AC/DG at Cruciverb-IT: Retrieval-Based Approaches for Italian Crossword Clue Answering

AC/DG at Cruciverb-IT: Retrieval-Based Approaches for Italian Crossword Clue Answering

Ali Yassine[1,†], Claudio Savelli[1,†], Davide Napolitano[1,†], Giuseppe Gallipoli[1,†], Luca Cagliero[1] and Elena Baralis[1]

[1]*Politecnico di Torino, Turin, Italy*

Abstract

We present our submission to Task 1 of the Cruciverb-IT challenge at EVALITA 2026, which focuses on answering clues extracted from Italian crosswords. The task is framed as a constrained question-answering problem, to generate a ranked list of candidate solutions for each clue, given the expected answer length. To address this problem, we explore three complementary approaches: a lexical retrieval model based on BM25, a dense semantic retrieval model that relies on sentence-level embeddings, and a hybrid retrieve-and-rerank architecture that combines the outputs of the two retrieval strategies using a large language model as a judge. Experimental results on the official evaluation benchmark show that dense semantic retrieval achieves the highest overall performance, whereas both lexical and hybrid methods consistently outperform the official task baseline. These findings highlight the effectiveness of embedding-based representations for Italian crossword-clue answering and provide insights into the role of hybrid retrieval and reranking strategies in this task.

Keywords

Large Language Models, Information Retrieval, Crossword Solving, Italian Natural Language Processing

1. Introduction

"*Opening remarks or preliminary section* (12 letters)" – While it may be straightforward to guess that the answer is "*introduction*", this is not always the case, particularly when dealing with clues that involve wordplay, ambiguity, lateral thinking, and cultural references, or, more generally, that require multiple reasoning steps and a higher level of abstraction in language understanding. For this reason, crossword puzzles have recently inspired the creation of new benchmarks in Natural Language Processing [1], as they require a unique synthesis of complex linguistic reasoning, open-domain knowledge retrieval, and strict constraint satisfaction. While earlier approaches mainly implemented rule- and retrieval-based solutions [2, 3], the widespread adoption of (Large) Language Models (LLMs), recently enriched with reasoning capabilities, has brought renewed attention to crossword solving and, more generally, to puzzle solving as challenging benchmarks [4].

The vast majority of research has been concentrated on the English language [5, 6], with few prior works addressing this task in non-English languages, such as Italian [7]. For this reason, we conduct our research to propose and investigate solutions for Italian crossword solving in the context of the Cruciverb-IT task [8] at EVALITA 2026 [9]. The proposed shared task comprises two subtasks: *Task 1*, which consists of answering clues given the clue text and the length of the target answer, without any additional grid constraints; and *Task 2*, which extends the previous task by requiring the solution of complete Italian crossword grids.

Despite their success across many scenarios, Large Language Models still suffer from known limitations in handling wordplay, culture-specific knowledge, or sub-token counting issues [10, 11, 5]. Furthermore, the challenging nature of crossword solving is exacerbated in Romance languages such

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

[†]Corresponding authors. These authors contributed equally.

ali\protect\TU_yassine@polito.it (A. Yassine); claudio.savelli@polito.it (C. Savelli); davide.napolitano@polito.it (D. Napolitano); giuseppe.gallipoli@polito.it (G. Gallipoli); luca.cagliero@polito.it (L. Cagliero); elena.baralis@polito.it (E. Baralis)

0009-0004-3517-8754 (A. Yassine); 0009-0005-1108-1170 (C. Savelli); 0000-0001-9077-4103 (D. Napolitano); 0009-0003-1744-6674 (G. Gallipoli); 0000-0002-7185-5247 (L. Cagliero); 0000-0001-9231-467X (E. Baralis)

as Italian. In these languages, rich morphology, gender–number agreement, and high inflection rates introduce greater entropy into the answer-generation process. While the research community has moved toward tackling "*extreme ambiguity*" in English cryptic crosswords [11, 12], Italian solvers must face a different set of complexities that differ significantly from the English-centric training data dominating current LLM pre-training. As an example, consider the English clue "*Fast* (5 letters)", which would map relatively straightforwardly to "*quick*" or "*rapid*". In contrast, the corresponding Italian clue, for example "*Veloce* (6 lettere)", would lead to a much larger set of possible solutions, such as "*celere*", "*fugace*", "*presto*", "*rapido*", "*pronto*", and "*svelto*", as well as inflectional variants like "*rapida*" and "*pronta*" to account for feminine gender agreement. This highlights that, compared to English, Italian exhibits both greater lexical variety and richer morphology due to gender-number agreement and other morphological processes (e.g., verb conjugation and derivational modification). On the other hand, richer morphology may impose tighter constraints during the grid-filling phase, thereby narrowing the solution space. Considering the previous example, if a crossing entry determines that the final letter is '*E*', the set of potential solutions listed above can be reduced to only the first two candidates.

This paper presents the solutions developed by the AC/DG team[1] to the Cruciverb-IT Task 1, including an LLM-based approach that exploits reasoning capabilities, a more traditional retrieval-based solution, and a combination of the two. The first method leverages lightweight LLM fine-tuning on synthetic reasoning traces to train the model to reason over the provided clue and generate the final answer. In the second approach, a retrieval-based solution is implemented that investigates both lexical- and semantic-level similarity as retrieval strategies. Lastly, we employ an LLM as a generative reranker that, given the outputs of the retrieval-based method, produces a final ordered list of candidate answers. The results confirm the difficulties LLMs face in addressing this task [6, 12], whereas simpler retrieval-based solutions achieve better performance.

2. Background and Related Work

Large Language Models have recently been employed and evaluated across a wide range of tasks that require reasoning. When referring to "*reasoning*", although there is no unique definition of the term [13], the most common benchmarks entail arithmetic, commonsense, or symbolic reasoning. However, in practice, many other reasoning tasks exist, such as causal judgment, temporal understanding, and word sorting [13]. Interestingly, the research community has recently highlighted puzzle solving as a promising testbed for evaluating the applicability of LLMs to complex reasoning tasks [4]. Among rule-based puzzles, deterministic games include crossword solving.

Unlike more narrowly focused tasks such as arithmetic reasoning that do not typically involve broad and diverse semantic knowledge, crossword puzzle solving demands multifaceted skills and knowledge. This is because crossword solving can be viewed as a form of open-domain question answering, requiring the interpretation and resolution of a wide range of clues that draw on substantial linguistic and real-world knowledge. Additionally, the generated answers must also satisfy the constraints of the crossword grid. According to Kulshreshtha et al. [1], crossword clues can be categorized into multiple classes, including factual, historical, abbreviations, word meanings, synonyms and antonyms, and wordplay.

The field of automated crossword solving has advanced significantly, addressing the distinct challenges posed by American-style (i.e., straight) crosswords and British-style (i.e., cryptic) crosswords. For American-style puzzles, which rely heavily on grid constraints and knowledge retrieval (e.g., "*Meryl of movies* (5 letters)", with the answer "*Streep*"), Wallace et al. [10] propose a system that combines neural question answering with belief propagation and local search. Building on this, Saha et al. [5] recently demonstrated that general-purpose LLMs can achieve high performance by leveraging their reasoning capabilities alongside a constraint-based search. Regarding cryptic crosswords, which demand complex linguistic disambiguation and wordplay interpretation (e.g., "*Bird colour* (4 letters)", with the answer "*teal*", which is both a colour and a type of bird), Efrat et al. [12] found that standard fine-tuned models such as T5 struggle to surpass rule-based solvers. Rozner et al. [11] proposed a curriculum learning

[1]From the first letters of the names of the first four authors.

approach, while Sadallah et al. [6] evaluated the zero- and few-shot capabilities of modern LLMs on these tasks, observing that models still lag behind human experts due to difficulties in decomposing wordplay and understanding complex clues.

Due to the abundance of resources in English, puzzle-solving has only recently attracted interest in other languages, such as Italian. For example, Sarti et al. [14] evaluate LLM performance on rebus resolution. With respect to crossword puzzles, the task of crossword generation has been explored for educational purposes [15, 16] by leveraging LLMs to automatically generate pedagogical clues from Wikipedia pages. Italian crossword solving has been addressed both through traditional retrieval-based approaches and by exploiting language model capabilities. Specifically, Angelini et al. [2] propose a framework integrating database, rule-based, dictionary, and web search modules to generate candidate solutions, which are then filtered and merged. In [3], a database is constructed and queried, combining search engine scores with statistical similarity features. Lastly, Ciaccio et al. [7] model the problem as a retrieval task while leveraging siamese and asymmetric dual encoder architectures to capture the underlying relations between crossword clues and their solutions.

In our work, inspired by recent advancements in LLM reasoning for crossword solving and by the lack of evaluation for this task in Italian, we first propose an LLM-based approach. However, prior work has observed that LLMs struggle with linguistic and culturally-relevant knowledge [7]. This limitation can be particularly critical for cryptic crosswords, especially for clues that require ambiguity or polysemy resolution and deeper semantic understanding, such as those involving wordplay, anagrams, nonliteral language, and culture-specific knowledge. These challenges are confirmed by known limitations of LLMs in handling figurative language [17] or non-standard lexical and morphological variation, potentially involving language varieties [18]. For this reason, we also investigate a retrieval-based approach, which may better handle knowledge-grounded clues. Finally, to exploit the strengths of both strategies, we also propose a hybrid approach that combines the two solutions.

3. Data and Task Setup

Challenge Overview. Cruciverb-IT at EVALITA 2026 consists of two tasks. Task 1 focuses on *Italian crossword clue answering*, while Task 2 involves *autonomously solving Italian crossword grids*. In this work, we participate in Task 1.

Task Definition. The first task of the challenge focuses on *Italian crossword clue answering*. The task is formulated as a question-answering problem, where a system must predict one or more candidate solutions for a given crossword clue. Formally, let's consider a set of clues $C = \{c_1, c_2, \dots, c_n\}$. For each clue c_i, the method must produce a ranked list of candidate answers $S = \{s_1, s_2, \dots, s_k\}$, sorted by decreasing likelihood, where the correct solution s_i may appear at any position in the list. To better approximate a realistic crossword-solving scenario and to constrain the search space, each clue is paired with the character length of the target answer. Systems are therefore required to generate candidate answers that respect the specified length constraint. The use of external data sources that explicitly contain crossword clues is prohibited to avoid simple lookup-based systems.

An example input consists of a clue and a target length, such as: "*Lo puoi trovare a tavola* (4 lettere)" (English: "You can find it on the table"). Given this input, a system may generate a ranked list of candidate answers such as {"*pane*" (bread), "*vino*" (wine), "*sale*" (salt), "*olio*" (oil), ...}, ideally including the correct answer "*pane*".

The methods are evaluated using ranking-based metrics. Specifically, Accuracy@1 and Accuracy@10 measure the proportion of clues for which the correct answer appears among the top 1 or top 10 ranked candidate solutions, respectively. Additionally, Mean Reciprocal Rank (MRR) is computed as the average of the reciprocal ranks of the first correct answer across all evaluated clues.

Baseline System. The official baseline treats crossword-clue answering as an information retrieval problem. Given a test clue $c_i \in C_{\text{test}}$, the system computes a similarity score between c_i and each clue in the training set C_{train}. The baseline selects the 10 most similar training clues and returns their associated answers as candidate solutions, ranked by Best Matching 25 (BM25) similarity scores.

Dataset. The dataset used for the task combines the ItaCW crossword dataset [19] with an additional collection of clue–solution pairs gathered from web sources. The dataset contains approximately 410,000 unique clue–answer pairs and covers a wide range of crossword clue types, including wordplay-based, cryptic, named-entity initials, and fill-in-the-blank clues.

4. Methodology

The methodology for Task 1 of the Cruciverb-IT challenge is structured around the comparison of three distinct architectural paradigms for solving crossword clues. Rather than treating each solution in isolation, our framework moves from independent retrieval methods to a unified hybrid system. All systems strictly adhere to the competition rules by utilizing only the provided training data C_{train} and general-purpose linguistic resources, without accessing external crossword-specific databases.

4.1. Reasoning-Based LLM Fine-Tuning (Exploratory)

As an initial approach, we investigated whether Italian crossword solving could be addressed as an end-to-end generative task by explicitly teaching a model how to reason over clues. To this end, we used the Qwen3 8B model[2] to generate structured reasoning traces by conditioning on pairs of crossword clues and their gold answers. The model was prompted to produce step-by-step solution rationales that capture linguistic, semantic, or cultural reasoning patterns leading to the answer, while ensuring that the response adhered to the exact answer length and the principles of progressive discovery.

These generated rationales were then used to fine-tune the same Qwen3 8B model on this synthetic reasoning corpus, to internalize recurring crossword-solving strategies and generalize them to unseen clues.

Preliminary experiments indicated that this approach was poorly aligned with the task's constraints. The model struggled to consistently reproduce the exact lexical forms required by the crossword grid and to recall culturally specific knowledge in a fully generative setting reliably. For this reason, we did not pursue this paradigm further and focused on retrieval-based methods as the core solution. The full Qwen3 8B prompt used to generate reasoning traces is provided in Appendix A.

4.2. Retrieval-Based Approaches

Based on the limitations observed in fully generative reasoning, we adopted retrieval-based methods to exploit observed clue–answer pairs. Our retrieval framework encompasses three components: sparse lexical retrieval, dense semantic retrieval, and a hybrid ensemble that leverages an LLM to rerank candidate answers, as described below.

4.2.1. Solution 1: Lexical Retrieval via BM25

The first solution leverages a sparse retrieval framework based on the BM25 ranking function. This approach is optimized to capture explicit lexical overlaps, serving as an effective mechanism for definitional clues. To improve retrieval efficiency and precision, we implement a **Length-Constrained Indexing** strategy.

- **Bucketed Search Space**: Let $\mathscr{D}$ be the unified training and validation corpus. We partition the corpus into disjoint buckets by answer length L. For a query q, the search is restricted to the specific bucket $\mathscr{B}_L = \{d \in \mathscr{D} : \text{len}(s_d) = L\}$. If $\mathscr{B}_L = \emptyset$, the system defaults to a global search across $\mathscr{D}$.
- **Exact Scoring Logic**: For each document $d \in \mathscr{B}_L$, the system calculates a relevance score by iterating over the query tokens $t \in q$. The score is computed using the inverse document

[2] https://huggingface.co/Qwen/Qwen3-8B

frequency (IDF) and a normalized term frequency component:

$$\text{score}(d, q) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{f(t, d) \cdot (k_1 + 1)}{f(t, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{\text{avgdl}})} \tag{1}$$

where $f(t, d)$ represents the raw count of term t in document d, avgdl is the average length of documents in the collection, and $|d|$ is the token count of the clue. k_1 and b are two hyperparameters that are set to their default values of 1.5 and 0.75, respectively. This implementation ensures that unique and highly descriptive keywords in a clue carry more weight in the final ranking.

4.2.2. Solution 2: Dense Retrieval in Latent Semantic Space

The second approach builds on the observation that handling morphological variation and out-of-vocabulary terms may be relevant in Italian crossword solving, where inflected and less frequent word forms are common [20]. This is done by mapping clues into a continuous latent space $\mathbb{R}^n$. This handles conceptual associations that lack token-level overlap.

- **Transformer-based Encoding**: We employ a Sentence-BERT (SBERT) architecture fine-tuned on Italian corpora[3]. The model functions as an encoder $E : \text{text} \to \mathbb{R}^{768}$, producing a dense vector representation $\mathbf{v} = E(\text{clue})$.
- **Geometric Search and Length Pruning**: Consistent with the lexical approach, the retrieval process is strictly governed by the target length L. For a test query q, we generate an embedding $\mathbf{q}$. The search is restricted to the subset of pre-computed embeddings $\{\mathbf{v}_j\}$ where the corresponding answer s_j satisfies $\text{len}(s_j) = L$.
- **Metric Formulation**: Proximity is quantified via the cosine similarity, measuring the distance between the query and candidate vectors:

$$S_{\cos}(\mathbf{q}, \mathbf{v}_j) = \frac{\mathbf{q} \cdot \mathbf{v}_j}{\|\mathbf{q}\|\|\mathbf{v}_j\|} \tag{2}$$

4.2.3. Solution 3: Generative Reranking via Heuristic Fusion

The third solution implements a hybrid **Retrieve-and-Rerank** strategy. This approach uses Qwen3 8B as a judge to integrate lexical and semantic evidence by synthesizing information from candidate sets generated by BM25 and dense retrieval models. The judge operates in a zero-shot configuration, without task-specific fine-tuning.

1. **Candidate Selection**: For each clue, the top candidate from BM25 lexical retrieval and the top candidate from dense retrieval are selected and presented to the judge. Together, these define a high-recall candidate pool $\mathcal{P}$ for evaluation.
2. **Evaluation and Reranking**: The judge evaluates all candidates and produces a reranked list according to plausibility, placing the most likely solution at the top.
3. **Generative Fallback**: If none of the candidates is deemed satisfactory, the judge may generate a new solution, which is incorporated at the top of the reranked list.
4. **Final Output**: The highest-ranked answer is selected according to the judge's evaluation, with candidates presented in descending order of plausibility.

The full prompt used to guide the model is provided in Appendix B.

[3] https://huggingface.co/nickprock/sentence-bert-base-italian-xxl-uncased

5. Results and Discussion

The evaluation of the proposed systems was conducted using the official metrics for Task 1: Accuracy at 1 (Acc@1), Accuracy at 10 (Acc@10), and Mean Reciprocal Rank (MRR). Table 1 provides a comprehensive comparison of our three implementations against the official Cruciverb-IT baseline, including the relative performance gain (Δ) achieved by our optimal configuration.

Table 1
Comparative performance and relative gain (Δ) of our solutions over the official task baseline.

System Configuration	**Acc@1** (Δ)	**Acc@10** (Δ)	**MRR** (Δ)
Official Task Baseline	0.394 (—)	0.612 (—)	0.457 (—)
Solution 1: Optimized BM25	0.470 (+19.3%)	0.670 (+9.5%)	0.530 (+16.0%)
Solution 2: Dense Embeddings	**0.510 (+29.4%)**	**0.730 (+19.3%)**	**0.570 (+24.7%)**
Solution 3: Hybrid Qwen3 Judge	0.460 (+16.8%)	0.690 (+12.8%)	0.520 (+13.8%)

The results indicate that all three solutions improve upon the official Cruciverb-IT Task 1 baseline. Solution 2, based on dense semantic retrieval, achieves the highest performance across all metrics, with a relative gain of 29.4% in Acc@1. This demonstrates the effectiveness of embedding-based similarity for capturing semantic and conceptual relationships in the dataset. Solution 1, which relies on optimized BM25 lexical retrieval, also shows substantial improvements, particularly when test-set clues overlap lexically or are closely related to those in the training set. This suggests that retrieval-based approaches are particularly useful for datasets with repeated or highly similar clues, thereby exploiting redundancy in the available training data.

Solution 3 introduces a hybrid reranking approach in which a zero-shot Qwen3 8B model serves as a judge for the candidates from Solutions 1 and 2. Its overall accuracy is lower than dense retrieval, which can be explained by a few factors: the judge was not fine-tuned for the task, it has to choose between candidates that may compete with each other, and the dataset includes many ambiguous or nearly identical clues, which can cause the judge to favor plausible but incorrect "distractor" terms. Despite this, the judge still offers conceptual value by reranking candidates based on plausibility and providing a fallback generation when both top candidates are incorrect. This shows that generative models can combine evidence from different sources, even without task-specific training.

Overall, the results highlight the strong effectiveness of retrieval-based approaches on this dataset, especially when lexical or semantic overlap exists between training and test clues. While generative evaluation did not improve accuracy here, it indicates a promising research direction for integrating retrieval- and reasoning-based models in structured, knowledge-grounded tasks.

6. Conclusion and Future Work

In this paper, we present our submission to Task 1 of the Cruciverb-IT challenge at EVALITA 2026, treating Italian crossword-clue answering as a constrained question-answering problem. We evaluated three complementary approaches: a lexical retrieval model based on BM25, a dense semantic retrieval model using sentence-level embeddings, and a hybrid retrieve-and-rerank architecture that combines lexical and dense candidates via a Large Language Model serving as a judge.

Experimental results show that all proposed approaches outperform the official baseline, confirming the effectiveness of retrieval-based formulations for Italian crossword clue answering. Among the evaluated methods, dense semantic retrieval achieves the strongest overall performance across all ranking-based metrics, indicating that embedding-based representations are more effective at capturing the semantic associations required by Italian crossword clues.

Future work may explore a more fine-grained analysis of system behavior across different categories of clues, such as definitional, cryptic, or wordplay-based clues, to better understand the limitations of current retrieval-based methods. Furthermore, since some types of clues require identifying and

following predefined patterns to derive the answer (e.g., clues that implicitly refer to the letters at the beginning, middle, or end of a given word), another research direction could involve employing alternative loss functions beyond the traditional language modeling objective to fine-tune models, with the goal of better constraining their generation and enforcing closer adherence to the structure of the clue and the expected answer [21]. Finally, extending the proposed approaches to the complete crossword grid completion task represents a natural direction for further work, requiring joint reasoning over multiple clues and grid-level constraints.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT 5.2 for proofreading and formatting assistance. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S. Kulshreshtha, O. Kovaleva, N. Shivagunde, A. Rumshisky, Down and across: Introducing crossword-solving as a new NLP benchmark, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2648–2659. URL: https://aclanthology.org/2022.acl-long.189/. doi:10.18653/v1/2022.acl-long.189.

[2] G. Angelini, M. Ernandes, M. Gori, Solving italian crosswords using the web, in: S. Bandini, S. Manzoni (Eds.), AI*IA 2005: Advances in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 393–405.

[3] G. Barlacchi, M. Nicosia, A. Moschitti, A retrieval model for automatic resolution of crossword puzzles in italian language, in: Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & the Fourth International Workshop EVALITA 2014: 9-11 December 2014, Pisa, Pisa University Press, 2014, pp. 33–37.

[4] P. Giadikiaroglou, M. Lymperaiou, G. Filandrianos, G. Stamou, Puzzle solving using reasoning of large language models: A survey, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 11574–11591. URL: https://aclanthology.org/2024.emnlp-main.646/. doi:10.18653/v1/2024.emnlp-main.646.

[5] S. Saha, S. Chakraborty, S. Saha, U. Garain, Language models are crossword solvers, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 2074–2090. URL: https://aclanthology.org/2025.naacl-long.104/. doi:10.18653/v1/2025.naacl-long.104.

[6] A. Sadallah, D. Kotova, E. Kochmar, What makes cryptic crosswords challenging for LLMs?, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 5102–5114. URL: https://aclanthology.org/2025.coling-main.342/.

[7] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell'Orletta, Crossword space: Latent manifold learning for Italian crosswords and beyond, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 245–255. URL: https://aclanthology.org/2025.clicit-1.26/.

[8] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell'Orletta, M. Nissim, Cruciverb-it @ evalita 2026: Overview of the crossword solving in italian task, in: Proceedings of the Ninth Evaluation Campaign of Natural

Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[9] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[10] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg, D. Klein, Automated crossword solving, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3073–3085. URL: https://aclanthology.org/2022.acl-long.219/. doi:10.18653/v1/2022.acl-long.219.

[11] J. Rozner, C. Potts, K. Mahowald, Decrypting cryptic crosswords: Semantically complex wordplay puzzles as a target for nlp, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 11409–11421. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/5f1d3986fae10ed2994d14ecd89892d7-Paper.pdf.

[12] A. Efrat, U. Shaham, D. Kilman, O. Levy, Cryptonite: A cryptic crossword benchmark for extreme ambiguity in language, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 4186–4192. URL: https://aclanthology.org/2021.emnlp-main.344/. doi:10.18653/v1/2021.emnlp-main.344.

[13] J. Huang, K. C.-C. Chang, Towards reasoning in large language models: A survey, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1049–1065. URL: https://aclanthology.org/2023.findings-acl.67/. doi:10.18653/v1/2023.findings-acl.67.

[14] G. Sarti, T. Caselli, M. Nissim, A. Bisazza, Non verbis, sed rebus: Large language models are weak solvers of Italian rebuses, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 888–897. URL: https://aclanthology.org/2024.clicit-1.96/.

[15] K. Zeinalipour, T. Iaquinta, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, M. Gori, Italian crossword generator: Enhancing education through interactive word puzzles, in: F. Boschetti, G. E. Lebani, B. Magnini, N. Novielli (Eds.), Proceedings of the Ninth Italian Conference on Computational Linguistics (CLiC-it 2023), CEUR Workshop Proceedings, Venice, Italy, 2023, pp. 455–464. URL: https://aclanthology.org/2023.clicit-1.55/.

[16] A. Zugarini, K. Zeinalipour, S. S. Kadali, M. Maggini, M. Gori, L. Rigutini, Clue-instruct: Text-based clue generation for educational crossword puzzles, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 3347–3356. URL: https://aclanthology.org/2024.lrec-main.297/.

[17] G. Gallipoli, L. Cagliero, It is not a piece of cake for GPT: Explaining textual entailment recognition in the presence of figurative language, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 9656–9674. URL: https://aclanthology.org/2025.coling-main.646/.

[18] G. Gallipoli, M. L. Quatra, D. R. Cambrin, S. Greco, L. Cagliero, DANTE at geolingit: Dialect-aware multi-granularity pre-training for locating tweets within italy, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), Parma, Italy, September 7th-8th, 2023, volume 3473 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3473/paper14.pdf.

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

[20] C. Savelli, F. Giobergia, Enhancing cross-lingual word embeddings: Aligned subword vectors for out-of-vocabulary terms in fasttext, in: 2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT), IEEE, 2024, pp. 1–6.

[21] D. Rege Cambrin, G. Gallipoli, I. Benedetto, L. Cagliero, P. Garza, Beyond accuracy optimization: Computer vision losses for large language model fine-tuning, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 12060–12079. URL: https://aclanthology.org/2024.findings-emnlp.704/. doi:10.18653/v1/2024.findings-emnlp.704.

A. LLM Prompt for Reasoning Trace Generation

The following prompt was used to generate structured step-by-step reasoning traces from crossword clues and their corresponding answers with Qwen3 8B model. These traces formed the synthetic dataset for our exploratory reasoning-based approach.

```
Sei un assistente AI che genera dati per un dataset di addestramento.
Il tuo compito è generare il ragionamento passo-passo che porta dall'indizio alla
risposta data.
Il ragionamento deve essere procedurale, senza salti da supposizioni a parole
inerenti alla risposta.
Il contesto è quello dei cruciverba in lingua italiana.

Indizio: "{clue}"
Risposta: "{answer}"

REGOLE FONDAMENTALI:
1. NON menzionare la "Risposta Target" ("{answer}") prima dell'ultimo passaggio.
2. Il ragionamento deve sembrare una scoperta progressiva.
3. La risposta finale deve essere una parola di esattamente {answer_length} lettere.

ISTRUZIONI PER IL RAGIONAMENTO:
1. Inizia analizzando l'indizio: "{clue}".
2. Se scomponi le parole dell'indizio, soffermati solo sugli elementi principali
   (non considerare articoli, ecc).
3. Analizza come è formulato l'indizio (es. metafora, abbreviazione, doppio senso, ecc).
4. Spiega il ragionamento, il gioco di parole, o la connessione logica che porta
   dall'indizio alla risposta.
5. Prova a fare ipotesi multiple, scartando quelle che non calzano.
6. Tieni presente che la risposta finale deve essere una parola di esattamente
   {answer_length} lettere.
7. Solo alla fine, come conclusione del tuo pensiero, identifica la risposta corretta.

STRUTTURA OBBLIGATORIA DEL RAGIONAMENTO:
Devi formattare il tuo output seguendo esattamente questi 4 punti:

1. **Analisi dell'Indizio**: Analizza le parole chiave dell'indizio letteralmente e
   figurativamente.
2. **Generazione Ipotesi**: Proponi possibili significati, giochi di parole o contesti
   (es. geografia, oggetti, modi di dire).
3. **Verifica e Filtro**: Scarta le ipotesi che non calzano perfettamente e focalizzati
   su quella più logica.
4. **Conclusione**: Dichiara la risposta finale in modo definitivo.
```

B. LLM Prompt for Generative Reranking

Below is the Italian-language prompt used for the Qwen3 8B model in Solution 3. This prompt defines the evaluation procedure and the expected output format for the candidate solutions.

```
Sei un valutatore imparziale che confronta due risposte di modelli, {top_ans_1}
(dal modello 1) e {top_ans_2} (dal modello 2), e il loro insieme combinato di risposte
candidate: {candidates_all}.
```

Valuta tutte le risposte rispetto alla seguente domanda.

Domanda: {clue}
Lunghezza della risposta prevista: {answer_length} lettere

Procedura di valutazione
1. Valuta individualmente le risposte {top_ans_1} e {top_ans_2}.
- Determina se una delle due risposte risponde correttamente alla domanda.
- Se esattamente una tra {top_ans_1} o {top_ans_2} è corretta, restituisci quella risposta come risposta_finale.
- Se entrambe le risposte sono ugualmente corrette, restituisci quella risposta come risposta_finale.
- Se entrambe le risposte sono errate, verifica se una delle risposte candidate è corretta.
- Se almeno una risposta candidata è corretta, scegli la migliore come risposta_finale.

2. Soltanto se nessuna delle risposte candidate è corretta, genera una nuova risposta corretta:
- La nuova risposta deve rispondere direttamente alla domanda e rispettare la lunghezza prevista ({answer_length} lettere).
- Restituisci questa nuova risposta come risposta_finale.

3. Ordinamento delle risposte candidate
- Ordina tutte le risposte candidate dalla più probabile alla meno probabile di essere corretta.

Formato di output:
risposta_finale: <la risposta selezionata o appena generata>
risposte_candidate:
1. <risposta candidata più probabile>
2. <seconda migliore candidata>
...
N. <risposta candidata meno probabile>