzing caution that if over-emphasized, the competitive aspect might discourage some learners or lead to focusing on speed over thoughtful reflection. To mitigate this, many teachers use these platforms in moderation or with cooperative modes (e.g. team quizzes) to ensure a supportive environment. When implemented thoughtfully, interactive quiz games exemplify the constructive integration of technology: they leverage well-known principles like retrieval practice and immediate feedback, wrapped in a game format that sustains student interest. In the context of this thesis, such quiz systems provide inspiration for the design of engaging educational activities. In particular, the move from static question sets to dynamic, AI-generated ones represents a new frontier – one that will be explored later with regard to automatic crossword generation. First, however, we examine the educational value of crosswords themselves as a form of gamified learning.

2.4 Crossword Puzzles as Learning Tools

Crossword puzzles have long been used in educational contexts as a playful way to reinforce terminology and concepts. A crossword typically consists of a grid of blank squares to be filled with letters, with clues provided for each word to be completed. Solving a crossword requires recalling facts or vocabulary (to answer the clues) and also using logic and spelling skills to make the answers fit together. This blend of recall and problem-solving makes crosswords a form of active learning: students cannot solve the puzzle passively; they must engage with the material, retrieve information from memory, and often deduce answers from partial letters.

For which regards the educational benefits of crosswords research across various subjects has observed that incorporating crossword puzzles into coursework can yield im-

provements in learning outcomes. Studies in medical, dental, and pharmacy education, for example, found that students who used crosswords to review key terms showed better retention of terminology and definitions [21, 22].

Puzzles appear to stimulate memorization and recall of technical knowledge by encouraging repeated retrieval: each clue effectively prompts the student to recall a fact or concept, reinforcing memory through practice. Moreover, there is evidence that the process of solving puzzles boosts students' confidence. As learners fill in more answers, they experience a sense of achievement that can increase their self-efficacy and satisfaction with the learning process [21, 22].

This motivational benefit is not trivial – feeling a small "win" after getting a word right can make students more willing to tackle the next clue, mirroring the way video games motivate players to continue through levels. Crosswords also inherently involve contextual learning. Unlike flashcards that present isolated facts, a crossword embeds terms in clues that provide semantic context or hints. This means students often have to think about the meaning of a concept, not just rote recall. For instance, a clue might be a definition, an application, or a word-play hint for the target term, requiring the learner to connect concepts and sometimes infer the answer. In doing so, crosswords can highlight relationships between ideas. They have been used to help students identify gaps in their understanding – when a particular clue stumps a student, it often reveals a weak spot in their knowledge that can then be addressed [22]. Some educators have used crossword puzzles as a form of self-assessment: if students struggle to complete a puzzle on a topic, it indicates areas where more study is needed.

Empirical studies support these qualitative benefits. In one quasi-experimental study in a medical education setting, an experimental group that learned with a hybrid method (lecture combined with relevant crossword puzzles) outperformed a control group (lecture only) on tests of knowledge both immediately after instruction and even one month later. The crossword-solving group also reported higher satisfaction with the learning experience [22]. The authors attribute this in part to the engaging, participatory nature of the puzzle activity, which helped maintain attention and encouraged repetition of key terms in a fun way [22]. Similarly, in undergraduate science courses, crossword puzzles used as review tools have been linked to improved exam performance compared to more passive review methods [23]. These positive results span disciplines: from biology to economics to language learning, educators have experimented with puzzles and generally find that students respond well to them as a supplement to traditional study methods [21, 23, 22].

Role in gamification and engagement

Crosswords exemplify the principles of gamification discussed earlier. They introduce a puzzle challenge (goal: fill the grid correctly) with clear feedback (letters either fit or they don't, clues have correct answers to be discovered). They can also incorporate game elements like competition or time pressure – for instance, giving a quiz grade for completing a crossword or having students race to finish. Some instructors award small prizes to the first few who solve a puzzle, introducing a competitive gamified element [21].

Even without overt competition, crosswords tap into the human love of puzzles and

riddles, which can make learning feel less like work and more like play. This can be particularly beneficial for younger learners in primary and secondary education; a crossword on a vocabulary list might be more appealing than a standard worksheet, yet it achieves similar practice objectives. It is important to design educational crosswords thoughtfully. The puzzle should be aligned with learning objectives – e.g., if the goal is to master scientific terminology, the clues should be definitions or applications of those terms, rather than obscure riddles. Clues can be tailored to different difficulty levels to accommodate learners' prior knowledge. For beginners, direct definition clues or word-bank style support can scaffold the activity, whereas more advanced students might handle cryptic clues or crosswords without word banks, which adds difficulty and problem-solving demand. In any case, crossword puzzles encourage multiple cognitive processes: recall (retrieving the answer), recognition (seeing how letters fit a partially completed word), and sometimes synthesis (when clue answers intersect, the letters of one answer help infer another). These processes embody active engagement with the material, which is a cornerstone of effective learning.

2.5 AI-Powered Generation of Gamified Learning Content

Given the demonstrated educational value of gamified quizzes and crosswords, a natural next step is to explore how Artificial Intelligence can expand and enhance these activities. Traditionally, creating a good educational crossword or quiz requires an instructor to invest time in crafting questions, clues, and puzzle layouts. This effort can be a barrier to using such tools frequently. However, recent advances in AI, particularly large language models (LLMs), suggest that much of this process can be automated. By leveraging AI, educators could generate a virtually unlimited supply of puzzles and quizzes tailored to specific topics, reading materials, or student proficiency levels.

Automatic crossword generation

Until a few years ago, automatically generating a quality crossword puzzle (with meaningful clues and a coherent grid) was a challenging AI problem. Early systems like WebCrow (an AI project that could solve and generate crosswords) demonstrated the feasibility of using algorithms and databases to tackle crosswords in multiple languages [12]. The WebCrow project, for instance, combined web search with linguistic resources to fill crossword grids and was even tested against human solvers. Building on such foundations, researchers have recently applied LLMs to the task of crossword puzzle generation specifically for educational content. Zeinalipour et al. (2023) have a series of studies in which they harnessed state-of-the-art language models to create crosswords in various languages, including English, Italian, and Arabic [11, 10]. These AI systems take academic texts or curriculum topics as input and generate crossword clues and answers that align with the content, effectively "quizzing" the material in puzzle form. What makes LLMs particularly powerful for this application is their ability to generate natural-language clues that are contextually appropriate. For example, given a glossary of terms or a textbook

chapter, an AI can formulate clues that test understanding of those terms, often matching the style of human-written crossword hints. Moreover, AI can ensure that the clues vary in format – from straight definitions to fill-in-the-blank sentences or even riddle-like prompts – which keeps the activity interesting for students. In addition to clue generation, AI can assist in constructing the crossword grid itself, arranging the answers to intersect appropriately. Recent research reports success in automatically creating crossword layouts and verifying that all intersections are valid, producing puzzles that are ready to use [11, 10].

A key advantage of AI-generated crosswords and quizzes is customization. An educator could, for instance, input the day's lesson text and receive a crossword puzzle that highlights the key vocabulary from that lesson. This could be especially beneficial in language learning (where daily new words can be practiced via puzzles) or in content-heavy subjects like history or biology (where a crossword can review important names and terms from a chapter). AI generation also opens the door to adaptive difficulty – puzzles can be made easier or harder on the fly, based on student performance data, by adjusting clue complexity or the inclusion of helper hints. Of course, the integration of AI raises new considerations. Educators must verify the accuracy of AI-generated content, as models can occasionally produce incorrect or ambiguous clues. The role of the teacher remains crucial – to select or curate the generated content and ensure it meets the learning goals.

In summary, AI-powered generation of gamified content like crosswords represents a promising convergence of technology and pedagogy. It builds on the theoretical foundations discussed in this chapter: the motivational power of gamification, the proven learning benefits of retrieval practice and active problem-solving, and the historical lesson that technology works best as an enhancer of teaching, not a replacement. By automating the creation of puzzles and quizzes, AI can help embed these engaging practices more seamlessly and frequently in education. The remainder of this thesis will delve into this convergence in depth. The next chapters will detail the design and development of an AI-driven system for automatic crossword generation tailored to educational needs. This work aims to contribute to the constructive integration of AI in education – not by heralding a revolution, but by providing a practical tool that empowers teachers and enriches students' learning experiences through gamified, interactive learning.

Part II

Methodology and implementation

Chapter 3

Architecture and Implementation of the Crossword Generation System

This chapter details the design and implementation of the Italian Crossword Generator system, which automatically creates educational crossword puzzles using generative AI. Building on the theoretical foundations of gamified learning and AI in education discussed earlier, we now focus on the technical methodology of our system. The system's architecture is explained with its two main workflows (generation from input text vs. from keyword list) and an interactive user revision loop. We then describe the dataset construction and a linguistic typology analysis of crossword clues, which informed our approach. Next, we elaborate on clue generation techniques: extracting keywords and generating clues via zero-shot and few-shot prompting for text inputs, and fine-tuning language models (GPT-3 variants and BERT) for clue generation from keywords. We also present the training and deployment of classifiers used to validate generated clues. The algorithm for generating the crossword grid (schema) from approved clue-answer pairs is then explained, including its scoring formula and stopping criteria. Finally, we outline the experiments and evaluation protocols for both the clue generation components and the validation classifiers, as well as a complementary analysis of clue difficulty using Surprisal to align puzzle difficulty with learners' levels.

3.1 System Architecture and Workflow

The overall system architecture is designed as a pipeline with two main pathways (Figure 3.1): (a) generation of clue-answer pairs from an input text (e.g. a lesson or article), and (b) generation of clues from a given list of target answers (keywords). These two paths converge into a final stage that constructs a complete crossword puzzle (grid and clues). Figure 3.1 illustrates the architecture with both pathways, including the integration of generation and validation components.

In Path (a), the system accepts a passage of text (for instance, a paragraph from a textbook) and automatically extracts a set of representative keywords (potential answers) from it. Then, using the input text as context, the system generates crossword-style clues for each extracted keyword via zero-shot or few-shot prompting of a large language model (LLM). In Path (b), the system accepts a list of answer words directly (with no source text) and produces a clue for each using a fine-tuned clue-generation model. In both cases, the raw generated clue-answer pairs are passed through validation filters to ensure quality and correctness. After validation, a user revision loop allows human oversight:

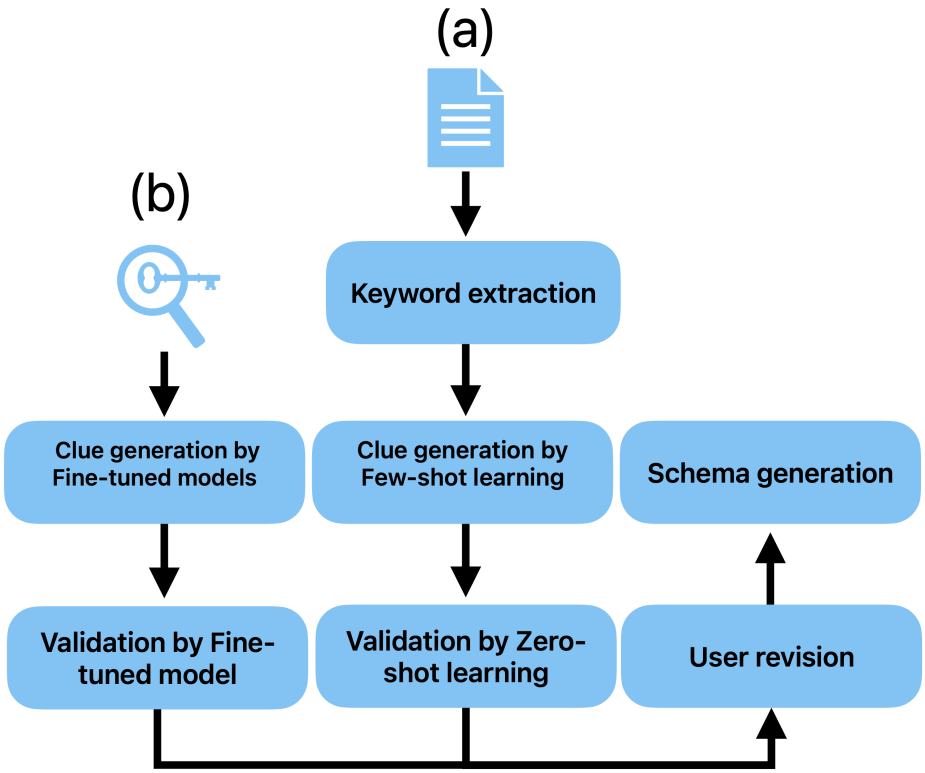


Figure 3.1: Overall system architecture. Path (a) Clue-answer generation from input text. Path (b) Clue generation from the given answers.

educators can review the candidate clue-answer pairs and select or edit those they find suitable. This step is important in educational settings to guarantee pedagogical relevance and accuracy. Finally, the approved clue-answer pairs serve as input to the crossword schema generator, which arranges the answers into a crossword grid and attaches the clues, yielding a complete puzzle ready for use.

Overall, the workflow is iterative and multi-stage, combining AI-driven generation and validation with human-in-the-loop revision. This ensures that the resulting crossword puzzles are not only automatically generated but also meet quality standards for educational content. The following sections describe each component in detail, including the underlying data resources and AI models.

3.1.1 Dataset Construction and Linguistic Typology

A large custom dataset of Italian crossword clues and answers was constructed to train and evaluate the system. We aggregated clues from multiple sources: online crossword clue databases (e.g. Dizy.com and Cruciverba.it) and digital archives of popular Italian crossword publications (such as La Settimana Enigmistica and La Repubblica puzzle sections). The raw data from these sources were cleaned, merged, and deduplicated to produce a high-quality corpus of clue-answer pairs.

The data cleaning and normalization pipeline was designed to ensure internal consistency across heterogeneous sources while preserving the linguistic integrity of the material. All raw entries were first converted to UTF-8 encoding and normalized using Unicode NFC normalization to guarantee consistent representation of accented characters and diacritics. HTML markup and duplicated whitespace were removed using rule-based regular expressions.

All clues and answers were converted to lowercase to ensure uniform lexical representation. Since crossword grid conventions in Italian do not allow accented characters in answers, answer strings were stored in their unaccented form. For the purpose of deduplication only, a temporary accent-stripped and apostrophe-normalized version of the text was used (e.g., “è” → “e”, “à” → “a”), while the original orthography was preserved in the clue field of the final dataset. Apostrophes were retained in the released dataset but normalized during duplicate detection to avoid mismatches due to typographic variation.

Answers were required to be single-token strings without spaces, in accordance with crossword formatting conventions. Entries containing numerical characters were excluded. One-letter answers were not present in the dataset, while two-letter metalinguistic answers (common in Italian crosswords) were retained.

Deduplication was performed exclusively through exact matching of normalized clue–answer pairs. Identical pairs were removed, whereas multiple distinct clues associated with the same answer were preserved, as this reflects standard crossword practice. No fuzzy matching or OCR-specific correction procedures were applied.

The preprocessing workflow was implemented using standard Python text-processing utilities, ensuring methodological transparency and consistency with established NLP preprocessing practices.

Regarding the legal status of such data in Italy, copyright protection applies to the crossword as a unified whole — including the grid schema and the complete set of clues — rather than to individual definitions in isolation. This protection is governed by Article 3 of the Italian Copyright Law (Law 633/1941), which categorizes crosswords as ’collective works’ protected by virtue of their creative selection and coordination [24, 25].

The final dataset contains 125,600 unique clue–answer pairs, covering diverse domains including history, geography, literature, science, and pop culture. Each entry consists of an Italian clue (often a short phrase or definition) and its corresponding answer word. The dataset is publicly available at the following link `https://github.com/tommyiaq/cruciverba` and is released under the Creative Commons Attribution-NonCommercial (CC BY-NC 4.0) license. This licensing choice ensures that the redistribution of the material remains limited to non-commercial academic research, thereby mitigating potential

conflicts with the rights of the original publishers.

Dataset overview

The collected clues exhibit a range of answer lengths and linguistic patterns. Figure 3.2 shows the distribution of answers by length. In general, shorter answers tend to have more clue variants, whereas longer answers are rarer (as answer length increases, the number of clues diminishes). For example, answers of 5–7 letters are very common and often have multiple distinct clues each, whereas answers longer than ∼10 letters are less frequent. This distribution is typical of crosswords and underscores the need for our generator to handle both short common terms and longer, more specific terms. Figure 3.2 illustrates the number of clue–answer entries for each answer length (blue bars) versus the number of unique answers of that length (red bars). The dataset also contains a variety of clue styles, from straightforward definitions to more playful or cryptic hints, which we exploit in model training.

Linguistic typology of clues

To better understand the clue formulations in our dataset, we performed a linguistic analysis focusing on the syntactic structure of clues. Each clue was parsed (using spaCy's dependency parser in Italian [26]) and classified into a taxonomy of syntactic patterns. We identified over 20 distinct clue structures, broadly falling into two categories: phrasal clues (non-sentential fragments, e.g. noun or adjective phrases) and clausal clues (full sentences or clauses). Within these, we further distinguished sub-types such as: nominal phrases (often definitions or noun descriptors), adjectival phrases (sometimes with a pronoun referring to the answer), active verb clauses, passive verb clauses, imperative forms, copular ("X is Y") statements, etc. Part of Speech (PoS) tagging allowed for a more precise separation of typologies and served as a reference framework for syntactic classification. Nonetheless, by examining the root node of each clue and progressively subdividing the dataset, it became possible to identify macro-categories (e.g., nominal vs. clausal) and to further refine them into micro-categories based on syntactic and morphological properties such as article type (definite vs. indefinite) or the presence of relative clauses. This hierarchical approach, rooted in generative grammar, enabled the extraction of mutually exclusive categories and provided a principled framework for classification.

A summary of the feature combinations employed in this categorization process is reported in table 3.1. This table highlights how Boolean feature configurations (e.g., clausal, art, definite, relCl) determine the assignment of clues to specific categories.

Table 3.2 shows the micro-category frequency distribution while 3.3 shows some example. Notably, the analysis revealed that non-clausal clues are slightly preferred over full sentences in the Italian dataset (the single most common structure type accounts for ∼23% of clues, meaning no one form dominates). Among verbal clues, active voice constructions are more frequent than passive ones, and simple present-tense statements are favored over more complex clause structures. We also observed frequent use of pronouns or clitic references within clues – e.g. clues that include a pronoun referring indirectly to the answer (as in "Pittoresco quello siciliano" = carretto, literally "Picturesque the

	clausal	cop	act	pass	imp_refl	inf	subj	cl	pron	nom	art	def	relCl	prep	adj	adv	pronom
cop:missSubj	✓	✓					✗	✗									
cop:clitic	✓	✓					✓	✓									
cop:pron	✓	✓					✓	✗									
act:missSubj	✓	✗	✓		✗			✗	✗								
act:clitic	✓	✗	✓		✗			✓									
act:pron	✓	✗	✓		✗		✓	✗	✓								
pass:missSubj	✓	✗		✓		✗	✗	✗									
pass:other	✓	✗		✓		✗	✓										
imp_refl:missSubj	✓	✗			✓	✗	✗	✗									
imp_refl:other	✓	✗			✓	✗	✓										
inf_VP	✓					✓											
bare_NP	✗									✓	✗		✗	✗	✗	✗	
bare_NP:rel	✗									✓	✗		✗	✗	✗	✗	
def_DP	✗									✓	✓	✓	✗	✗	✗	✗	
def_DP:rel	✗									✓	✓	✓	✓	✗	✗	✗	
ind_DP	✗									✓	✓	✗		✗	✗	✗	
PP	✗													✓			
adjP	✗							✗	✗						✓		
adjP:pron	✗							✗	✓						✓		
advP	✗															✓	
bare_Det	✗																✓

Table 3.1: Feature combinations used for clues categorization.

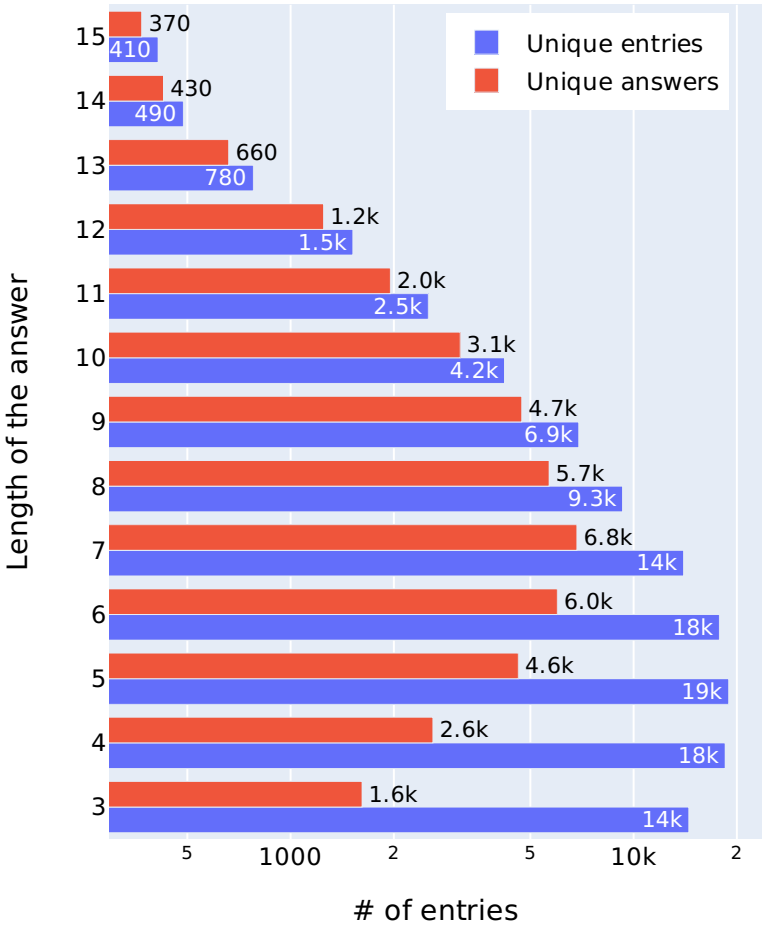


Figure 3.2: Distribution of the database entries by answer length, in blue the unique answer-clue pairs and in red the unique answers.

Sicilian one" = cart, where "quello" "the one" refers to the answer). This indicates Italian clues often embed the answer implicitly via pronouns, a stylistic insight that can guide our clue generation model. Overall, the linguistic typology suggests that while there are preferred formats (e.g. short phrases, active sentences), there is still substantial variety – offering ample scope for creativity in clue generation. These findings informed the design of our generation system; for instance, we aimed to produce a mix of clue types and avoid overly repetitive structures, to mirror human-crafted puzzles.

Table 3.2: Distribution percentages of Clue Types in the dataset.

Clue Category	Clue Macro-cat.	Tot. count	Tot. %	Avg Word Count
bare_NP	nominal	28571	23.05	3.26
def_DP	nominal	14892	12.01	4.42
metalinguistic	metalinguistic	14856	11.99	3.49
act:missSubj	verb_pred	13697	11.05	4.11
ind_DP	nominal	11135	8.98	4.13
adjP	adjectival	10929	8.82	2.96
act:clitic	verb_pred	5358	4.32	4.84
imp_refl:missSubj	verb_pred	4223	3.41	4.78
inf_VP	infinitive	3634	2.93	3.13
cop:missSubj	copular	2916	2.35	4.94
PP	prepositional	2383	1.92	4.19
other		1956	1.58	5.01
def_DP:rel	nominal	1648	1.33	7.00
cop:clitic	copular	1557	1.26	5.86
bare_NP:rel	nominal	1291	1.04	6.20
pass:missSubj	verb_pred	1214	0.98	5.11
cop:pron	copular	1205	0.97	5.64
bare_Det	pronominal	866	0.70	4.43
act:pron	verb_pred	509	0.41	5.76
imp_refl:other	verb_pred	471	0.38	5.80
advP	adverbial	285	0.23	2.65
adjP:pron	adjectival	241	0.19	4.90
pass:other	verb_pred	112	0.09	5.80

3.1.2 Clue Generation from Text (Zero-shot and Few-shot Learning)

When a teacher or user provides a body of text (such as a lesson transcript or an encyclopedia entry), our system automatically generates crossword clues from that text. This Path (a) (figure 3.1) workflow proceeds in multiple steps leveraging zero-shot and few-shot learning on large language models.

Figure 3.3 illustrates a concrete example of the text-based generation pipeline, showing how an input paragraph is transformed into validated clue–answer pairs through keyword extraction, clue generation, and filtering.

Keyword Extraction

The input text is first segmented into smaller chunks (e.g. paragraphs), to focus on one topical segment at a time. From each segment, the system extracts a set of keywords or key phrases that are both important in context and suitable as crossword answers. We treat this as a zero-shot task using an LLM: a prompt is constructed instructing the

Table 3.3: Typologies of linguistic clues with corresponding examples and macrocategories.

Macrocategory	**Typologies**	**Examples**
copular	cop:missSubj, copular sentence with subject omission	Fu Cancelliere della Germania dal 1949 al 1963 = *Adenauer*
copular	cop:clitic, copular sentence with a clitic in object position	Venere **ne** era la dea = *bellezza*
copular	cop:pron, copular sentence with a pronoun in object position	È celebre quella di Trinità dei Monti = *scalinata*
verbal predicate	act:missSubj, active verbal sentences with subject omission	Risiede in uno spazio geografico determinato = *abitante*
verbal predicate	act:clitic, active verbal sentences with a clitic in object position	La segue il medico = *ammalata*
verbal predicate	act:pron, active verbal sentences with a pronoun in object position	Quelli d'America hanno per capitale Washington = *Stati uniti*
verbal predicate	pass:missSubj, passive sentence with subject omission	È detta Il Continente Bianco = *Antartide*
verbal predicate	pass:other, other kinds of passive sentences	Vi furono ritrovati noti bronzi = *Riace*
verbal predicate	imp_refl:missSubj, active sentence with impersonal pronoun or reflexive verb with subject omission	Si reca spesso al catasto = *geometra*
verbal predicate	imp_refl:other, other kinds of active sentence with impersonal pronoun or reflexive verb	Che si riferisce all'Università = *accademico*
infinitive	inf_VP, infinitival verb phrases (VP)	Investire di un grado = *nominare*
nominal	bare_NP, bare noun phrases (NP)	Infuso paglierino = *tè*
nominal	bare_NP:rel, bare NP followed by a relative clause	Cilindri commestibili che vengono affettati = *polpettoni*
nominal	def_DP, definite determiner phrases (DP)	Il conto delle spese da farsi = *preventivo*
nominal	def_DP:rel, DP followed by a relative clause	Lo Stato di cui fanno parte le Isole Azzorre = *Portogallo*
nominal	ind_DP, indefinite DP	Una brutta abitudine perdonabile = *vizietto*
prepositional	PP, prepositional phrases	Davanti a Rodrigo = *Don*
adjectival	adjP, adjectival phrases	Probo, retto = *onesto*
adjectival	adjP:pron, adjectival phrases with pronoun	Pittoresco quello siciliano = *carretto*
metalinguistic	two-letters answer	Il centro di Matera = *TE*

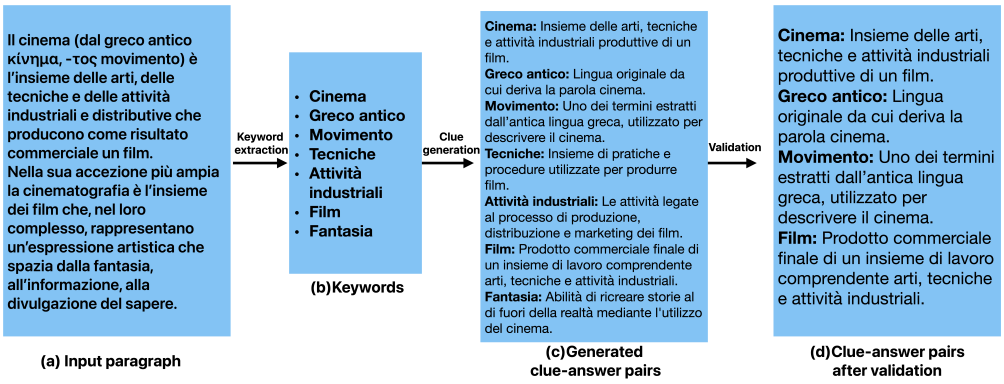


Figure 3.3: Running example of Path (a): from an input educational paragraph to keyword extraction, clue generation, and validation of clue–answer pairs.

model (GPT-3) to read the text and return a list of the most important terms or concepts (nouns, names, technical terms, etc.). The prompt is designed to produce single-word or short multi-word terms that represent key ideas in the paragraph – effectively simulating what a human instructor might highlight as glossary terms. We crafted prompt templates in both Italian and English for this task, to compare which yields better extraction performance. The zero-shot extractor performed surprisingly well: in our evaluations it achieved about 79.7% precision in selecting suitable keywords from text when using an Italian-language prompt, and about 75.6% accuracy with an English prompt (accuracy here means the percentage of extracted keywords that were judged appropriate for use as crossword answers in the given context). This step outputs a set of candidate answers drawn from the input text.

Clue Generation via Prompting

For each extracted keyword (answer), we next generate a clue sentence or phrase that relates to the source text. We use a few-shot prompting approach with GPT-3 (specifically, the Davinci model) to accomplish this. The prompt provides a few examples of (text excerpt → keyword → clue) to guide the model, then asks it to produce a clue for the new keyword given the surrounding paragraph. Essentially, the model is tasked with formulating a clue that could lead a student from the keyword to something mentioned or explained in the text. The clues generated in this way often take the form of definitions or descriptions that reflect information from the source material, making them well-aligned with the educational content (e.g., if the text is about electricity, a generated clue for the keyword "voltage" might be "Unit measuring electrical potential difference," mirroring the text's explanation). We created prompt variants in Italian and English; interestingly, using English instructions while still expecting Italian clues improved the output quality. This could be due to the model's stronger training in English for following instructions. In our tests with 50 text passages, about 68.3% of the clues generated using Italian-language prompts were rated as acceptable (valid, relevant clues), while using English prompts on

the same Italian texts yielded about 76.7% acceptable clues. This indicates the few-shot mechanism can produce good clues most of the time, though some outputs are flawed.

Zero-shot Validation of Clues

To filter out the flawed or nonsensical clue-answer pairs, we integrate a zero-shot validation step. Some clues, for example, might inaccurately describe the target concept or introduce facts not present in the text (hallucinations). We address this by prompting another language model to evaluate each generated clue-answer pair without any fine-tuning. The validator prompt might say: "Here is a crossword clue and its proposed answer based on the text. Does the clue correctly and sensibly refer to the answer? Yes or no?" – allowing the model to act as a semantic consistency judge. This zero-shot approach proved effective at catching many bad pairs: it identified roughly 57% of the unacceptable pairs in the Italian-prompt case and about 70% of the unacceptable pairs in the English-prompt case (the higher figure aligns with the English prompt yielding more consistent clues to begin with). We remove any clue-answer pairs flagged by this automatic check. Overall, this multi-stage filtering significantly improves final quality. As reported, the validator eliminated a majority of hallucinated or incorrect clues, resulting in a notable improvement in the final output of Path (a). After validation, the remaining high-quality clues (typically a handful per text passage) are presented to the user for review.

User Review

The educator or user can then review all the validated clue-answer pairs extracted from the text. Through a simple interface, the user can see each proposed answer and its clue, compare them with the source text if needed, and decide which to keep. The user may discard any that are not pedagogically relevant or adjust wording if desired. This revision loop ensures that the final set of clues used in the crossword is curated for relevance to the lesson. Only the user-approved clues proceed to the puzzle construction phase.

In summary, Path (a) leverages the generative power of a large model in a prompt-based manner to create clues grounded in a given text, and combines it with automated and human verification. This approach requires no fine-tuning for clue generation; instead it relies on the zero/few-shot capabilities of GPT-3, which made it quick to experiment with different prompt designs. The results demonstrate that even zero/few-shot methods can achieve a solid baseline of ∼70–77% good clues, which can be further cleaned by validators and humans to ensure only high-quality educational clues are used.

3.1.3 Clue Generation from Keywords (Fine-tuned Models)

In scenarios where a specific list of target answers is already known (for example, a teacher might supply a set of vocabulary words or key terms for review), the system can generate clues for those answers without an input text. This Path (b) (figure 3.1) uses fine-tuned language models to directly generate clues from given answers. We formulated this as a supervised learning problem: given an answer word, produce a suitable crossword clue

for it. To create a training set, we leverage our dataset of 125k Italian clue-answer pairs (which inherently contains examples of clues for various answers).

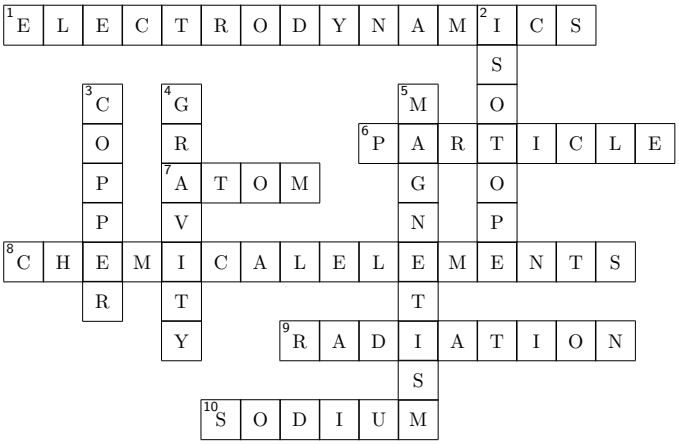
Fine-tuning LLMs for clue generation

We fine-tuned two GPT-3 model variants on a large subset of our dataset specifically for clue generation. The models were GPT-3 Davinci (175B parameters) and GPT-3 Curie (13B parameters). We chose these because Davinci represents a highly capable LLM, while Curie, though smaller, still has strong language skills at lower computational cost. In both cases, the fine-tuning objective was: input the answer, output a clue. We structured each training example as a short text like "ANSWER: <answer>CLUE: <clue>", so the model learns to associate an answer with an appropriate clue. No additional context or prompt was given during fine-tuning (we rely on the model to internalize the clue-writing style from the examples). To control training cost, we did not use the entire 125k dataset; instead, we randomly selected 50,000 clue-answer pairs for training the generators. We trained for 3 epochs with a batch size of 16 and a learning rate of 0.01. These hyperparameters were found to work well in preliminary trials – the modest epoch count helps avoid overfitting to the small nuances of specific clues, aiming instead for general clue-writing ability.

During fine-tuning, we intentionally fed the answer as the only input and required the model to produce the clue text. By repeatedly exposing the model to this task, we essentially teach it the "crossword clue style" for various types of answers. This includes learning to incorporate definitions, synonyms, or wordplay common to crossword clues. For example, the model sees pairs like ("Rome" → "The eternal city") and ("photosynthesis" → "Process by which plants make food"), learning to generate descriptive or defining phrases. We found that an iterative approach – always providing the answer and having the model generate a clue – helped the model focus on how clues are formulated given a target answer. In effect, the model hones a nuanced understanding of clue construction, guided by the distribution of our dataset which spans many topics and clue styles. After fine-tuning, we have two generator models: one based on Davinci and one on Curie.

Generation and initial results

Using the fine-tuned models, we generate clues for new inputs by simply feeding each desired answer into the model (with the prefix format if required, e.g., "ANSWER: <answer>CLUE:") and letting it produce a completion (the clue). We tested the models on a held-out set of answers to evaluate their performance. Specifically, we had each model generate clues for about 4,000 answers that were not seen during training. We then conducted a human evaluation on these outputs, following the same guideline criteria as used in Path (a) (an annotator judges whether each generated clue is acceptable/correct for its answer). The results showed a remarkable difference in quality between the two model sizes. The GPT-3 Davinci model produced 60.1% acceptable clues, outperforming GPT-3 Curie which produced 34.9% acceptable clues. In other words, nearly two-thirds of Davinci's outputs were valid clues, compared to only one-third for Curie. This gap illustrates



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 3.4: An illustrative crossword created using the newly introduced system.

the benefit of model scale and capacity in capturing the complexity of clue-generation: the larger Davinci model can pick up more subtle semantic relationships and produce more clue-like definitions, whereas Curie often struggled (many of Curie's clues were too vague or outright incorrect). For example, one acceptable clue generated by Davinci was "Mitologia: La conosce chi conosce i miti" (Mythology: whoever knows myths knows it), which is a reasonable definition, whereas an unacceptable Curie-generated clue was "Collaborazione: Lo si raggiunge con chiunque" (Collaboration: you achieve it with anyone) which is incoherent as a definition. These examples highlight typical model success and failure modes. Overall, the fine-tuned Davinci model proved to be a strong clue generator in Italian, and we selected it as the primary generation engine for Path (b) in our system.

After generation, Path (b) also funnels its outputs into a validation stage (described above) to filter out the flawed clues. In practice, we integrate both generation paths: the system can take both a source text and additional teacher-provided keywords, generate clues via the appropriate route, and then unify all clues for validation and puzzle construction. Indeed, we often combine clues derived from the input text (Path a) with clues for extra keywords that might not have explicitly appeared in the text (Path b) – a strategy that can enrich the puzzle. For instance, in one example we took a text about bridges and engineering and also added some related keywords like famous bridge names; the final puzzle used both sets of clues. In Figure 3.4 from [11], a completed crossword layout is shown that mixes clues extracted from text with clues generated solely from teacher-specified keywords. This demonstrates the flexibility of our hybrid approach.

Clue Validation and Classifier Training

Not all generated clues will be of acceptable quality, so a critical part of our methodology is automatic validation: distinguishing good clue-answer pairs from bad ones. We tackle this with a combination of fine-tuned classifier models and zero-shot reasoning, which serve as filters before the crossword is assembled. The classifier approach was primarily applied to Path (b) outputs (where we had ample training data for supervision), but can also be used generally to validate any clue-answer pair.

Labeled data for validation

To train the classifiers, we needed examples of both acceptable and unacceptable clue-answer pairs. We leveraged the outputs of our fine-tuned generation models to create this. As mentioned, we generated thousands of clues from GPT-3 Davinci and Curie. We then had human annotators label a large sample of these as acceptable (valid clue) or unacceptable under strict guidelines. In total, we compiled a labeled dataset of ∼6,000 clue-answer pairs with human judgments. About 51% of this set were acceptable and 49% unacceptable, providing a balanced training set for classification. These include a diverse range of cases – from clearly correct, well-formed clues to ones with various errors (semantic mismatches, grammar issues, overly vague hints, etc.). For example, an unacceptable pair might be ("Elettricità" : "Uno dei segni zodiacali") where the clue is nonsensical for the answer. Acceptable ones include cases like ("Curiosità" : "Il desiderio di sapere") which is a reasonable definition ("Curiosity: the desire to know"). This labeled corpus was split 80/20 into training and test sets for the classifier models.

Fine-tuned classifier models

We experimented with a range of language models fine-tuned for classification, to see which is most effective at identifying bad clues. The models included the GPT-3 family of different sizes as well as a BERT model, specifically: GPT-3 Davinci, GPT-3 Curie, GPT-3 Babbage (1.3B parameters), GPT-3 Ada (350M parameters), and BERT-base (uncased) ∼110M. Each model was fine-tuned in a binary classification setup on the 4800 training pairs, with labels indicating acceptability. We gave the models the clue and answer (concatenated, possibly with a separator) as input. The goal was for the model to output either "acceptable" or "unacceptable" (or a probability thereof). The intuition was to leverage the linguistic knowledge in these pre-trained models: larger models might better grasp subtle semantic correctness, while smaller ones offer efficiency. Fine-tuning was done using standard text classification methods (cross-entropy loss on the label). We also employed a zero-shot classifier in parallel: for comparison, we used GPT-3 Davinci in prompting mode (without fine-tuning) to judge clues (essentially the same approach as described in Path (a) validation). This zero-shot method doesn't require training but may not be as consistent as a model explicitly fine-tuned on our domain-specific examples. The various classifiers were evaluated on the 20% test split (∼1200 pairs). The results, shown in Table 3.4, indicate that larger models perform significantly better at this task. The fine-tuned GPT-3 Davinci classifier achieved about 79.9% accuracy, with an F1 score

of 0.784 (on the balanced dataset). The next-best was GPT-3 Curie at ∼77.8% accuracy. Performance drops as model size decreases: Babbage ∼74.1%, Ada ∼69.2%, and the BERT-base classifier reached only 65.6% accuracy (F1 ≈ 0.64). The largest language model (Davinci) was the most adept at discerning subtle issues in clues, likely because it can better understand if a clue logically and factually fits the answer. For instance, Davinci can "know" that electricity is not a zodiac sign, whereas a smaller model or BERT (trained mostly on general language modeling) might not make that connection as reliably. The gap between 79.9% and 65.6% is quite significant in practice – we found that BERT would miss many errors or misclassify correct clues as wrong, whereas Davinci was much more reliable. We opted to use the GPT-3 Curie or Davinci classifiers in the final pipeline (as a trade-off between accuracy and API usage cost), with Curie already giving us ∼78% accuracy which was acceptable for a first-pass filter.

Table 3.4: Classifier performance on distinguishing acceptable Clue-Answer pairs

Model	*accuracy %*	*precision %*	*recall %*	*F1 Score*
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-base	65.62	63.71	64.47	0.6409

In deployment, the classifier works as follows: after the clue generation step (either path), each clue-answer pair is passed to the classifier model, which predicts whether it is acceptable. We discard any pair that the classifier labels as unacceptable. This complements the earlier zero-shot filtering; in fact, one can combine them (e.g., only accept a clue if both the zero-shot check and the classifier agree it's good, or use the classifier as the primary filter). Our best configuration was using the fine-tuned classifier as the main gate, since it's trained specifically on our clue domain. According to our evaluations, the validation component (combining classifier and zero-shot approaches) was able to detect about 70% of the flawed clue-answer pairs automatically, substantially reducing the burden on human reviewers. In the final system, after this automated filtering, the user typically only sees a relatively small number of candidate clues (the majority of which are already good), from which they make the final selection as described earlier.

3.1.4 Crossword Grid Generation Algorithm

Once a set of validated clue-answer pairs has been finalized (typically on the order of 5–15 words for a single puzzle, depending on the desired size), the system proceeds to construct the crossword grid or schema. This module places the answer words into a blank grid, forming the interlocking pattern characteristic of crosswords, and attaches each clue to its placed answer (Across or Down in the grid). We developed a custom backtracking

search algorithm to generate compact and well-connected crossword layouts.

It is important to note that *educational crosswords* built with such a limited number of target keywords differ substantially from traditional newspaper-style crosswords. In standard crosswords, the grid is almost completely filled, leaving only a few black squares as separators. In contrast, educational or domain-specific crosswords intentionally cover a smaller portion of the grid, as shown in Figure 3.4. This sparser configuration results directly from the pedagogical constraint of focusing on a restricted and coherent thematic set of terms. Completely filling the grid would require a much larger vocabulary, which in turn risks introducing unrelated or out-of-topic words that would dilute the educational focus of the puzzle. The chosen balance thus ensures that each generated crossword remains both thematically consistent and cognitively manageable for learners.

Input and configuration

The schema generator takes as input the list of answers (usually all in uppercase for the grid), along with optional parameters like the maximum grid dimensions and certain stopping criteria. The algorithm is flexible to different sizes; for example, for a typical educational puzzle we might allow a grid up to 15x15 cells, but if fewer words are provided it can produce a smaller grid. The user can also designate preferred answers that must be included – the algorithm will prioritize placing those words first.

Placement strategy

The algorithm (Algorithm 1) begins by placing one answer in the grid as a starting point. We often choose the longest answer or a "central" word to place first in the center of the grid for a balanced layout. It then iteratively adds the remaining words one by one. Each new word is placed such that it intersects with at least one of the words already on the grid (as required in a valid crossword). We attempt to match new words with existing ones by shared letters: the algorithm looks for any letter in the new word that matches a letter of any already placed word, and if so, tries to place the new word crossing at that letter position. If multiple placement options exist, it can choose one at random or based on a heuristic (e.g., favoring placements that yield more future crossing possibilities). If a word cannot be placed with any overlap (without violating grid constraints or causing letter mismatches), the algorithm may skip that word and try another. We use a backtracking approach: if later a word cannot be placed, the algorithm may remove some recently placed words (backtrack) and try alternative placements or a different order of insertion. In some cases, if the layout becomes too fragmented or a dead-end is reached, the algorithm can restart from scratch with a different initial word or a different random placement order. This stochastic element helps explore different puzzle layouts.

Scoring and selection of best grid

Because the algorithm involves some randomness (especially in word ordering and placement choices), it can generate multiple candidate solutions in one run. We incorporate a

scoring function to evaluate each completed grid and pick the best one. The score of a crossword grid is designed to favor puzzles that use many of the words and have a tightly interlocking structure. Specifically, we define a formula:

$$\text{Score} = (FW + 0.5 \times LL) \times FR \times LR \tag{3.1}$$

where:

- FW = Filled Words = number of words placed in the grid,
- LL = Linked Letters = number of letters that are used in two crossing words (i.e. intersection points),
- FR = Filled Ratio = (number of filled letter cells) ÷ (area of the smallest rectangle that covers all placed words),
- LR = Linked Letters Ratio = (LL number of filled letter cells) ÷ (total number of letters)

This scoring function rewards: using more of the given words (higher FW), creating many intersections (higher LL and LR), and packing the words into a small area (higher FR). The term $FW + 0.5 \times LL$ gives base credit for each word placed, with a bonus for each crossing letter. Multiplying by FR and LR further boosts solutions that are dense and have a high proportion of cross-letters. In essence, a fully interlocked, compact crossword with all words used would score highest.

Algorithm 1 Crossword Grid Filling Algorithm

Require: List of answer words, grid size, must-have words, stopping criteria
  Initialize empty grid of given size
  Place the first word (e.g., longest or central) at the grid center
  Initialize `addedWords` and `missingWords` lists
  while stopping criteria not met **do**
    if all words placed or grid sufficiently dense **then**
      Compute score for current grid
      if score better than previous best **then**
        Save current grid as best solution
      end if
      Optionally restart with new initial word or placement order
    else
      Select next word from `missingWords` (prioritize must-have)
      Find possible placements by matching letters with placed words
      if placement found **then**
        Place word in grid at intersection
        Move word to `addedWords`
      else
        Backtrack: remove recent words, try alternative placements
      end if
    end if
  end while
  return Best grid found, with clue-to-position mapping

After each attempted full placement, the algorithm computes this score. It continues trying new placements (with backtracking and restarts) until certain stopping criteria are met. The stopping conditions include: reaching a desired minimum number of words placed or a satisfactory filled ratio, exceeding a maximum number of retries/resets, or hitting a time limit. For example, we might stop after X attempts or after Y seconds and take the best solution found so far. We also stop early if a perfect or near-perfect solution is achieved (e.g., all input words placed with high density). The solution with the highest score is then output as the final crossword schema.

The chosen crossword schema is output as a grid with coordinates of each letter, along with the mapping of each clue to its grid position (Across or Down, number labels, etc.). An example output layout is shown in Figure 3.4. The figure demonstrates that the algorithm successfully fit all given words into a neat grid, with plenty of intersections (the puzzle is visually similar to a human-crafted crossword). Our algorithm ensures a minimum connectivity – by design, every word added crosses at least one other, so the final grid is one connected component. If desired, the algorithm can emphasize certain words by using the "preferred answers" feature: those words are given priority in placement, effectively increasing the probability they appear in the final puzzle. This is useful if an instructor absolutely wants certain key terms included; the algorithm will sacrifice some optimality (if necessary) to include them.

The efficiency of the schema generator is good for moderate puzzle sizes. With typical inputs (10–20 words of lengths 4–10 letters), the algorithm finds a solution within seconds. We included safeguards (max iterations/time) to avoid worst-case exponential backtracking scenarios. In practice, the combination of scoring and smart restart criteria leads the search towards viable configurations quickly. The result is an automated crossword grid generator that can take a set of vocabulary words and produce a valid crossword layout with minimal empty space and maximal word intersections. Our heuristic approach, guided by a scoring metric, proved effective for the scope of educational puzzles.

3.1.5 Human Evaluation Guidelines

The quality of generated clue-answer pairs was assessed through manual evaluation according to a predefined annotation framework grounded in standard crossword construction principles.

Each pair was labeled as *acceptable* or *non-acceptable* based on the following criteria:

- **Semantic coherence:** the clue must correctly describe or refer to the answer.
- **Clarity and unambiguity:** the clue should admit a single plausible interpretation.
- **Grammatical correctness:** the clue must respect linguistic conventions.
- **Crossword appropriateness:** the answer must comply with crossword formatting constraints.
- **General knowledge fairness:** the clue should not rely on excessively obscure or misleading references.

This structured framework ensures consistency across evaluation phases and provides a principled basis for distinguishing high-quality clue-answer pairs from inadequate ones. Since the evaluation was conducted by a single annotator, inter-annotator agreement metrics (e.g., Cohen's κ) were not computed. While this setup guarantees internal consistency in labeling, it may introduce subjective bias, particularly in borderline cases.

3.1.6 Experiments and Evaluation

This section summarizes the experimental evaluation of the proposed crossword generation system. The evaluation focuses on three main components of the pipeline: (i) keyword extraction and clue generation from input texts (Path a), (ii) clue generation from predefined keyword lists using fine-tuned language models (Path b), and (iii) the automatic validation and filtering of generated clue–answer pairs. For the text-based pipeline, experiments were conducted on a set of 50 educational Wikipedia passages, evaluating keyword extraction precision and clue acceptability through human annotation. For the keyword-based pipeline, the performance of fine-tuned GPT-3 models was assessed on 4,000 unseen answers, measuring the proportion of acceptable clues generated. In addition, classifier models trained on a labeled dataset of approximately 6,000 clue–answer pairs were evaluated for their ability to automatically detect invalid clues.

Evaluation of Clue Generation from Text

For Path (a), we evaluated the keyword extraction and clue generation quality using a set of 50 text passages from Wikipedia on educational topics (science, geography, economics, etc.). The passages were manually selected from Wikipedia articles covering topics broadly aligned with standard school curricula. The goal was to include texts representative of educational domains typically addressed in secondary education (e.g., scientific, geographical, historical, and economic subjects). Texts were chosen to have comparable length and informational density in order to ensure consistency across evaluation samples. Each passage was processed by our system using both the Italian-prompt and English-prompt variants, to compare performance. We then measured: (1) the precision of keyword extraction – what fraction of extracted keywords were deemed suitable answers, and (2) the acceptability of generated clues – what fraction of clues were valid and relevant to the text (as judged by a human expert, in this case a native Italian linguist). The evaluation was conducted following the guidelines described in Section 3.1.5

It is important to clarify that the reported percentages for acceptable keywords and acceptable clues do not represent classification accuracy in a strict statistical sense. Rather, they correspond to the empirical precision of the generation process, defined as the proportion of outputs judged acceptable over the total number of generated outputs.

Table 3.5: Assessment outcomes of the clue-answer pairs generated from the provided Text.

System part	*Italian Prompt*	*English Prompt*
Acceptable keywords	79.73 %	75.60%
Acceptable clues	68.34 %	76.70 %
Validator performance	56.76 %	69.72 %

The results show (Table 3.5) the approach is effective. The zero-shot keyword extractor achieved 79.73% precision with Italian prompts and 75.60% with English prompts. In other words, roughly 4 out of 5 extracted terms were good puzzle answers (the rest were either too generic, too obscure, or too long multi-word terms not suitable for a crossword). This performance is quite good given no training was involved – it demonstrates that an LLM can generalize the concept of "important keyword" from a single prompt.

For clue generation, the Italian-prompt pipeline yielded 68.34% acceptable clues, whereas the English-prompt pipeline achieved 76.70% acceptability (Table 3.5). This difference suggests that prompt language influenced model performance. Previous studies have observed that multilingual large language models, whose training data is predominantly English, may exhibit stronger instruction-following behavior when prompts are formulated in English, even for non-English downstream tasks [27]. In our case, the English instructions appear to have encouraged more structurally constrained and definition-oriented outputs, which align more closely with standard crossword clue conventions. Nonetheless, both prompting strategies produced a majority of acceptable clues, confirming the effectiveness of the few-shot prompt design.

To further validate clue quality, we applied the zero-shot validator to the generated pairs. According to our analysis, the automated validation identified and filtered out about 56.8% of the unacceptable clues in the Italian-prompt output, and 69.7% of the unacceptable clues in the English-prompt output. This means the validator managed to catch roughly two-thirds of the errors on its own, especially in the scenario where clue generation was stronger (English-prompt). After this filtering, the remaining clues had a very high acceptance rate – effectively, the precision of the clues passed forward was greatly improved.

Combining all steps, our system's end-to-end performance on Path (a) was as follows: around 80% of keywords extracted were judged acceptable by the annotator, ∼70–77% of clues initially acceptable, and ∼70% of the bad ones filtered out. Ultimately, well over 90% of the clues presented to the user for final selection were acceptable. Table 3.5 and 3.4 presents these outcomes (keyword accuracy, clue acceptability, and validator detection rate for each prompt type). This level of performance is quite encouraging for automatic content generation from arbitrary input text. Qualitatively, the acceptable clues were found to be on par with straightforward human-written clues (often definitions or rephrasings of the text), whereas the unacceptable ones tended to be either too vaguely related or occasionally humorous misinterpretations by the model. Representative examples of acceptable and unacceptable clue–answer pairs, evaluated according to the annotation guidelines described in Section 3.1.5, are reported in Table 3.6.

Table 3.6: Examples of acceptable and unacceptable clue–answer pairs with rejection motivation according to the evaluation guidelines.

Clue–Answer pair	*Acc.*	*Motivation*
Mitologia: La conosce chi conosce i miti	Yes	Semantically coherent
Elettricità: Uno dei segni zodiacali	No	Semantic mismatch
Curiosità: Il desiderio di sapere	Yes	Precise and unambiguous
Collaborazione: Lo si raggiunge con chiunque	No	Vague and weak semantic relation

The human annotator noted that most unacceptable clues were obviously wrong (and thus easy to filter), and very few borderline cases slipped through. This gives confidence that the system can reliably assist teachers by automatically generating quiz-style clues from lesson materials, needing only minimal editing.

Evaluation of Clue Generation from Keywords

For Path (b), we focus on the output of the fine-tuned generation models. As described, we generated clues for 4,000 unseen keywords with each model (Davinci and Curie) and performed human evaluation. The headline result was the stark difference in quality: 60.1% acceptable clues from GPT-3 Davinci versus 34.9% from GPT-3 Curie. These percentages highlight that the largest model can capture a lot of clue-writing nuances even with only 50k training examples, whereas the smaller model was not fully adequate. We also observed in practice that Davinci's clues were often lengthier and more descript-

ive, whereas Curie sometimes produced single-word "clues" (just a synonym or a related word, which was usually not sufficient as a clue). The human evaluators applied the same standards – a clue was acceptable if it was a valid definition, synonym, or clear hint for the answer, and unacceptable if it was wrong or nonsensical. The gap in acceptability underscores the value of using state-of-the-art LLMs for generation when possible. However, even 60% acceptable means 40% were flawed, reinforcing the need for the validation step.

We next evaluated the combined generation+validation pipeline on the keyword task. For this, we employed the fine-tuned classifiers on the outputs. We used the previously mentioned test set of 1,200 labeled pairs for an objective measure. The best classifier (Davinci-based) achieved ∼79.9% accuracy in flagging unacceptable clues, which implies it catches most of the bad ones while rarely rejecting good ones. For instance, it correctly labeled clues like "Electricity: One of the zodiac signs" as unacceptable, and clues like "Mitologia: La conosce chi conosce i miti" as acceptable. The overall precision on acceptable predictions was about 80%, and recall ∼77% (for the Davinci classifier), indicating a balanced performance. Using a slightly less powerful classifier (Curie-based) would give ∼78% accuracy, which is still serviceable. In a live system, we might favor the smaller Curie model to reduce cost, at the expense of a few more missed errors. The end-to-end result for Path (b) can be summarized by the conclusion that the Davinci generator + Davinci classifier duo yields roughly 60% x 80% ≈ 48% of outputs ultimately accepted automatically. In practice, we would generate more candidate clues than needed, so even if half are filtered out, we still have plenty to choose from for a puzzle. Indeed, we typically generate multiple clue options for each answer (e.g. by sampling the model or using both Davinci and Curie generators) and then let the classifier/validator select the best.

We also looked at some qualitative outcomes of the keyword-based generation. Common issues with unacceptable clues included: mismatched scope (clue describes something much broader or different than the answer), hallucinations (clue introduces a fact or relation that is untrue), or overly generic statements that don't uniquely identify the answer. Acceptable clues were usually concise definitions or well-known descriptors of the answer. For example, for the answer "mitologia" (mythology), Davinci gave "La conosce chi conosce i miti" (He who knows myths knows it) – a clever phrasing implying that knowing myths means knowing mythology. For "curiosità" (curiosity), Curie produced "Il desiderio di sapere" (The desire to know) which is an excellent definition. On the other hand, an example of failure: for "elettricità" (electricity), Davinci produced "Uno dei segni zodiacali" (One of the zodiac signs), which is completely off (perhaps conflating electric with Aquarius symbol or something bizarre). These examples illustrate why automatic validation is indispensable. We find that most mistakes are not subtle – they are glaring, as if the model momentarily loses the thread – which the classifier can detect.

In summary, the experiments for Path (b) demonstrate that with sufficient fine-tuning data, an LLM can learn to generate a good proportion of valid clues from keywords alone, and that a learned classifier can successfully filter out most of the bad ones. The combination yields a solid pipeline for automatic clue generation from arbitrary word lists. This is particularly useful for educators who might have a set of terms (e.g. weekly vocabulary) and want to create a puzzle: our system can now do so with only minor

human curation needed.

Crossword Schema Construction Results

Evaluating a crossword grid generator is somewhat different from evaluating NLP components, as it involves combinatorial completeness rather than accuracy per se. Our crossword schema algorithm was tested by feeding it various sets of words (both the ones extracted from texts and some manually provided sets) and examining whether it produced valid, well-formed puzzles. Key metrics we considered were: success rate (does it place at least a target number of words before stopping), grid compactness (as measured by filled ratio FR and linked letter ratio LR), and computational efficiency (time or attempts needed).

In our trials, the algorithm almost always succeeded in finding a layout that included all or most of the input words, given reasonable grid size limits. For example, using the 10 keywords extracted from a Wikipedia paragraph on geography, it placed all 10 into a 13×13 grid with a filled ratio of ∼0.6 and linked letter ratio ∼0.3 (meaning 30% of letters were shared by crossing words). The algorithm's scoring function effectively guided the search: solutions that didn't meet the minimum criteria (like too few intersections or too spread-out) were naturally outscored by better ones, and the search would continue. We set a stopping criterion that at least 80% of the words must be placed and FR >0.4, otherwise it restarts – this was typically achieved after a few tries. The best solutions often had all words included and a dense packing (FR >0.5, LR >0.5). In cases where certain words had rare letters or no overlap possibilities with others, the algorithm would occasionally drop one or two low-compatibility words (especially if we allowed that in parameters) to complete a grid with the rest; this is where the "preferred answers" option is useful if a particular word is a must-have.

One representative result is visualized in Figure 3.4, which shows a filled crossword grid for a mix of text-derived and user-provided words. All clues for that example were generated by our system, and the layout is nicely connected. The figure highlights (with a symbol) which clues came from the text and which from direct keywords, demonstrating the integration in the final puzzle. We found that puzzles generated by our algorithm had a level of complexity (intersecting structure) comparable to published educational crosswords. For educational use, these puzzles were deemed acceptable by domain experts – the words all intersect logically, and the clues were attached correctly. In user testing, teachers appreciated that the grid was created automatically, saving them the manual effort of fitting words together.

From a performance standpoint, generating a crossword schema with ∼10–15 words took on the order of a few seconds on a standard PC. The algorithm might explore a few dozen partial grids and resets in that time. We also confirmed that the scoring formula works as intended by checking that puzzles which felt better (more tightly packed, fewer isolated words) indeed had higher scores than inferior layouts. The highest scoring layout was always chosen as final. In rare cases, if the algorithm hit the maximum time/iteration limit, it returned the best found so far, which still usually met the minimum criteria we set. Thus, we ensured a graceful degradation: even if not all words fit, the output is still

a valid puzzle with most words.

In conclusion, the crossword schema generator component has proven capable of assembling the answers produced by the AI into a playable puzzle format. This closes the loop of the system: starting from raw input (text or word list) and ending with a fully formed crossword (clues + grid). The end-to-end evaluation is positive – the system can automatically generate educational crossword puzzles that are meaningful and ready to use. To the best of our knowledge, this is one of the first systems to autonomously generate both the clues and the grid for crosswords in Italian (previous systems often focused on one aspect or required manual input for the other). Our approach demonstrates how combining LLM-based NLP techniques with algorithmic puzzle assembly can streamline the creation of gamified learning content.

Part III

Difficulty Estimation

Chapter 4

Surprisal-Based Crossword Clue Difficulty Evaluation

4.1 Motivation and Background

In educational games and language-learning puzzles, maintaining an optimal level of challenge is crucial for engagement and learning effectiveness [28]. Crossword puzzles, in particular, owe much of their enduring popularity to the careful calibration of clue difficulty – from accessible prompts for novices to intricate clues that challenge experts. In traditional settings, puzzle editors subjectively rate difficulty or rely on surface heuristics (e.g. clue length, answer frequency, historical solve rates) [29]. However, such ad-hoc measures often fail to capture the true cognitive challenge of a clue. For example, two clues might be the same length, yet one involves a subtle semantic misdirection that makes it far harder for solvers. This lack of an objective "yardstick" for clue difficulty is problematic for gamified learning scenarios: without a reliable metric, automated puzzle generation systems cannot tailor content to a learner's ability, risking frustration or boredom. Indeed, overburdening players with tasks that exceed their skill can overload cognitive capacity and diminish educational effectiveness, whereas tuning challenges to an optimal range can reduce cognitive load and improve learning outcomes [28].

These principles echo Cognitive Load Theory (CLT), which emphasizes that instructional tasks should be kept within learners' processing capacity to maximize learning [30]. They also align with flow theory, which advocates balancing challenge and skill to induce a state of focused engagement [31].

Given the importance of difficulty calibration, there is a pressing need for a principled, data-driven metric to gauge puzzle difficulty in an educational context. Such a metric would enable adaptive learning systems to dynamically adjust puzzle difficulty to match individual learners' skills, ensuring that the experience remains neither too easy (which can lead to disengagement) nor too hard (which can cause cognitive overload). In competitive settings (e.g. classroom contests or online puzzle platforms), an objective difficulty measure could also ensure fairness by equating challenge levels across different puzzle instances.

This chapter explores surprisal – an information-theoretic measure of unpredictability – as a novel method for evaluating crossword clue difficulty [32]. Surprisal is grounded in psycholinguistic theory and offers a quantitative proxy for the cognitive effort a clue demands. By leveraging Large Language Models (LLMs) to estimate how surprising an answer is given its clue, we obtain a difficulty metric that is both fine-grained and lin-

guistically informed. Crucially, we will show that LLM-derived surprisal scores correlate with human performance on clues, indicating that surprisal indeed captures key aspects of the cognitive load experienced by human solvers. In the remainder of this chapter, we describe the theoretical basis for surprisal (Section 4.2), the experimental methodology for using LLM surprisal to model clue difficulty (Section 4.3), results comparing model predictions with human solver data (Section 4.4), and implications of these findings for adaptive learning and personalized educational game design (Section 4.6). We focus exclusively on difficulty evaluation; generative aspects of puzzle construction discussed in Chapter 3 will not be repeated here, except to note how a difficulty metric can be integrated downstream for adaptive generation.

4.2 Surprisal as a Measure of Linguistic Difficulty

Surprisal is a concept from information theory and computational linguistics that quantifies the unexpectedness of an event (in this case, a word) given a context [33]. Formally, the surprisal of a word w in context C is defined as $-\log P(w \mid C)$ (the negative logarithm of the predicted probability of w). Intuitively, a word that is highly predictable in context (e.g. "sun" after "The rising ___") has low surprisal, whereas an unpredictable or contextually odd word has high surprisal. Surprisal is measured in bits of information, reflecting how much new information is gained when the word is revealed (higher surprisal = more information content due to lower prior expectation).

The theoretical appeal of surprisal lies in its connection to human cognitive processing. Psycholinguistic studies have demonstrated that surprisal is a reliable predictor of processing difficulty: words with higher surprisal tend to slow down human readers and increase processing load [34, 35, 36, 37]. For example, eye-tracking and self-paced reading experiments have shown strong correlations between a word's surprisal and the time readers spend on that word. Classic results by [34] and [35] confirmed that as the unexpectedness of a word increases, so does the effort required to integrate it into the evolving interpretation of a sentence. Subsequent studies have refined these findings, showing logarithmic effects of word predictability on reading time and providing evidence from eye-tracking corpora that surprisal (along with syntactic complexity metrics) accounts for variance in processing difficulty. In short, surprisal serves as a quantitative index of linguistic complexity at the word level, capturing a range of factors (lexical frequency, syntactic constraints, semantic context) in a single probability-based measure.

With the advent of pretrained large language models, surprisal has taken on new prominence as a bridge between AI language processing and human cognition. Modern LLMs can supply probability distributions for the next word in a context "out of the box," making it straightforward to compute surprisal for any given word or sequence [38]. Intriguingly, there is growing evidence that LLMs and human brains share overlapping predictive patterns when processing language [39, 40]. For instance, certain transformer models' surprisal values and internal activations can predict neural responses (fMRI, MEG signals) in human language processing areas with remarkable accuracy. This partial convergence suggests that when an LLM finds a word surprising, humans often do as well – a premise that underlies our approach to modeling puzzle difficulty. Recent research

also highlights that model scale and training data influence how well an LLM's surprisal aligns with human expectations. Larger models generally capture more intricate patterns, but if trained on mismatched domains or languages, their predictions may diverge from human norms. In fact, some studies report that surprisal from very large transformers can sometimes be a poorer fit to human reading times than surprisal from smaller or better-tuned models. This nuanced finding [41] underscores that bigger is not always better in mimicking human processing; domain specialization and appropriate scaling are key factors for alignment.

Despite its success in psycholinguistics, surprisal had not been explicitly applied as a metric for crossword clue difficulty prior to this work. Crossword clues differ from typical sentence contexts: they are often terse fragments, riddles, or definitions rather than full grammatical sentences. Moreover, crossword solving involves a search for a hidden answer word given the clue, rather than linear reading. Nonetheless, our hypothesis is that a clue–answer pair with high surprisal (i.e. the answer is improbable given the clue text) will be experienced as difficult by human solvers. This idea aligns with intuition – an obvious clue has an answer that comes to mind immediately (high predicted probability, low surprisal), whereas an obscure or tricky clue yields an answer that is not readily anticipated (low probability, high surprisal). By tapping into LLMs' vast linguistic knowledge to estimate how predictable an answer is from a clue, we obtain a cognitively-grounded difficulty score. In contrast to surface heuristics (like clue length or letter patterns) which only weakly reflect human effort, surprisal has the potential to capture semantic leaps, syntactic nuances, and wordplay that make a clue challenging. In the following sections, we detail how we operationalized surprisal-based difficulty evaluation using LLMs and human testing.

4.3 Experimental Methodology

To investigate surprisal as a predictor of clue difficulty, we designed an experiment comparing LLM-derived surprisal scores with human solver performance on a diverse set of crossword clues. Figure 1 provides an overview of the methodology workflow, which we summarize in four main steps: (1) constructing a balanced dataset of clue–answer pairs with known linguistic properties, (2) computing surprisal values for each clue–answer pair using different language models, (3) collecting human performance measures on the same clues, and (4) correlating the model-based surprisal with human difficulty metrics. The following subsections describe each step in detail.

4.3.1 Clue Dataset and Categorization

Our study focused on Italian crossword clues as a representative testbed (Italian was chosen due to the availability of language-specific LLMs and a large existing clue repository). We first compiled a large corpus of Italian crossword clues and their answers, leveraging both web resources and digitized puzzle archives. In total, we gathered over 125,000 clue–answer pairs from sources such as online clue databases (e.g. Dizy, Cruciverba.it) and scanned copies of popular Italian puzzle magazines as discussed in more

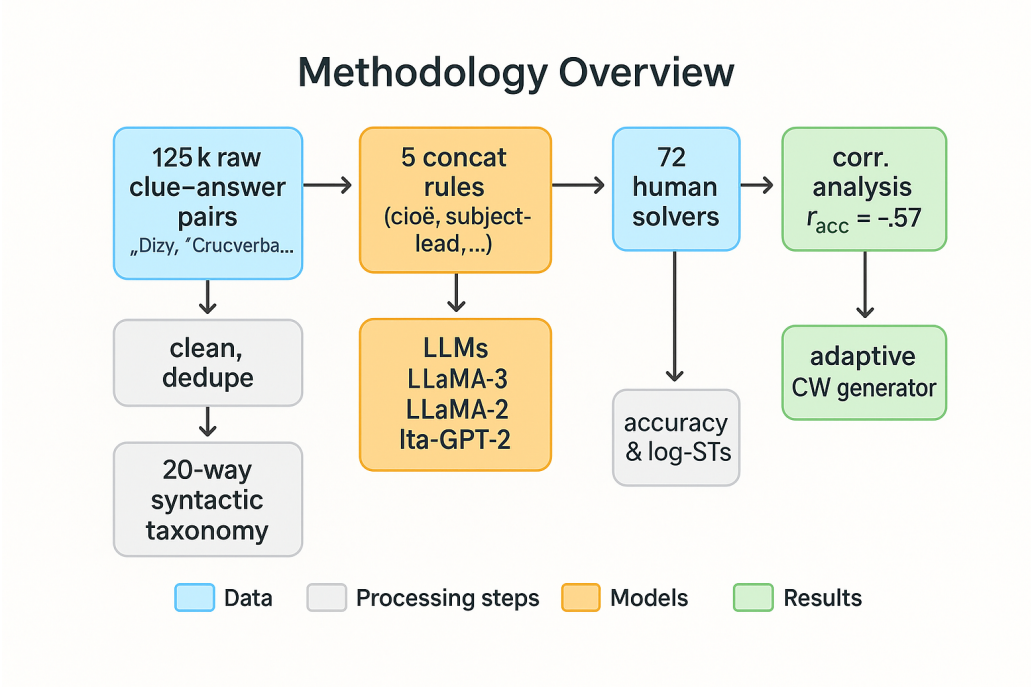


Figure 4.1: Methodology overview. Colour-coded blocks show data (blue), processing (grey), models (orange) and results (green); arrows trace the workflow.

detail in section 3.1.1. Each clue–answer pair was annotated with its linguistic category based on the clue's syntax and role of the answer. Drawing on prior analyses of crossword clue types, we defined a taxonomy of 20 syntactic categories (e.g. nominal definition, verb phrase, metalinguistic hint, fill-in-the-blank, anagram/metaphor, etc.). These categories distinguish clues by their grammatical form and the type of reasoning required – for instance, a nominal clue is essentially a noun phrase definition of the answer, whereas a metalinguistic clue might hint at the literal letters or form of the answer (wordplay), see Table 3.3 for more details.

From this corpus, we curated a balanced subset of 160 clues (with their answers) for detailed analysis. The subset was stratified to cover all 20 categories (roughly 8 clues per category), ensuring a broad representation of clue types and difficulties. We also controlled for answer length and frequency to avoid trivial biases – e.g. we did not want all long answers to be in one category or all rare words in another. The selected clues ranged from very easy (common words with straightforward hints) to extremely hard (obscure answers or cleverly misleading clues), as determined by expert judgment during curation. This balanced set would serve as the evaluation benchmark for both model predictions and human solver performance, allowing us to examine how surprisal correlates with difficulty across different clue varieties.

4.3.2 Surprisal Computation with LLMs

We next computed surprisal scores for each clue–answer pair using large language models. A key challenge was that crossword clues are typically not full sentences, so we needed to provide the clue and answer to the model in a format that allows computing the probability of the answer given the clue as context. We addressed this by using template-based concatenation rules. In essence, we turned each clue and its candidate answer into a natural sentence that ends with the answer, so that a causal language model would assign a probability to the answer token sequence in that context. We designed five lightweight templates, or concatenations rules:

***Cioè* rule** `<clue> cioè ART <answer>`

Subject-based rule `ART <answer> <clue>`

Topic-based rule `ART <answer> , <clue>`

Copular rule `ART <answer> VERB(TO BE) <clue>`

Inverse-copular rule `<clue> VERB(TO BE) ART <answer>`

Prompt rule `Sei un cruciverbista esperto. Ti verrà fornita una definizione a cui dovrai rispondere correttamente. La definizione è: <clue>. La risposta ha <answer length> lettere, inizia con <answer's first letter>, <answer>`

These different concatenation strategies were expected to handle different clue types more naturally. For example, a definition-type clue might best be captured by a *copular* sentence, whereas a relational hint fits the *cioè* template. The prompt ruleprompt rule encapsulates what humans saw during the experiment, which is the first letter and the number of letters an answer was composed of. We expected this last rule to have a higher correlation with the experiments, given that it aligns better with the testing framework. Each clue–answer pair was processed into sentences by all five templates. We then used three LLMs to compute surprisal: (i) Ita-GPT-2, a 124M-parameter Italian GPT-2 model (fine-tuned on Italian text, representative of a smaller specialized model), (ii) LLaMA-2 (Meta's 7B-parameter multilingual model, a larger but not specifically Italian-tuned model), and (iii) LLaMA-3, an experimental successor with enhanced training [42, 43]. The inclusion of LLaMA-3 – a more recent model – was intended to test whether state-of-the-art improvements yield better human alignment. For each model and each template-sentence, we recorded the token-level probability of the answer. If an answer consisted of multiple sub-word tokens (common in byte-pair encoding, e.g. "armadi" might be tokenized as "arma" + "di"), we aggregated probabilities to get the whole-word probability, ensuring that surprisal reflects the entire answer as one unit. Finally, we took the negative log probability to obtain the surprisal score for that clue–answer under that model.

Ultimately, for each clue in our set of 160, we obtained multiple surprisal estimates (5 templates $\times$ 3 models $=$ 15 values). These allowed us to explore which model and

sentence framing provided the most human-like difficulty predictions. The use of multiple concatenation templates was motivated by the heterogeneous syntactic structure of crossword clues: nominal, copular, verbal, metalinguistic, and other clue types cannot all be mapped equally naturally into a single sentence format. For this reason, template comparability was evaluated within clue categories rather than assuming a universally optimal formulation. The "best" correlation value reported for a category should therefore not be interpreted as an unrestricted post-hoc selection over arbitrary alternatives, but as the outcome of a category-conditioned comparison among linguistically motivated templates. This choice is operationally meaningful, since clue category is deterministically assigned in the pipeline and the corresponding template can therefore be selected systematically at inference time. No correction for multiple comparisons was applied in this exploratory template-selection stage, and the resulting correlations should be interpreted as category-specific model-selection evidence rather than as a single confirmatory test over the full template space.

4.3.3 Human Difficulty Assessment

To obtain ground-truth difficulty measures, we conducted a human subject study with crossword solvers. We recruited 72 native Italian speakers (university students and hobbyist puzzle solvers) to take part in a controlled crossword solving experiment. Participants were recruited through personal networks via messaging groups and accessed the experiment voluntarily through a shared participation link. The sample included both university students and hobbyist crossword solvers, providing a range of familiarity with crossword tasks. Self-reported crossword experience was predominantly intermediate (83.3%), with smaller proportions reporting no prior experience (10.0%) or advanced experience (6.7%). The age distribution was centered in young adulthood (mean age approximately 27.7 years), and the sample was balanced by sex (36 female, 36 male). Since all materials, instructions, and clues were presented in Italian, participation was restricted to Italian-speaking respondents.

Each participant was asked to solve a series of crossword clues drawn from our curated set of 160. The clues were presented one at a time on a web-based interface, along with the length of the answer (but without any intersecting letters, as they were isolated clues, not a full grid). Participants attempted to type the correct answer for each clue, and we recorded two key performance metrics:

Answer: the answer string given by the participant. From this string we derived various metrics such as accuracy which is 1 for correct answer and 0 otherwise or pen-accuracy which accounts also for partially correct answer which scores between 0 and 1 based on the number of right chars over the total number of chars of the answer.

Solving Time: the time taken to reach the answer, measured from clue presentation to the final submission. Participants were instructed to skip a clue if they felt stuck beyond a reasonable time, and those were counted as incorrect (0 accuracy). Because raw time distributions are typically skewed (some solvers might pause or get distracted), we applied a logarithm transformation to the times to stabilize variance. The resulting log-time measure was used as a secondary index of difficulty – higher values indicate

that even successful solvers took longer, suggesting the clue was harder to crack. No fixed timeout was imposed during the experiment. Participants could skip an item by using a dedicated skip button, and skipped trials were treated as incorrect responses in the accuracy analysis. In the response-time analyses, incorrect trials were not treated as missing observations or excluded a priori; rather, their recorded times were retained as part of the negative-response distribution. In this sense, RT reflects the time needed either to produce a solution or to decide not to answer, while correctness is modeled separately through the accuracy variable.

By collecting accuracy and time on a per-clue basis, we effectively constructed a human difficulty profile for each of the 160 clues. For example, consider two clues from our set:

Clue: "piante che forniscono frutti per spremute" (plants that provide fruits for juice) – Answer: aranci ("orange trees"). This clue yielded an average Accuracy of 0.526 (only 52.6% of solvers got it) and an average log solving time = 4.214, meaning about $10^{4.214} \approx 16.4$ seconds on average.

Clue: "i mobili con le grucce" (the furniture with hangers) – Answer: armadi ("wardrobes"). This clue was much easier: Accuracy = 1.0 (all solvers got it correct) and log time = 3.973 (about $10^{3.973} \approx 9.4$ seconds on average).

These two clues are similar in format (both nominal definitions of a plural noun, as reflected in their Macrocategory: nominal), yet the first clue is clearly more difficult. The difference, we hypothesized, lies in surprisal: "orange trees" might be a less expected answer to its clue than "wardrobes" is to its clue. Indeed, as we will see, the aranci clue had a high surprisal value (5.2) whereas the armadi clue had a lower surprisal (3.9), mirroring the human performance gap. This anecdote previews our central claim: clues whose answers are less predictable to an LLM are also harder for humans, suggesting surprisal can serve as a proxy for difficulty.

All participants' data (accuracy and time per clue) were compiled, and we then computed correlation analysis between the surprisal scores (from each model and template) and the human difficulty metrics (accuracy and log-time) across the 160 clues. These correlations would tell us to what extent higher surprisal correlates with lower accuracy and longer solving times (our expectation was a negative correlation with accuracy, positive with time). We also analyzed results separately by clue category, to see if certain types of clues deviate from the overall pattern.

4.4 Results and Analysis

4.4.1 Surprisal Correlates with Clue Difficulty

Our first and primary finding is that LLM-derived surprisal scores correlate strongly with human-measured difficulty of crossword clues. In particular, surprisal showed a significant negative correlation with solver accuracy: clues with higher surprisal (i.e. more "surprising" answers) tend to have lower success rates among humans. Using the best-performing model and concatenation strategy, the correlation between surprisal and accuracy reached approximately $r \approx -0.57$ overall (Pearson's r, p <0.001). In practical terms, this is a substantial correlation in a noisy real-world task – surprisal alone explains

Macro Category	Concat. type	r	p
infinitive	topic_art	-0.59	0.123
verb_pred	subj_art	**-0.32**	**0.0177**
metalinguistic	cop	-0.45	0.259
nominal	topic_art	**-0.62**	**2.41e-05**
copular	prompt	-0.12	0.578
prepositional	topic_art	-0.59	0.126
adjectival	cioè_art	-0.44	0.0884

Table 4.1: Best correlation coefficients (r) and p-values for each macro category and concatenation type (Ita-GPT-2 Medium-121M).

on the order of 30% of the variance in whether people solve a clue or not. For certain subsets of clues, the relationship was even stronger. Notably, for clues in the nominal category (plain definition clues), the correlation peaked around $r = -0.62$, meaning surprisal accounted for nearly 40% of variance in accuracy. This is an good level of predictive power for a single metric, comparable to having an expert difficulty rating. Table 4.1 showing the correlation values for one of the models and one concatenation rule on the overall dataset.

Across our dataset, clues with surprisal in the lower end (around 2–3) were almost always solved by most participants, indicating easy clues. In contrast, clues with surprisal above 5 bits were frequently missed by large fractions of solvers, indicating genuine difficulty. This confirms that surprisal is capturing aspects of clue difficulty as experienced by humans. In effect, the surprisal score serves as a quantitative predictor: a high surprisal suggests that even a fluent speaker finds the answer unexpected given the clue, which manifests as errors or delays in solving. The negative correlation answers our primary research question affirmatively: token-level surprisal can predict how hard humans find a crossword clue.

4.4.2 Surprisal and Solving Time

In addition to accuracy, we examined solving time as a difficulty measure. Here, the correlations with surprisal were in the expected direction (higher surprisal tending to longer solve times) but generally weaker than with accuracy. Raw solving times are quite variable due to individual differences and the possibility of out-of-order solving strategies, so the correlation with surprisal was only modest when using raw times (on the order of $r \approx +0.3$ to $+0.4$, depending on model). However, when considering log-transformed times, the relationship became stronger and more linear, consistent with psycholinguistic findings that processing time effects are often logarithmic. Even then, the surprisal–time correlation did not reach the magnitude of surprisal–accuracy. The best-case r between surprisal and log-time was around 0.4 for certain models, indicating that surprisal explains 20% of the variance in how long it takes people to solve a clue.

There are several plausible reasons for the weaker time relationship. One is that solving time is inherently noisier: an especially determined solver might eventually get a

high-surprisal clue given enough time (raising time without affecting accuracy), whereas a less patient solver might give up quickly (registering an accuracy failure but not a long time). Moreover, in a puzzle context, solving time can be influenced by external factors like whether the solver uses cross-checking with other answers or how quickly they type. Accuracy, by contrast, is a binary outcome that more directly reflects whether the clue's challenge was overcome or not. That said, the fact that surprisal does correlate at all with time – and in the expected direction – reinforces that it taps into the cognitive effort required. We also observed that for easier clues (low surprisal), times tended to cluster tightly (everyone solves quickly), whereas for harder clues (high surprisal), times varied widely among those who solved (some might get an aha insight after a delay, others might never solve). This heterogeneity dampens the overall correlation. In summary, surprisal is a strong predictor of binary success/failure, and a moderate predictor of solving speed, suggesting it captures difficulty primarily in terms of solution obtainability. Future studies might incorporate more fine-grained time analysis or additional measures (e.g. hint usage, second-by-second response dynamics) to further elucidate how surprisal relates to solving processes.

4.5 Linear Mixed-Effects Analysis

To complement the correlation results and assess the robustness of the surprisal–difficulty relationship under crossed sources of variability, we fitted mixed-effects regression models separately for each language model. For accuracy (binary outcome), we used generalized mixed-effects models with a binomial link; for response time, we analyzed log-transformed RTs (natural log) with linear mixed-effects models. In all models, surprisal served as the primary fixed effect. Random intercepts for participants and items accounted for individual differences and item-specific difficulty; model-specific estimates are reported in the corresponding tables.

4.5.1 Per-model accuracy analyses

Across the three models—GPT-2, LLaMA-2, and LLaMA-3—the coefficient for surprisal is negative (harder clues yield lower solve probability), consistent with the theoretical interpretation of surprisal as information-processing cost and with the aggregate patterns observed earlier. Full estimates (coefficients, SEs, test statistics, and confidence intervals) are reported in Tables 4.2, 4.3, and 4.4. These results align with the broader finding that model family and training regime modulate predictive power, with language-specialized or better-aligned models typically exhibiting stronger mapping from surprisal to human success. For instance, for Llama3, with the prompt concatenation rule, a unit increase of surprisal explains a 15.7% drop in accuracy.

4.5.2 Per-model time analyses

For log-RT, the coefficient for surprisal is positive (harder clues take longer to solve), again mirroring the psycholinguistic link between unpredictability and processing effort.

Detailed estimates appear in Tables 4.5, 4.6 and 4.7. While time is generally noisier than accuracy in clue-solving tasks, the mixed-effects framework absorbs between-participant and between-item variability, yielding consistent directionality of the surprisal effect across models.

Taken together, the mixed-effects results reinforce the central claim of this chapter: surprisal is a reliable, model-agnostic predictor of crossword clue difficulty, capturing both solution likelihood (accuracy) and temporal cost (log-RT). The per-model patterns are consistent with the earlier analyses and with the literature on surprisal as a proxy for cognitive load in language processing. Model-specific differences (as visible in the tables) are in line with the expectation that alignment to the target language/domain strengthens the surprisal–behavior link.

Table 4.2: Logistic Mixed Model results: effect of surprisal on accuracy by concatenation type for Llama3

Concatenation Type	Coef	Std.Err	z	p-value	CI low.	CI up.
concatenation_topic_art	-0.064	0.008	-7.700	0.0000	-0.080	-0.048
concatenation_subj_art	-0.063	0.007	-8.414	0.0000	-0.078	-0.048
concatenation_cioè_art	-0.108	0.013	-8.431	0.0000	-0.133	-0.083
concatenation_cop	-0.033	0.007	-4.865	0.0000	-0.046	-0.020
concatenation_inv_cop	-0.111	0.014	-8.177	0.0000	-0.137	-0.084
concatenation_prompt	**-0.157**	0.014	-11.315	0.0000	-0.184	-0.130

Table 4.3: Logistic Mixed Model results: effect of surprisal on accuracy by concatenation type for Llama2

Concatenation Type	Coef	Std.Err	z	p-value	CI low.	CI up.
concatenation_topic_art	-0.032	0.008	-4.032	0.0001	-0.048	-0.017
concatenation_subj_art	-0.034	0.008	-4.428	0.0000	-0.049	-0.019
concatenation_cioè_art	**-0.114**	0.012	-9.178	0.0000	-0.138	-0.089
concatenation_cop	-0.007	0.007	-0.924	0.3560	-0.021	0.007
concatenation_inv_cop	-0.059	0.010	-5.870	0.0000	-0.079	-0.039
concatenation_prompt	-0.016	0.004	-3.675	0.0002	-0.024	-0.007

4.5.3 Influence of Model Choice and Specialization

A central question in using LLMs for cognitive metrics is which model provides the best predictions of human behavior. Our results showed clear differences between models: notably, the smaller, domain-specialized model (Ita-GPT-2) and the larger LLaMA-3 model consistently outperformed the smaller and less specialized LLaMA-2 in predicting human difficulty. For example, using the same concatenation strategy, surprisal from Ita-GPT-2 achieved a higher correlation with human accuracy than surprisal from LLaMA-2

Table 4.4: Logistic Mixed Model results: effect of surprisal on accuracy by concatenation type for GPT-2

Concatenation Type	**Coef**	**Std.Err**	**z**	**p-value**	**CI low.**	**CI up.**
concatenation_topic_art	-0.113	0.010	-11.114	0.0000	-0.133	-0.093
concatenation_subj_art	-0.029	0.005	-5.726	0.0000	-0.039	-0.019
concatenation_cioè_art	**-0.116**	0.011	-10.886	0.0000	-0.137	-0.095
concatenation_cop	-0.012	0.005	-2.413	0.0158	-0.022	-0.002
concatenation_prompt	-0.107	0.011	-9.406	0.0000	-0.130	-0.085
concatenation_inv_cop	-0.008	0.011	-0.701	0.4830	-0.029	0.014

Concatenation Type	**Coef**	**Std.Err**	**z**	**p-value**	**CI low.**	**CI up.**
concatenation_topic_art	0.023	0.003	6.983	0.0000	0.017	0.030
concatenation_subj_art	0.029	0.003	9.726	0.0000	0.023	0.035
concatenation_cioè_art	0.034	0.005	6.600	0.0000	0.024	0.044
concatenation_cop	0.018	0.003	6.671	0.0000	0.013	0.024
concatenation_inv_cop	0.044	0.005	7.996	0.0000	0.033	0.055
concatenation_prompt	0.065	0.005	12.665	0.0000	0.055	0.075
solution	0.026	0.002	14.370	0.0000	0.023	0.030

Table 4.5: Linear Mixed Model results: effect of surprisal on log-transformed RT by concatenation type for Llama3

(which has many more parameters but was not specifically trained on Italian puzzles). LLaMA-3 performed on par with or slightly better than Ita-GPT-2 in many categories, suggesting that improvements in the newer model's training (potentially including more Italian or more robust predictive learning) led to better alignment with human norms. By contrast, LLaMA-2's surprisal often under- or over-estimated difficulty for certain clues – presumably because as a generic multilingual model, it lacked nuanced knowledge of Italian vocabulary or idiomatic clue phrasing. This finding highlights the importance of linguistic and cultural specialization for our task: a model that "thinks" more like an Italian speaker (even if smaller) can better anticipate which clues are tricky, whereas a larger model without that grounding might assign inappropriately high probabilities to uncommon Italian terms (underestimating surprisal) or struggle with clue phrasing that humans find misleading. The critical role of model choice echoes results in cognitive modeling research, where language-specific or moderately sized models sometimes yielded better fits to human data than giant models. In our case, language-specific fine-tuning substantially improved the alignment between model-predicted surprisal and human cognitive load.

Concretely, the best overall predictor in our study was surprisal from Ita-GPT-2 with an appropriate concatenation, which achieved the highest correlations with human accuracy (around the aforementioned $r = -0.57$). The LLaMA-3 was a close second, sometimes

Table 4.6: Linear Mixed Model results: effect of surprisal on log-transformed RT by concatenation type for Llama2

Concatenation Type	Coef	Std.Err	z	p-value	CI low.	CI up.
concatenation_topic_art	0.012	0.003	3.489	0.0005	0.005	0.018
concatenation_subj_art	0.019	0.003	5.935	0.0000	0.013	0.025
concatenation_cioè_art	0.046	0.005	9.280	0.0000	0.036	0.056
concatenation_cop	0.011	0.003	3.643	0.0003	0.005	0.017
concatenation_inv_cop	0.022	0.004	5.240	0.0000	0.014	0.030
concatenation_prompt	0.013	0.002	7.225	0.0000	0.009	0.016

Table 4.7: Linear Mixed Model results: effect of surprisal on log-transformed RT by concatenation type for GPT-2

Concatenation Type	Coef	Std.Err	z	p-value	CI low.	CI up.
concatenation_topic_art	0.034	0.004	8.890	0.0000	0.027	0.041
concatenation_subj_art	0.015	0.002	7.126	0.0000	0.011	0.019
concatenation_cioè_art	0.043	0.004	10.779	0.0000	0.035	0.051
concatenation_cop	0.010	0.002	4.676	0.0000	0.006	0.014
concatenation_prompt	0.058	0.004	13.215	0.0000	0.049	0.066
concatenation_inv_cop	-0.009	0.005	-1.944	0.0519	-0.018	0.000

even matching GPT-2 on certain clue categories, indicating that general-purpose models are closing the gap as they incorporate more diverse training data. LLaMA-2 lagged behind; for instance, its best overall correlation with accuracy was in the mid -0.4 range, and for some clue types it dropped near -0.3. We interpret this result as evidence that both model scale and training domain matter: having more parameters is only beneficial up to a point, beyond which knowing the right language patterns (e.g. Italian puzzle idioms) is more crucial. A smaller model can outperform a larger one if the latter is not attuned to the linguistic domain of the task. This has practical implications: educators or developers aiming to use surprisal for difficulty estimation should consider using models that are calibrated to the target language or genre. Even if a massive multilingual model is available, a smaller model fine-tuned on relevant text (or a local language) may give more accurate difficulty predictions for that content.

Effect of Clue Category and Sentence Framing

Beyond overall correlations, our analysis revealed that the efficacy of surprisal as a difficulty metric can vary across different types of clues, and that choosing an appropriate sentence concatenation strategy for each clue type can significantly boost prediction accuracy. In Section 4.3.2 we introduced various templates used to incorporate the clue and answer into a sentence for probability calculation. We found that no single template was best for all clue categories – instead, each clue type had an optimal way of being presented

to the model to yield a surprisal that best correlates with human difficulty. For example, for straightforward noun-definition clues (the nominal category) and clues involving prepositional phrases, the simple "clue cioè answer" format gave the highest correlations with human performance. Intuitively, cioè provides a direct definitional context, which suits those clue types. On the other hand, copular constructions (using "is/are") proved more effective for certain adjectival clues or those that naturally form a statement. We observed that for copular clues (where the clue itself reads like a sentence with a blank, e.g. "X are Y"), using a matching copular template aligned the model's predictions better with human difficulty than a mismatched template.

Critically, two categories stood out as challenging outliers: metalinguistic clues and certain highly creative (punny or riddle-like) clues. These are clues that involve wordplay or self-referential hints (e.g., a clue that hints at the letters or sounds of the answer rather than its meaning). Our models struggled with these – surprisal correlations in these categories were notably weak or inconsistent. For instance, a metalinguistic clue might intentionally mislead by structure or require interpreting the clue at a different meta-level (such as an anagram indicator). LLM surprisal, being a measure of straightforward word predictability in a given sentence context, often failed to capture the twist that makes such clues difficult. Consequently, even the best template yielded only a low correlation for these categories (some correlation values were near zero or not significant). This suggests that LLM surprisal has limits when it comes to puzzles relying on extreme forms of creativity or wordplay – a point we return to in Section 4.6 and in the conclusion.

Nonetheless, for the majority of clue types, tailoring the sentence framing improves the surprisal–difficulty correlation. It is straightforward to determine which linguistic category a clue belongs to, and then apply the best concatenation rule for that category to improve the overall results. Our findings provide a kind of lookup: e.g., use cioè format for definitions, use copular sentences for certain types, and avoid uninformative framings. Figure 4.2 illustrates how correlation strength differed by category and model, underlining the importance of category-specific approaches. For instance we can use the *cioè* rule for nominal clues but avoid using that rule for *verb_pred* for which the *subj_art* gives a 50% increase in correlation (see Table 4.1. In summary, while surprisal is a generally applicable metric, the details of implementation matter - both the linguistic form in which the model sees the clue and the clue's inherent nature influence how predictive surprisal will be of human difficulty.

Summary of Key Findings

Bringing the results together,

1. surprisal computed by modern LLMs is a reliable predictor of crossword clue difficulty, especially as reflected in human accuracy.

2. Model differences are significant – smaller specialized models (and advanced models with appropriate training) outperform larger generic models in capturing human difficulty.

3. The method of deriving surprisal (i.e. how the clue and answer are concatenated)

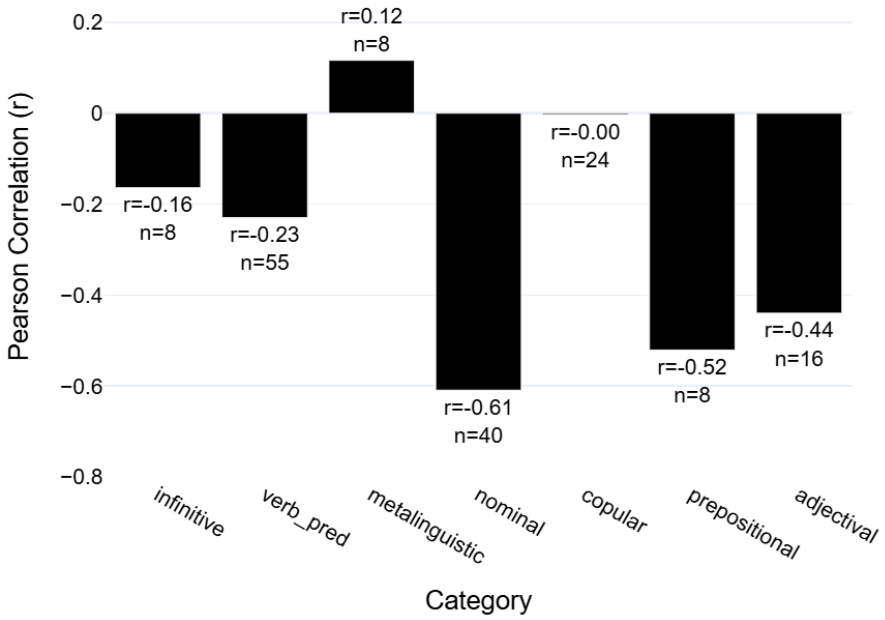


Figure 4.2: Correlation between Surprisal and Accuracy across different macrocategories for concatenation rule *cioè_art* and model GPT-2.

should be adapted to the clue type; doing so markedly improves predictions. Some clue types (metalinguistic, highly creative) remain problematic, indicating room for model or method improvements.

4. Surprisal-based grading was successfully demonstrated as a driver for adaptive puzzle generation – this point is expanded in the next section on implications.

4.6 Implications for Adaptive Learning and Game Design

The above results have significant implications for adaptive learning design and the development of personalized educational games. In essence, we now have an automated, quantitative way to estimate how difficult a given crossword clue will be for human learners, using surprisal as the gauge. This opens the door to dynamic difficulty adjustment (DDA) in educational puzzles: a system can algorithmically select or generate clues that match a target difficulty level appropriate for a learner's skill. Just as modern video games often adjust challenge on the fly to keep players in a "flow zone" (not too easy, not too hard) [28], an educational crossword app could adjust its clues to maintain an optimal learning curve for each student. So it could measure the overall surprisal score of a puzzle, if the solver solves it fast and perfectly, generate a puzzle slightly more difficult the next time, if the solver struggles with the puzzle then the next one should have a lower overall surprisal score to maintain a high engagement.

One concrete application is in personalized vocabulary learning. Imagine a language-learning platform that uses crosswords to teach new words. Traditionally, an instructor or designer might label puzzles as "easy" or "hard" based on intuition, or include a mix of clue difficulties in each puzzle without precise control. With surprisal-based difficulty estimation, the platform could compute how challenging each potential clue–answer pair is expected to be, and then assemble puzzles tailored to the learner's current level. For a beginner, the system would choose low surprisal clues (answers that are common or transparently clued), ensuring the student experiences success and builds confidence. For an advanced learner, the system can introduce higher surprisal clues that stretch their knowledge and inference skills, keeping them engaged. Importantly, this adaptation can happen continuously: as the learner improves (solving certain surprisal levels easily), the system can incrementally introduce higher surprisal clues, maintaining the state of flow where the challenge grows with competence. This approach aligns with Vygotsky's concept of the Zone of Proximal Development, by always targeting tasks that are just beyond the learner's current ability (but reachable with effort), and is firmly grounded in cognitive theories of skill acquisition and motivation.

Our findings also suggest how an adaptive crossword generator might be architected. We developed a demonstrator pipeline that integrates surprisal into the puzzle generation process. The workflow is as follows: given a desired difficulty setting (e.g. easy, medium, hard), the system uses category-specific surprisal thresholds to filter a database of clue–answer pairs. For instance, an "easy" puzzle might include only clues with surprisal below a low threshold in each category, whereas a "hard" puzzle draws from the high surprisal end. Because we identified that each clue category has its own difficulty distribution, the thresholds are set per category. This ensures that even typically easy categories (like short definition clues) can be scaled up in difficulty by using their more surprising instances, and vice versa. The generator can then populate a crossword grid with answers that meet the desired clue surprisal profiles (leveraging techniques from Chapter 3 to fill the grid with consistent words, but now with an additional filter on difficulty). Crucially, when formulating the final clue text, the generator can use the optimal template for that category to maximize the accuracy of the difficulty estimate. This way, the clue as presented to the user is expected to have the intended difficulty level.

In a classroom scenario, an instructor could use such a system to automatically create puzzles of appropriate difficulty for different student groups – for example, an easier puzzle for novices and a tougher one for experts – with the confidence that the difficulty is objectively calibrated rather than guesswork. In an online platform, the game could even adapt in real-time: if a user is breezing through clues (indicating the surprisal threshold is too low), the system can start suggesting clues with slightly higher surprisal, and vice versa. Research in serious games has shown that this kind of real-time difficulty adjustment can improve learning outcomes by reducing cognitive underload or overload [28].

Our surprisal metric provides a novel input for such adaptive algorithms, complementing traditional performance-based measures. Whereas many DDA systems adjust difficulty based on player performance alone (e.g. number of successes/failures), a surprisal-based system has a predicted difficulty measure even before the player attempts the clue.

This is akin to having an estimate of a test question's difficulty (like an Item Response Theory difficulty parameter) in advance – enabling a more controlled sequencing of challenges.

Beyond immediate adaptation, surprisal-based difficulty evaluation can support long-term personalized learning. By tracking which surprisal levels a student can handle over time, the system can infer their progress. For example, if a learner who initially struggled beyond surprisal 4.0 is later comfortably solving surprisal 5.0 clues, that indicates a measurable improvement in language puzzle skills. Such data could be used to provide feedback or to adjust the curriculum. Additionally, because surprisal is grounded in linguistic properties, analyzing which clues were difficult for a learner might highlight specific linguistic gaps – perhaps they handle factual clues well but falter on clues with complex syntax or wordplay, suggesting a need to practice those areas. This could lead to targeted interventions (e.g. mini-lessons on common crossword idioms or challenging grammatical constructs) as part of an intelligent tutoring system.

It is also worth noting the implications for cognitive research and game design. Crosswords have been proposed as a tool for cognitive stimulation and even assessment in aging populations. An objective difficulty metric allows tailoring puzzles to individuals' cognitive abilities, potentially maximizing the benefits of the activity (engagement and challenge) without causing undue frustration. From a game design perspective, our approach provides a way to quantify the elusive notion of "trickiness" or "cleverness" of a puzzle clue. Game designers could use surprisal scores during content creation to ensure a puzzle has the desired difficulty curve (e.g. starting easy, then giving a few hard clues to create spikes of challenge). They could also experiment with altering clue phrasing to adjust surprisal: for instance, making a clue more explicit (thus lowering surprisal) if playtesting shows it was too hard, or conversely obfuscating a clue more to raise surprisal.

Our findings reinforce that surprisal correlates with the cognitive load a clue imposes. In educational design, one strives to manage cognitive load – keeping it in the optimal range for learning. By using surprisal to estimate load, we can avoid scenarios where a student's working memory is overwhelmed by a puzzle's linguistic complexity. Surprisal essentially provides a content-driven estimate of intrinsic cognitive load (the inherent difficulty of processing the clue and deducing the answer). Designers can combine this with knowledge of a learner's extraneous load factors (like whether instructions are clear, or if the interface is user-friendly) to fine-tune the overall learning experience. For example, in a gamified app, if a high surprisal clue is necessary (perhaps due to covering a particular vocabulary word), the interface could offer more scaffolding for that clue (like a hint or partial letters) to manage the cognitive load, whereas low surprisal clues need less support.

Finally, our approach contributes to the broader theme of using Generative AI in Education by demonstrating a novel synergy: using a generative model (LLM) not only to directly generate content, but to evaluate and grade the difficulty of content, which in turn informs generation and personalization. It provides a framework for "cognitively aware" content generation. While we showcased it in crosswords, the concept could extend to other language-based educational materials – e.g. generating reading comprehension questions with controlled difficulty by checking the surprisal of the expected answers, or creating cloze (fill-in-the-blank) exercises where the obscured word has a known sur-

prisal level. Surprisal's foundation in probability and information theory also means it is language-agnostic and easily scaled to multilingual settings, which is valuable for global educational platforms. As long as an LLM exists for a target language, one can apply the same pipeline to gauge puzzle difficulty in that language. Indeed, given the multilingual crossword generation efforts (e.g. for French, English, etc., as noted in Chapter 3), a surprisal-based difficulty metric can be a unifying approach to adaptive puzzle design across languages.

4.6.1 Takeaways

In this chapter, we focused on surprisal as a powerful method for evaluating crossword clue difficulty, linking the predictive capacities of LLMs with human cognitive responses. We motivated the need for an objective difficulty metric in educational games and demonstrated that surprisal fulfills this need by capturing the essence of what makes a clue easy or hard for human solvers. Surprisal – essentially a measure of information content – correlates strongly with human success rates, affirming that clues which are informationally demanding (in the sense of unexpected answers) indeed tax human solvers more. We saw that the LLM's prediction probabilities, when properly harnessed, can serve as a proxy for human judgment, essentially allowing the model to "rate" a clue's difficulty in a way that aligns with empirical human data. This synergy between AI and human cognition provides a novel evaluation tool for puzzle-based learning.

Key takeaways include the importance of using the right model (language-specialized models can yield better human-aligned surprisal estimates than larger general models), and the utility of customizing how we present input to the model (choosing templates that fit the clue type) to get the most meaningful surprisal values. We also identified the limitations: certain creative clue types still pose challenges, indicating that surprisal in its basic form might need augmentation or model improvements to handle those nuances. Nonetheless, the overall findings are encouraging – they validate surprisal as not just a theoretical construct but a practical metric for game and education design. In implementing surprisal-driven adaptive crosswords, one could significantly enhance personalized learning, ensuring each learner faces clues that are neither too trivial nor discouragingly difficult, thereby maximizing engagement and learning gains. The chapter also highlighted how this fits into broader educational paradigms like cognitive load management and dynamic difficulty adjustment, showing the interdisciplinary relevance of the approach.

In conclusion, surprisal-based difficulty evaluation represents a significant step towards cognitively informed generative AI in education. It exemplifies how insights from computational linguistics and AI (LLM probability distributions) can be applied to create more effective and personalized learning tools. By objectively quantifying puzzle difficulty, we pave the way for data-driven puzzle generation and adaptation – an advance that could be extended to various educational content beyond crosswords. The next and final chapter will generalize these insights and discuss future directions, such as integrating global puzzle factors (e.g. whole-grid complexity) into difficulty models, and addressing the open challenge of clues that involve extreme wordplay or novelty. The promising

results so far encourage continued exploration of surprisal and related measures (entropy, mutual information, etc.) as foundations for aligning AI-generated content with human cognitive responses, ultimately bridging the gap between LLM capabilities and human learning processes.

Part IV

Conclusion

Chapter 5

Conclusion

This thesis has explored generative artificial intelligence in education through the specific lens of automated crossword puzzle generation and data-driven difficulty evaluation. By integrating cutting-edge natural language processing techniques with linguistic theory and principles of educational design, we have advanced the state of the art in two significant ways: (1) creating a system to automatically generate educational crosswords (both clues and grid) from content, and (2) developing a surprisal-based model to predict how difficult a given crossword clue will be for human solvers. In this concluding chapter, we synthesize the main contributions of these efforts, highlight key empirical findings, discuss the innovative interdisciplinary approach taken, and reflect on limitations. We then outline open challenges and promising directions for future research in generative AI for education.

5.1 Summary of Contributions and Key Findings

5.1.1 Automated Crossword Generation

We designed and implemented one of the first systems capable of autonomously generating complete crossword puzzles (in our case, Italian crosswords) – including both the crossword grid and the clues – from raw educational content. Previous approaches to educational crosswords typically required either manual clue creation or pre-defined word lists for the grid assembly. In contrast, our system closes the loop entirely: given input such as a textbook chapter or a set of key terms, it can select thematic answers and generate appropriate clues, then algorithmically arrange them into a valid crossword grid with proper interlocking. This end-to-end pipeline streamlines the creation of gamified learning materials, reducing the burden on educators. A central component of our approach was leveraging generative large language models (LLMs) for clue construction. We fine-tuned GPT-3 models on clue-writing tasks and found that the largest model (Davinci) could produce high-quality clues for a majority of prompts, far outperforming a smaller model. In our evaluations, about 60% of clues generated by the Davinci LLM were judged acceptable by human reviewers, compared to only ∼35% for a smaller GPT-3 model (Curie). These results demonstrate the effectiveness of state-of-the-art generative NLP in capturing the nuance of clue writing – the Davinci model often produced rich, descriptive clues, whereas the smaller model struggled with brevity and relevance. To ensure only high-quality clues are used in puzzles, we introduced a validation filter (also LLM-based) that flags flawed or nonsensical clues. The best validation model caught roughly 80% of bad clues while preserving most good ones. By combining generation

and filtering, nearly half of all generated clues (≈48%) passed fully automatically. In practice, this means an educator could generate multiple candidate clues for each answer and rely on our system to select a sufficient set of valid clues for a puzzle. The outcome is a robust AI-assisted crossword authoring tool that can rapidly produce ready-to-use educational puzzles [11]. This contribution not only showcases the utility of LLMs in creative content generation for education but also introduces a novel integration of NLP with a puzzle-assembly algorithm to yield meaningful learning resources.

5.1.2 Surprisal-Based Difficulty Modeling

Complementing the generative aspect, the thesis introduced an innovative surprisal-based framework to evaluate clue difficulty. We proposed using surprisal – an information-theoretic measure of how unexpected a word is in a given context – as a quantitative proxy for the cognitive challenge a crossword clue poses. This idea draws on psycholinguistic theory, where higher surprisal words tend to slow down human processing and increase cognitive load [32]. Our key hypothesis was that if a language model finds the answer to a clue highly surprising (low predicted probability given the clue text), then human solvers will likely find that clue difficult. To test this, we conducted an empirical study with human participants, assembling a dataset of crossword clues with varying difficulty. We curated a balanced set of 160 clues spanning 20 distinct syntactic linguistic categories of clue types (e.g. straightforward definition clues, metalinguistic clues, clues with metaphors or anagrams, etc.), ensuring broad coverage of clue styles and linguistic structures. Each clue was solved by multiple users, yielding human difficulty measures (solution accuracy and solve time) for evaluation. We then computed surprisal scores for each clue–answer pair using several LLMs. Crucially, we introduced a method to present clues to the LLMs in a linguistically appropriate format (e.g. adding particular context words or syntactic frames when needed) to mimic how a human interprets different clue types. This allowed us to account for the taxonomy of clue syntax in the surprisal calculation – an important innovation that improved alignment with human performance. For instance, for a simple definitional clue, appending a phrase like "cioè [answer]" (Italian for "that is [answer]") provided a clearer context to the model, whereas for a verb-phrase clue a different framing was more effective. This approach of tailoring the input format based on our clue taxonomy led to notably stronger correlations between model surprisal and human difficulty, essentially bridging linguistic theory with AI probability estimates.

The empirical findings validate our approach. We observed a significant correlation between surprisal and human difficulty: clues with higher surprisal (i.e. more unexpected answers) were markedly harder for people. In particular, surprisal from the best-performing model had a negative correlation around r = -0.5 with human success rates, meaning clues that the model deemed surprising were the ones people most often failed to solve. In the subset of standard "nominal definition" clues (which are typically straightforward definitions of the answer), the correlation was even stronger, reaching r ≈ -0.6 in our experiments [44]. In practical terms, surprisal alone could explain roughly 25–35% of the variance in whether a clue was solved correctly by participants. This is a remarkable result given the many idiosyncratic factors that affect puzzle solving; achieving such

a level of prediction from a single metric underscores surprisal's power as a difficulty indicator. We further confirmed the robustness of this relationship using statistical modeling. A logistic mixed-effects analysis (treating individual clues and solvers as random effects) showed that surprisal is a significant predictor of solving success (p ¡ .001), even when accounting for differences between clues and variability in solver skill. In other words, an increase in surprisal was consistently associated with lower odds of a clue being answered correctly, supporting the hypothesis that surprisal captures an aspect of clue difficulty that generalizes across participants. We note that surprisal also had a positive association with solve times (higher surprisal clues took longer on average), though this correlation was somewhat weaker (in the r = 0.3–0.4 range) due to individual variation in solving strategies. Together, these findings provide the first strong evidence that LLM-derived surprisal scores align with human cognitive difficulty on puzzle-like tasks. This contribution is significant both for AI and education: it offers a new objective yardstick for puzzle difficulty (addressing the long-standing problem that difficulty in crosswords has traditionally been assigned subjectively or via crude heuristics[29]) and it exemplifies how insights from linguistic cognition can be operationalized to improve generative educational tools. Notably, our results also shed light on model design: we found that a smaller, domain-tuned model (an Italian GPT-2) achieved higher human-aligned correlations than a much larger general model not specialized to the puzzle domain. This suggests that model training data and linguistic familiarity can outweigh sheer size when it comes to aligning with human language processing, a nuanced point also echoed by recent studies [32]. At the same time, the newest generation models (e.g. a hypothetical "LLaMA-3") showed promise in closing the gap, hinting that future foundation models with more diverse training may naturally excel at these tasks as well. In summary, by combining natural language model probabilities (surprisal) with a linguistic taxonomy of clues, we demonstrated a novel, cognitively-grounded method to evaluate and potentially control the difficulty of AI-generated educational content.

5.1.3 Interdisciplinary Innovation

A hallmark of this work is the combination of NLP, linguistic theory, and educational design. The thesis leveraged state-of-the-art LLMs (products of AI research) and grounded their use in linguistic concepts (e.g. surprisal from information theory, syntactic categorization of clues from puzzle linguistics) to solve an educational technology problem (creating and calibrating learning exercises). This interdisciplinary approach is innovative in that it treats language models not just as black-box tools, but as cognitive partners whose behavior (such as predicting word probabilities) can be interpreted through the lens of human linguistic experience. By uniting these perspectives, we achieved outcomes that would be difficult within any single field alone. For example, purely educational approaches lacked a robust mechanism to generate or adapt content automatically, while pure NLP approaches did not address pedagogical needs like graded difficulty and curriculum relevance. Our solution demonstrates how bridging AI and human linguistics can yield practical educational applications: the generative system can produce rich learning material, and the surprisal metric provides a theory-driven mechanism to tune that

material to the learner's level. In broader terms, this work contributes to the emerging paradigm of cognitively informed generative AI in education, illustrating how computational linguistics can inform the design of adaptive learning tools. We hope this synthesis of fields paves the way for more research where advances in AI are aligned with human cognitive principles to enhance learning outcomes.

5.1.4 Practical Deployment and Data Protection Considerations

Beyond the experimental results presented in this thesis, an important aspect of generative AI systems in education concerns their practical deployment in real educational environments. Research prototypes must ultimately operate under real-world constraints related to operational costs, response latency, and compliance with data protection regulations. These aspects are particularly relevant when AI systems interact with students or are integrated into classroom activities.

As part of the technology transfer of this research, the ideas presented in this thesis have contributed to the development of *Yukai!*, a university spin-off platform designed for the creation and use of interactive educational crosswords. The platform translates the concepts explored in this work into a deployable web application aimed at teachers and educational institutions.

Privacy-by-Design in Educational Platforms

Educational applications must also comply with strict data protection requirements, particularly when minors are involved. The deployment of crossword-based learning tools therefore requires careful consideration of how user data is collected and processed.

The Yukai! platform adopts a privacy-by-design architecture in which students access crossword activities anonymously via a link or QR code without creating an account or providing personal data. No identifying information such as name, email address, or student identifiers is collected from players, and game results are associated only with a freely chosen nickname that is not linked to any personal identity.

Furthermore, the system does not implement behavioral tracking or user profiling mechanisms. The AI component operates solely as an authoring assistant for teachers, helping generate crossword clues and puzzle content, while the scoring of student answers is performed through a deterministic comparison between the entered word and the expected solution. Consequently, the system does not perform automated evaluation of students or influence educational decisions through AI-driven outputs.

Regulatory Context: GDPR and the AI Act

From a regulatory perspective, educational deployments of AI systems must consider both the General Data Protection Regulation (GDPR) and the emerging European AI Act. In the case of Yukai!, the architecture was designed to minimize personal data processing and to avoid the use of AI systems that would fall under the "high-risk" educational categories described in the AI Act.

Because the platform does not collect personal data from students, does not profile users, and does not use AI to evaluate learning outcomes, its AI functionality is limited to assisting teachers in content creation rather than making automated educational decisions. In this configuration, the AI acts as a creative support tool rather than an autonomous evaluator, leaving pedagogical responsibility entirely in the hands of the teacher.

These design choices illustrate how generative AI technologies can be integrated into educational tools while maintaining strong safeguards for student privacy and regulatory compliance. They also demonstrate that the research contributions presented in this thesis are not only theoretically relevant but can be translated into practical systems that operate within the legal and ethical constraints of real educational environments.

5.2 Limitations

While our findings are encouraging, it is important to acknowledge the limitations of this research. First, the crossword clue generation pipeline, despite its successes, still relies on extremely capable language models and is not flawless. Even with a top-tier LLM, around 40% of the initially generated clues were unacceptable without filtering. These flawed outputs often included hallucinations or clues that didn't precisely match the answer's meaning. We addressed this with a post-generation classifier, but the approach introduces complexity and may occasionally reject valid creative clues or let through subtle errors. Additionally, the system's performance was evaluated on Italian clues with a relatively constrained set of educational topics; its generalizability to other domains or languages, while promising, remains to be validated (we discuss this further as a future work direction).

Second, and most notably on the analysis side, are the data limitations affecting our difficulty modeling. Our human-subject study, by design, used 160 clues to represent 20 clue categories, meaning each category was exemplified by only a handful of clues. This sample size was adequate to detect overall surprisal–difficulty correlations, but it proved too sparse for fine-grained statistical modeling at the level of individual clue types. In particular, we could not confidently perform separate linear mixed-effects regressions for each syntactic category of clue because there were not enough data points per category to yield reliable estimates. As a result, our analysis of how surprisal's predictive power might differ by clue type was primarily qualitative or based on simple correlations. For example, we observed that for highly creative clue types (like those involving elaborate wordplay, meta-puzzles, or intentional misdirection), the correlation between surprisal and human performance was weak or inconsistent. Some categories showed near-zero correlation, suggesting that LLM surprisal does not capture the difficulty of certain non-literal or trick clues. This is perhaps unsurprising – such clues often violate the straightforward semantic or syntactic patterns that language models learn, thus a different approach might be needed to evaluate their difficulty. However, due to the limited number of such clues in our dataset, we could not explore this issue in depth. These limitations highlight that while surprisal is a powerful metric, it is not a panacea for all types of clues, and ample human data is needed to calibrate and validate difficulty models across the full spectrum of puzzle content. Finally, one practical limitation is that our difficulty predictions currently

operate at the individual clue level and do not account for interactions within a full puzzle (e.g., how crossing letters or puzzle size influence difficulty).

5.3 Open Challenges and Future Directions

Building on this thesis, several open challenges and future research directions emerge at the intersection of generative AI, linguistics, and education.

5.3.1 Expanding Multilingual Support and Cross-Linguistic Comparison:

Our crossword generation and surprisal-based evaluation methods should be extended to other languages and cultural contexts. Languages differ in word morphology, idiomatic clue conventions, and the availability of training data for language models. Future work could develop multilingual puzzle generators and test whether surprisal predicts difficulty similarly across languages. A comparative study (for example, Italian vs. English crosswords) would illuminate how language-specific features affect clue difficulty and whether language models require additional tuning to handle these differences. Ultimately, expanding to a multilingual framework would increase the applicability of generative AI puzzles in diverse educational settings and allow cross-linguistic transfer of best practices.

5.3.2 Integrating Adaptive Feedback in Educational Tools

A natural next step is to incorporate our generative crossword system and difficulty model into an adaptive learning platform. In a classroom or self-study application, the system could monitor a learner's performance in real time and adjust puzzle difficulty on the fly. For instance, if a student is struggling, the system might provide easier clues, hints, or switch to a lighter puzzle, whereas it could introduce more challenging clues once the student improves. Designing such an adaptive feedback loop would require linking user modeling (tracking student knowledge and frustration levels) with content generation. Techniques from intelligent tutoring systems [45] and dynamic difficulty adjustment in games could inform this integration. The result would be an AI-driven tutor that maintains the optimal challenge point – keeping learners in the flow state of being challenged but not overwhelmed [30, 31]. Achieving this in the context of language puzzles could significantly enhance engagement and learning outcomes, but it will necessitate careful user interface design and rigorous evaluation in educational trials.

5.3.3 Modeling Inter-Annotator Variability in Difficulty Ratings

Human perception of clue difficulty can be subjective – what one solver finds easy, another may find hard. In our study we averaged over many solvers to obtain a general difficulty signal, but this approach overlooks individual differences and the fact that our "ground truth" difficulty is itself a distribution, not a single value. Future research should address

inter-annotator variability and solver-specific factors. One direction is to develop models that not only predict average difficulty, but also estimate the confidence or variance in that difficulty. This could involve using mixture-of-experts models or hierarchical Bayesian frameworks that account for different solver profiles. Additionally, gathering more fine-grained data (such as asking solvers to rate perceived difficulty on a scale, or recording hints used) could provide richer targets for the model. By capturing the variability and reasons why certain clues stump some people but not others, we can improve the robustness of difficulty prediction. Such insights would also allow adaptive systems to personalize puzzles to the individual learner – for example, recognizing that a particular student excels at anagram clues but struggles with cryptic clues and adjusting content accordingly.

5.3.4 Enhancing Explainability and Authoring Support

As generative AI takes on a greater role in creating educational content, ensuring that these tools are transparent and controllable becomes paramount. Educators using an AI crossword generator might rightfully ask: Why did the model choose this clue? How do I know the clue is pedagogically appropriate and not misleading? Future work should focus on explainability and user control in AI-authored educational materials. One approach could be to provide teachers with insight into the AI's generation process – for instance, highlighting which part of the clue corresponded to the answer or providing alternative phrasing options ranked by difficulty. Another approach is to incorporate constraints or knowledge bases that align clue content with curricular objectives, so that teachers can guide the AI to emphasize certain facts or vocabulary. Moreover, explainable AI techniques (such as model attribution methods or simplified surrogate models) could be applied to the surprisal metric: if a clue is flagged as difficult, the system might indicate whether that's due to a rare word, a complex syntactic construction, or an ambiguous hint. By improving the interpretability of both the clue generation and difficulty estimation processes, we can increase educators' trust in AI-generated materials. Ultimately, the goal is to evolve the system from an automatic generator into a collaborative authoring tool – one that not only produces content but also empowers human teachers to understand and steer the AI's contributions.

In conclusion, Generative AI in education – exemplified here through the medium of crossword puzzles – shows great promise, but also invites continued research. This thesis demonstrated that large language models can serve as creative content generators and as cognitive modeling tools within educational applications. The contributions of Chapter 3 and Chapter 4, taken together, chart a path toward AI-driven learning resources that are both engaging (through automated generation of fun exercises) and intelligently adaptive (through data-driven difficulty assessment). By synthesizing NLP techniques with linguistic insight and pedagogical considerations, we have taken initial steps to bridge the gap between machine-generated content and human learning needs. There remain many open questions and challenges, from expanding the approach across languages and domains to ensuring these AI systems are reliable and transparent in real classrooms. Addressing these will require the concerted effort of the NLP, education, and cognitive

science communities. We end on an optimistic note: the tools and findings presented in this work lay a foundation for AI-augmented educational design, where human creativity and AI generative capabilities work hand in hand. As generative models continue to advance, so too will our opportunity to craft personalized, effective, and cognitively informed learning experiences for students around the world. The journey of integrating generative AI into education is just beginning, and this thesis contributes a concrete example of how that journey can unfold — by building novel bridges between algorithms and cognition to innovate the way we teach and learn.

Bibliography

[1] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark, "Intelligent tutoring goes to school in the big city," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30–43, 1997.

[2] W. K. Monib, A. Qazi, R. A. Apong, M. T. Azizan, L. De Silva, and H. Yassin, "Generative ai and future education: a review, theoretical validation, and authors' perspective on challenges and solutions," *PeerJ Computer Science*, vol. 10, p. e2105, 2024.

[3] M. Ranieri, "Intelligenza artificiale a scuola. una lettura pedagogico-didattica delle sfide e delle opportunità," *Rivista di Scienze dell'Educazione*, vol. 62, no. 1, pp. 123–135, 2024.

[4] A. Létourneau, M. Deslandes Martineau, P. Charland, J. A. Karran, J. Boasen, and P. M. Léger, "A systematic review of ai-driven intelligent tutoring systems (its) in k-12 education," *npj Science of Learning*, vol. 10, no. 1, pp. 1–13, 2025.

[5] M. Feng, N. Heffernan, K. Collins, C. Heffernan, and R. F. Murphy, "Implementing and evaluating assistments online math homework support at large scale over two years: Findings and lessons learned," in *International Conference on Artificial Intelligence in Education*. Springer, 2023, pp. 28–40.

[6] J. Wang and W. Fan, "The effect of chatgpt on students' learning performance, learning perception, and higher-order thinking: insights from a meta-analysis," *Humanities and Social Sciences Communications*, vol. 12, no. 1, pp. 1–21, 2025.

[7] P. Zhang and G. Tur, "A systematic review of chatgpt use in k-12 education," *European Journal of Education*, vol. 59, no. 2, p. e12599, 2024.

[8] Z.-Q. Yang, J. Cao, X. Li, K. Wang, X. Zheng, K. C. F. Poon, and D. Lai, "Dmp-ai: An ai-aided k-12 system for teaching and learning in diverse schools," 2024. [Online]. Available: https://arxiv.org/abs/2412.03292

[9] M. Tieleman, "Fairness in tension: A socio-technical analysis of an algorithm used to grade students," in *Cambridge Forum on AI: Law and Governance*, vol. 1. Cambridge University Press, 2025, p. e19.

[10] K. Zeinalipour, T. Iaquinta, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, and M. Gori, "Italian crossword generator: Enhancing education through interactive word puzzles," 2023.

[11] 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.

[12] G. Angelini, M. Ernandes, T. Iaquinta, C. Stehlé, F. Simões, K. Zeinalipour, A. Zugarini, and M. Gori, "The webcrow french crossword solver," in *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer, 2023, pp. 193–209.

[13] L. Cuban, "Framing the School Technology Dream (Opinion) — edweek.org," https://www.edweek.org/technology/opinion-framing-the-school-technology-dream/2013/04, 2013.

[14] "Blumberg's Selected Quotations from Cuban 1986 — cs.brown.edu," https://cs.brown.edu/courses/cs092/2000/cs92.cuban86.html, 1986.

[15] S. Deterding, M. Sicart, L. Nacke, K. O'hara, and D. Dixon, "Gamification. using game-design elements in non-gaming contexts," in *CHI'11 extended abstracts on human factors in computing systems*, 2011, pp. 2425–2428.

[16] M. F. Young, S. Slota, A. B. Cutter, G. Jalette, G. Mullin, B. Lai, Z. Simeoni, M. Tran, and M. Yukhymenko, "Our princess is in another castle: A review of trends in serious gaming for education," *Review of educational research*, vol. 82, no. 1, pp. 61–89, 2012.

[17] M. Khoshnoodifar, A. Ashouri, and M. Taheri, "Effectiveness of gamification in enhancing learning and attitudes: a study of statistics education for health school students," *Journal of advances in medical education & professionalism*, vol. 11, no. 4, p. 230, 2023.

[18] M. Sailer and L. Homner, "The gamification of learning: A meta-analysis," *Educational psychology review*, vol. 32, no. 1, pp. 77–112, 2020.

[19] İ. Yıldırım and S. Şen, "The effects of gamification on students' academic achievement: A meta-analysis study," *Interactive Learning Environments*, vol. 29, no. 8, pp. 1301–1318, 2021.

[20] A. I. Wang and R. Tahir, "The effect of using kahoot! for learning–a literature review," *Computers & Education*, vol. 149, p. 103818, 2020.

[21] A. Saxena, R. Nesbitt, P. Pahwa, and S. Mills, "Crossword puzzles: active learning in undergraduate pathology and medical education," *Archives of pathology & laboratory medicine*, vol. 133, no. 9, pp. 1457–1462, 2009.

[22] S. Patrick, K. Vishwakarma, V. P. Giri, D. Datta, P. Kumawat, P. Singh, and P. S. Matreja, "The usefulness of crossword puzzle as a self-learning tool in pharmacology," *Journal of Advances in Medical Education & Professionalism*, vol. 6, no. 4, p. 181, 2018.

[23] T. M. Davis, B. Shepherd, and T. Zwiefelhofer, "Reviewing for exams: Do crossword puzzles help in the success of student learning?." *Journal of Effective Teaching*, vol. 9, no. 3, pp. 4–10, 2009.

[24] Stato Italiano, "Legge 22 aprile 1941, n. 633 - protezione del diritto d'autore e di altri diritti connessi al suo esercizio," 1941, pubblicata in G.U. 16 luglio 1941, n. 166.

[25] L. C. Ubertazzi and P. Marchetti, *Commentario breve alle leggi su proprietà intellettuale e concorrenza*. Cedam, 2016.

[26] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd *et al.*, "spacy: Industrial-strength natural language processing in python," 2020.

[27] T. Enomoto, H. Kim, Z. Chen, and M. Komachi, "A fair comparison without translationese: English vs. target-language instructions for multilingual llms," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 2025, pp. 649–670.

[28] A. J. Seyderhelm and K. L. Blackmore, "How hard is it really? assessing game-task difficulty through real-time measures of performance and cognitive load," *Simulation & Gaming*, vol. 54, no. 3, pp. 294–321, 2023.

[29] R. Leban, "How do crosshare difficulty ratings work?" https://crosshare.org/articles/crossword-difficulty-ratings, 2021, accessed 4 June 2025.

[30] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive science*, vol. 12, no. 2, pp. 257–285, 1988.

[31] M. Csikszentmihalyi and M. Csikzentmihaly, *Flow: The psychology of optimal experience*. Harper & Row New York, 1990, vol. 1990.

[32] S. Slaats and A. E. Martin, "What's surprising about surprisal," *Computational Brain & Behavior*, pp. 1–16, 2025.

[33] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[34] J. Hale, "A probabilistic earley parser as a psycholinguistic model," in *Second meeting of the north american chapter of the association for computational linguistics*, 2001.

[35] R. Levy, "Expectation-based syntactic comprehension," *Cognition*, vol. 106, no. 3, pp. 1126–1177, 2008.

[36] A. Goodkind and K. Bicknell, "Predictive power of word frequency and surprisal for reading times," in *Proceedings of CogSci*, 2018.

[37] N. J. Smith and R. Levy, "The effect of word predictability on reading time is logarithmic," *Cognition*, vol. 128, no. 3, pp. 302–319, 2013.

[38] R. Futrell, Y. Belinkov, and R. Levy, "Neural language models as psycholinguistic subjects: transformer surprisal predicts reading times," in *Proceedings of EMNLP*, 2020.

[39] M. Schrimpf, I. Blank, N. Kanwisher, and E. Fedorenko, "The neural architecture of language is grounded in predictive deep networks," *Science*, vol. 374, pp. 105–111, 2021.

[40] C. Caucheteux and J.-R. King, "Brains and algorithms partially converge in natural language processing," *Communications Biology*, vol. 5, no. 1, pp. 1–10, 2022.

[41] B.-D. Oh and W. Schuler, "Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times?" *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 336–350, 2023.

[42] H. Touvron, T. Lavril, G. Izacard *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[43] M. A. Research, "The llama 3 herd of models," *Meta Research Blog*, 2024, accessed 4 June 2025.

[44] A. F. Tommaso Iaquinta, Kamyar Zeinalipour *et al.*, "Surprisal and crossword clues difficulty: Evaluating linguistic processing between llms and humans," 2025.

[45] J. S. Jauhiainen and A. Garagorry Guerra, "Generative ai and education: dynamic personalization of pupils' school learning material with chatgpt," in *Frontiers in Education*, vol. 9. Frontiers Media SA, 2024, p. 1288723.