Cruciverb-IT at EVALITA 2026: Overview of the Crossword Solving in Italian Task

Cristiano Ciaccio[1,2], Gabriele Sarti[3,†], Alessio Miaschi[2], Felice Dell'Orletta[2] and Malvina Nissim[4]

[1]*Department of Computer Science, University of Pisa, Italy*
[2]*Institute for Computational Linguistics "A. Zampolli" (CNR-ILC) - ItaliaNLP Lab, Pisa, Italy*
[3]*Khoury College of Computer Sciences, Northeastern University, USA*
[4]*Center for Language and Cognition (CLCG), University of Groningen, The Netherlands*

Abstract

Cruciverb-IT is the first shared task on Italian crossword solving, held at EVALITA 2026. The task comprises two subtasks: (1) answering individual crossword clues given the expected answer length, and (2) autonomously solving complete crossword grids of varying sizes. We release a dataset of approximately 410,000 Italian clue-answer pairs along with automatically generated crossword grids ranging from size 5×5 to 13×13. Five teams participated in the evaluation, submitting a total of 17 system runs. The best-performing system on Subtask 1 achieved 69% accuracy at rank 1 and 0.72 MRR using a retrieval-augmented LLM approach, while the top system on Subtask 2 reached an average character accuracy of 92%, fully solving 34% of grids by means of a fine-tuned encoder-decoder model paired with a constraint-driven depth first search and ranking heuristics. Results show that while modern approaches achieve strong performance on individual clues and smaller grids, solving larger crosswords remains an open problem, with full match performance decreasing rapidly for grids larger than 5x5.

Keywords

NLP, Crossword Solving, Evaluation, Language Models, Italian, Shared Task

1. Introduction and Background

Historically, language games have been an important testbed for creating and studying complex decision-making programs, largely due to a fundamental property: no fixed set of rules will be sufficient to define the overall gameplay. Given the involvement of natural language, in which the interpretation of meaning play a crucial role, judgment is needed not only to produce a solution but even to interpret the rules themselves and, since natural language can be used to describe the full range of human experiences [1], language games are inherently inconsistent with the closed-world assumption [2], according to which anything not explicitly defined is assumed not to hold.

Consequently, prior work has characterized language games such as crossword puzzles as AI-complete problems [3], as solving them requires human-level knowledge and natural language understanding capabilities. For these reasons, language games are emerging as valuable testbeds for evaluating and enhancing the reasoning abilities of Language Models (LMs).

Among language games, crossword puzzles represent a particularly challenging and multifaceted task that requires not only linguistic competence but also cultural knowledge, lateral thinking, and the ability to interpret ambiguous or polysemous clues [4, 5, 6, 7]. As a result, solving crosswords involves complex semantic and pragmatic reasoning, making this setting ideal for testing models' deeper language understanding capabilities beyond surface-level aspects.

Before the advent of modern LMs, most approaches to crossword solving and clue answering relied on retrieval-based methods and shallow lexical and semantic features [3, 8]. For example, Barlacchi et al.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT
[†] Work done at the University of Groningen, The Netherlands.
cristiano.ciaccio@ilc.cnr.it (C. Ciaccio); g.sarti@northeastern.edu (G. Sarti); alessio.miaschi@ilc.cnr.it (A. Miaschi); felice.dellorletta@ilc.cnr.it (F. Dell'Orletta); m.nissim@rug.nl (M. Nissim)
0009-0001-6113-4761 (C. Ciaccio); 0000-0001-8715-2987 (G. Sarti); 0000-0002-0736-5411 (A. Miaschi); 0000-0003-3454-9387 (F. Dell'Orletta); 0000-0001-5289-0971 (M. Nissim)

[9] proposed a system that exploited lexical resources and similarity metrics to match clues to candidate answers in Italian, while the SACRY system [10] incorporated syntactic information and ranking strategies to improve clue-answer matching. However, these systems typically struggle with clues that require deeper interpretative reasoning, such as wordplay, anagrams, or polysemous expressions. Consider, for instance, the clue “Producono con procedimenti lenti”, where "lenti" can mean both “slow” and “lenses” in Italian;[1] a viable answer could be *ottici* (opticians), illustrating the type of ambiguity traditional systems often fail to resolve.

Despite the impressive advancements in Large Language Models (LLMs), their performance on language games such as crosswords remains limited, especially in morphologically rich and less-resourced languages like Italian [11, 12, 13]. Existing LMs and retrieval-based systems still fall short when faced with clues requiring subtle reasoning or cultural grounding.

Building on this line of research, the Cruciverb-IT task organized at EVALITA 2026 [14] represents the first shared task specifically dedicated to crossword solving. The initiative was designed to encourage research in this direction by providing a challenging testbed for developing and evaluating systems on crossword puzzle solving.

2. Definition of the Task

The Cruciverb-IT shared task is organized into two subtasks:

Subtask 1: Clue Answering. The first task consists of answering clues extracted from Italian crosswords. Specifically, the task is formatted as a question-answering problem: participants are presented with a set of clues $C = \{c_1, c_2, \ldots, c_n\}$ and are asked to build a system that for a given clue c_i is able to produce one or multiple candidate solutions $\hat{S} = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_m\}$, possibly containing the correct answer s_i. To simulate a more realistic crossword solving scenario and to further guide the systems towards the correct answer space, each clue c_i is paired with the character length of the target answer s_i. For example: given the clue and the target character length *Sono un fiore di straordinaria bellezza, 4*, the systems should produce a list of one or more candidates, i.e. *[iris, **rosa**, rose, yuzu, fior, ...]* eventually containing the correct answer ***rosa***.

Subtask 2: Grid Solving. The second task consists of autonomously solving Italian crossword grids. The participants are presented with a set of empty crossword grids $G = \{\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_k\}$ where each grid $\mathbf{G}_i$ is paired with a list of clues, each one annotated with the (x, y) coordinates of the square where the corresponding solution starts in the grid and the direction, either down (*verticale*) or across (*orizzontale*). A crossword grid consists of a matrix $\mathbf{G}_i$ of size $\mathbb{R}^{n \times n}$ and each square is either blank or a black square. The developed systems should autonomously fill the grid with appropriate solutions, yielding a fully or partially filled crossword grid that ensures consistent overlap between the characters of crossing words and maximizes the number of correctly placed solutions.

2.1. Task Constraints

Participants were allowed to take part in either subtask or both. For both tasks, we enforced a set of constraints to ensure a fair comparison across systems and prevent training contamination. In particular, the use of external data sources that explicitly contain crossword clues or clue–answer pairs was strictly forbidden. All other types of external data and resources were permitted, including but not limited to dictionaries, encyclopedic resources (e.g., Wikipedia), lexical databases (e.g., WordNet), pre-trained or fine-tuned language models, and distributional representations. Participants were required to explicitly report all external data and resources used in developing their systems. This restriction was introduced to avoid trivial memorization effects and to prevent scenarios in which systems could

[1] The phrase can be translated as "[they] produce with slow procedures", or "[they] produce lenses with procedures", with an unusual but acceptable constituent order in the latter.

exploit large collections of gold crossword data combined with highly engineered search strategies to achieve artificially high performance, potentially comparable to or even exceeding that of professional human solvers [15][2]. By disallowing crossword-specific external resources, we aimed to foster the development of models that genuinely address clue interpretation, lexical retrieval, and reasoning.

3. Dataset

For the proposed task, we relied on both the ItaCW crossword dataset [16] and on a collection of additional clue-solution pairs found on the web. The final dataset, after duplicates are removed, contains approximately 410,000 clue-answer pairs, encompassing various types of puzzles, including wordplay, cryptic clues, named-entity initials, and fill-in-the-blank clues. For the first task, the dataset was divided into training (90%), validation (5%), and test (5%) sets, resulting in approximately 370,000 training examples, and 20,000 examples each for validation and testing. Splits were released as *.csv* files containing three columns: *clue*, *answer* and *answer_length*, with *answer* columns omitted in the test set.

For the second task, we automatically generated crossword grids by employing a constraint-driven, search-based construction algorithm designed to populate a predefined crossword layout with valid words from the list of answers contained in the aforementioned train, validation and test splits, respectively. Specifically, we first generated several empty and square matrices by placing black squares (with various proportions of the total number of squares) randomly, although ensuring symmetry in the layout, and, subsequently, we populated the grid with the aforementioned algorithm. Lastly, we collected the corresponding clues for each word in the grid, therefore obtaining several complete and plausible crosswords. We generated crosswords of different sizes in order to account for various levels of complexity: 5×5, 7×7, 9×9, 11×11 and 13×13 with the number of black squares (as a percentage of the overall available squares) being, respectively, 15%, 16%, 22%, 27% and 27%. Specifically, each empty crossword grid is represented as a matrix, i.e. a list of lists, where each square is either blank (noted as a whitespace ' ') or a black square (noted as a dot '.'). On the other hand, given a grid, the corresponding clues are a list of dictionaries with keys *answer*, *clue*, *x*, *y*, *direction*, *length*, where the coordinates, (*x*, *y*, respectively, rows and columns) expresses where the corresponding solution starts in the grid, the length is the solution number of characters and the direction, noted as *D* or *A*, indicates if the solution should be placed either down (*D*) or across (*A*). As for the first task, we divided the dataset into training (500 grids), validation (50 grids) and test (50 grids) sets. More specifically, the grids were generated following a predefined distribution over grid sizes. The training set consists of 300 grids of size 5×5, 150 of size 7×7, 25 of size 9×9, 15 of size 11×11, and 10 of size 13×13. Both the validation and test sets include 10 grids for each grid size (i.e., 10 grids of size 5×5, 7×7, 9×9, 11×11, and 13×13). Accordingly, participants were asked to return as output the completed crossword grids produced by their systems, represented in the same matrix-based format as the input empty grids, with blank squares filled with the predicted answers[3].

4. Evaluation

The evaluation of the systems was conducted with specific metrics per task, as follows:

- Task-1: **Accuracy@1/10**, that is the accuracy in retrieving the correct solution word given the corresponding clue, considering the top 1 and 10 candidates produced by the system; **Mean Reciprocal Rank (MRR)**, that is the average of the reciprocal ranks of the first relevant item across all clues.
- Task-2: **% of correct characters (CharAcc, CA)**, that is the accuracy in inserting the correct characters in the correct slots; **% of correct words (WordAcc, WA)**, accuracy in inserting the

[2] https://en.wikipedia.org/wiki/Dr.Fill

[3] The dataset can be found at the following HuggingFace repository: https://huggingface.co/datasets/cruciverb-it/evalita2026

correct word in the correct slots; **% of grids solved correctly (FullMatch, FM)**, the accuracy in solving the entire grid. Partially filled grids were evaluated by counting empty squares as errors.

Baselines For clues-answering, our baseline is obtained by approaching the task as an information retrieval problem: given a clue c_1 from the test set $C_{test} = \{c_1, ..., c_n\}$, our system ranks the most similar clues by computing a similarity score between c_1 and each clue in the training set $C_{train} = \{c_1, ..., c_m\}$. After selecting the top ten most similar clues, we extract the corresponding ten answers. The similarity scores between clues are estimated using the BM25 algorithm [17], a well-established ranking function in Information Retrieval. For solving crossword grids, our baseline is computed by combining the aforementioned ranker baseline with an additional module that optimizes for a solution by maximizing satisfied constraints while respecting the grid's hard constraints. Specifically, by treating crossword puzzles as a weighted Max-SMT problem, as partially described in [18], the baseline optimizes for a solution by defining hard constraints (the grid structure) and soft constraints (candidate ranking preferences). Each clue corresponds to a disjunctive group of grid variables constrained to match candidate answers, combined conjunctively across the grid. The formulation uses the Z3 optimizer [19][4] with 10 candidates per clue.

5. Submitted Systems and Participants

Following a call for interest, 5 teams registered for the task and submitted their predictions, for a total of 17 runs (11 for subtask 1 and and 6 for subtask 2). As shown in Table 1, three teams participated only in subtask 1, while two submitted runs for both tasks.

AC/DG [20] AC/DG adopts a retrieval-based framework that combines lexical, semantic, and hybrid reranking strategies. Given a clue and a target length, all retrieval methods operate under strict length constraints, restricting the search space to training instances whose solutions match the target length. The first component is a sparse lexical retriever based on BM25, designed to capture explicit term overlap and definitional clues. In parallel, a dense retrieval model, fine-tuned on Italian and based on a Sentence-BERT encoder, maps clues into a latent semantic space, allowing the system to retrieve morphologically and semantically related candidates even when lexical overlap is limited. The two retrieval streams are combined in a hybrid retrieve-and-rerank strategy, where the top candidates from both BM25 and dense retrieval are passed to an LLM (Qwen3 8B [21]) acting as a zero-shot judge. This model evaluates, reorders, and, when necessary, augments the candidate set by generating a fallback solution. The final output is obtained by selecting the highest-ranked answer from this reranked list, thereby balancing precision from lexical matching with semantic generalization and generative reasoning.

FFT-UniBa [22] FFT-UNIBA adopts a two-stage approach corresponding to the two subtasks. For Task 1, the authors fine-tune an encoder-decoder model based on IT5 [23], pre-trained on Italian texts, introducing length-aware special tokens to explicitly control answer generation. Each input is augmented with a pair of tokens marking the expected solution length, including a length-dependent end-of-sequence token. This design encourages the model to internalize length constraints during training, reducing generation errors caused by length mismatches without relying on post-hoc filtering. For Task 2, each crossword is formulated as a constraint satisfaction problem, where variables correspond to clue slots, and domains consist of ranked candidate answers generated by the Task 1 model. When necessary, the candidate sets are augmented with a small number of dictionary-based words matching the required length and letter pattern. Grid constraints enforce character consistency at word intersections. Crossword solving is performed via a depth-first backtracking search, initialized from single-word seed configurations and guided by model perplexity scores. To ensure tractable inference, the system

[4]We modified an open-source implementation: https://github.com/pncnmnp/Crossword-Solver.

Team	Members	Affiliation	Task	Runs T1	Runs T2
AC/DG	4	Politecnico di Torino	1	3	-
FFT-UniBa	5	Università degli Studi di Bari Aldo Moro	1,2	4	4
MINDS	1	Politecnico di Torino	1	1	-
UNIBA	1	Università degli Studi di Bari Aldo Moro	1,2	2	2
UniTor	2	Reveal Srl; Università degli Studi di Roma Tor Vergata	1	1	-

Table 1
Teams participating in the EVALITA 2026 Cruciverb-IT shared task. For each team, we detail the number of team members, their affiliations, the sub-task(s) they participated in, and the number of submitted runs per subtask (T1 and T2).

enforces explicit limits on node expansions and dynamically adapts candidate cutoffs and search budgets based on grid size.

MINDS [24] MINDS frames crossword clue answering as a masked language modeling problem using an encoder-only architecture. Given a clue and the target word length in characters, the input is constructed by appending a templated sequence in which the answer is represented by a span of `[MASK]` tokens, together with an explicit indication of the expected length. An Italian BERT model is fine-tuned to reconstruct the masked span from the clue context, using standard masked language modeling with cross-entropy loss applied only to the answer positions. At inference time, since the number of subword tokens corresponding to the answer is unknown, the system queries the model with multiple hypothesized mask lengths. For each length, top-K predictions are extracted for each masked position and combined to form candidate answers, which are scored using the geometric mean of token probabilities. Candidates generated across different mask lengths are merged into a single ranked list, keeping the highest score for duplicates. Invalid candidates are pruned based on character length and symbol constraints, and the final output consists of the top-ranked valid answers. This strategy allows an encoder-only model to approximate variable-length generative behavior for crossword solving.

UNIBA [25] UNIBA addresses the crossword-solving task by mimicking a human-like incremental solving strategy based on partial solutions and cross-checking. The approach relies on an encoder–decoder transformer trained to generate answers conditioned not only on the clue, but also on a partially filled solution, when available. Training data are expanded by masking one or more characters in each gold answer, generating all possible partial solutions, and enriching the input with the number of missing characters and the expected answer length. The model is based on IT5-Large and is trained exclusively on the task-provided data, without external lexical resources. During inference, candidate answers are generated dynamically at each step based on the current grid state. The crossword filling task is instead performed using a beam search strategy that iteratively selects and expands the most promising partial grids. To improve candidate selection, the system employs a binary classifier trained on simulated crossword-solving trajectories, which scores candidate answers using grid-level, clue-level, and generation-based features. Candidates are ranked by classifier confidence and decoder scores, allowing the system to balance solution quality and search efficiency. The final submission excludes models using special tokens, which showed inferior validation performance.

UniTor [26] UniTor proposes a retrieval-grounded, LLM-based system that formulates crossword clue answering as a constrained ranking problem. Given a clue and target length, the system combines retrieval-augmented evidence with structured LLM prompting to generate and rank candidate solutions. In a first stage, UniTor retrieves length-compatible clue–solution pairs from a large indexed repository using lexical (i.e. BM25), neural (i.e. BGE-M3 [27]), or hybrid similarity, and injects them into the prompt as lightweight few-shot evidence. In a second stage, candidate generation and ranking are performed within a single LLM call through a structured two-phase prompt. The model is first instructed to explore a diverse set of plausible candidates, prioritizing recall and semantic coverage, and then to filter, normalize, and re-rank them under hard structural constraints such as exact length. This

Subtask 1			
Systems	**Acc@1**	**Acc@10**	**MRR**
UniTor	0.69	0.83	0.72
FFT-UniBa_Constrained1	0.58	0.75	0.63
MINDS	0.59	0.71	0.62
FFT-UniBa_Constrained2	0.57	0.75	0.62
FFT-UniBa_Unconstrained1	0.55	0.72	0.60
FFT-UniBa_Unconstrained2	0.54	0.73	0.60
AC/DG_Embeddings	0.51	0.73	0.57
AC/DG_BM25	0.47	0.67	0.53
AC/DG_Hybrid_Qwen3_Judge	0.46	0.69	0.52
UNIBA RUN2	0.43	0.59	0.47
Baseline	0.40	0.62	0.46
UNIBA RUN1	0.36	0.54	0.41

(a)

Subtask 2			
Systems	**CA**	**WA**	**FM**
FFT-UniBa_c1000_2	0.92	0.85	0.34
FFT-UniBa_c1000NODICT_2	0.92	0.85	0.32
FFT-UniBa_c1000_1	0.93	0.84	0.28
FFT-UniBa_c1000NODICT_1	0.91	0.82	0.22
UNIBA RUN2	0.82	0.67	0.16
UNIBA RUN1	0.82	0.66	0.16
Baseline	0.73	0.58	0.08

(b)

Table 2
Cruciverb-IT leaderboard. Subtask 1 (**a**) ranked by MRR; Subtask 2 (**b**) according to the three metrics: CharAcc (CA), WordAcc (WA) and FullMatch (FM). Subtask 2 results are ranked by FM.

explicit separation between exploration and selection is designed to improve ranking stability and constraint adherence while avoiding multiple LLM interactions. The system outputs a probability-ranked list of candidate answers, where scores represent relative confidence rather than calibrated probabilities. UniTor is evaluated across multiple instruction-tuned LLMs of varying scale to analyze the contributions of model capacity, retrieval grounding, and structured reasoning strategies to crossword-solving performance. The final submitted run is based on the GLM 4.6 model [28].

6. Results and Discussion

In the following, we report and discuss the results achieved by the participants with a further quantitative analysis highlighting relevant influencing factors across both tasks, the potential of an ensemble that leverages all participants predictions and similarities between systems predictions.

6.1. Subtask 1: Clue Answering

Table 2 (**a**) reports the leaderboard of systems participating in Subtask 1. First of all, we observe that most systems outperformed the baseline, indicating that the task cannot be reduced to a simple retrieval problem and that neural and hybrid approaches can capture deeper semantic and inferential patterns than a traditional BM25-based matching strategy. The retrieval-augmented LLM approach adopted by UniTor achieved the best performance, outperforming fine-tuned encoder-decoder models by a significant margin (+11% Acc@1 gain over the second-best system), leveraging a backbone LLM that is, parameter-wise, approximately 6000 times larger than the one used by FFT-UNIBA. This suggests that, while grounding generation with retrieved items provides a valuable contextual guidance for LLMs, the number of parameters remains a strong predictor of final performance, even in crossword clues. Despite this, the Acc@10 gap between UniTor and FFT-UNIBA, MINDS, AC/DG (ranging from 12% to 8%) with respect to the large gap in terms of free parameters, clearly showcases the standalone strength of traditional clues-retrieval approaches and the usage of small neural language models trained with a task-specific fine-tuning strategy, especially in terms of the efficiency/accuracy trade-off. To further support this, we replicate the asymmetrical dual-encoder approach proposed in Ciaccio et al. [13] on the official subtask 1 data, yielding an Acc@10 of 78%, further closing the gap (+5%) with the best system despite, again, a sensibly smaller parameter size[5].

[5]We used the paraphrase-multilingual-mpnet-base-v2 in a dual-encoder setup with $\approx$ 556M free parameters.

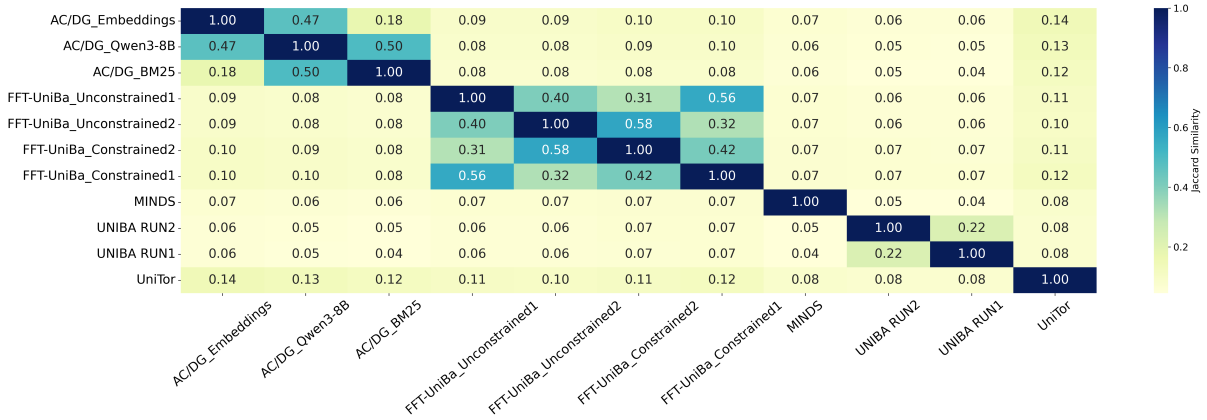

Figure 1: Average pairwise Jaccard similarity (Top10) between all systems sets predictions (a value of 1 indicates a perfect overlap).

System Similarity and Oracle Ensemble. To assess potential similarities between systems, we computed the average pairwise Jaccard similarity across all systems' prediction sets. Specifically, given a test instance t_i and two systems f_o and f_p producing the candidates lists $\hat{S}_o$ and $\hat{S}_p$, the Jaccard similarity between $\hat{S}_o$ and $\hat{S}_p$ is obtained by $J_{t_i} = \frac{|\hat{S}_o \cap \hat{S}_p|}{|\hat{S}_o \cup \hat{S}_p|}$. By comparing all possible pairs of systems and averaging these values across all clues in the test set, as shown in Figure 1, we show that runs from the same team tend to cluster together and exhibit strong similarity, while there is almost no overlap across different teams. These results reveal that participants leveraged different approaches, yielding heterogeneous candidate lists. To further assess the impact of prediction diversity between systems, we built an oracle ensemble by taking the union of all systems' top-k candidate sets (k=1 and k=10) for each clue, and counting a prediction as correct if any system included the gold answer. This approach resulted in upper-bound Acc@1 and Acc@10 of 85% and 94%, respectively. The marked improvements from the best system scores (Acc@1 +16%, Acc@10 +11%) highlight the diversity between systems' predictions and the synergistic potential of combining the approaches proposed by the participants.

Influencing Factors. Several influencing factors were found with respect to the system's predictions[6]. Specifically, accuracy scores tend to clearly decrease as the answer's characters number increases, suggesting that longer answers are harder to predict while shorter ones are easier (see Figure 2, on the right) despite the presence of an initial drop for answers of length 2 (a set that usually includes wordplay, abbreviations, initials, etc.). Interestingly, while all systems follow the same trend, the UNIBA RUN2 is more resilient to this aspect, achieving competitive results for answers longer than 7 characters. Coherently, by inspecting the impact of the answers' frequencies[7], we found a strong positive correlation across all systems (see Figure 2, on the left). Moreover, we also report a negative correlation between Acc@10 and clues lengths – probably denoting longer clues that are harder to interpret – across all systems, ranging from -0.53 to -0.9, with the notable exception of UniTor showing no statistically significant correlation.

System Agreement and Lexical Exposure. We further analyzed the systems' errors by inspecting the degree of agreement across participants. For each clue–answer instance in the test set, we computed the percentage of systems that correctly predicted the gold answer. We then investigated how this agreement relates to lexical exposure by distinguishing whether the gold answer appeared in the training data[8]. Our analysis reveals that lexical exposure has a pronounced effect primarily at higher agreement levels. Instances for which all systems correctly predicted the answer almost exclusively

[6]All reported correlations are Pearson coefficients with $p < 0.05$.
[7]Frequencies are computed on a 2021 Italian Wikipedia dump.
[8]The distribution of agreement levels with respect to training set coverage is reported in Appendix A.

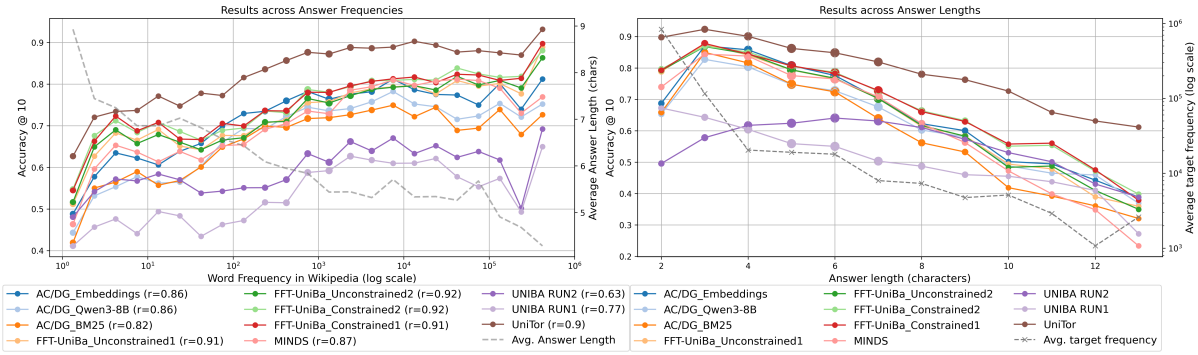

Figure 2: On the left (**a**), the plot shows the Acc@10 of each run across log frequency bins along with their respective Pearson (r) correlations; the gray dashed-line marks the average answer length per bin. On the right (**b**), the plot shows Acc@10 of each run across different answer lengths.

involve words that were observed during training (6236 and 2 instances, respectively), whereas unseen answers are almost absent in this subset. In contrast, for instances that were mistakenly predicted by (almost) all systems, the distribution between seen and unseen words is nearly balanced. These results suggest that, given the heavy training set dependence of the proposed approaches, the presence of an answer word in the training data has a substantial impact on the system's ability to consistently retrieve it at test time, leading to strong agreement across participants. Conversely, in instances characterized by widespread errors, the presence or absence of the answer in the training set does not provide a clear advantage, indicating that lexical exposure alone is insufficient to overcome more challenging clues.

Qualitative Analysis of Shared Errors. Finally, we conducted a qualitative analysis of the errors shared by all systems, focusing on the subset of test instances that were consistently mistaken and therefore represent the most challenging cases. To this end, we applied a TF–IDF representation to the clues and performed unsupervised K-Means clustering to identify recurring patterns in these hard instances. Inspecting the results from a qualitative standpoint[9], we noticed the presence of two clear macro-categories. A first group comprises clues that require access to specific cultural or encyclopedic knowledge (e.g., references to well-known public figures, films, or classical quotations), which are likely harder to solve. A second group consists of inherently ambiguous clues, for which a unique answer may not exist in isolation. Such clues are typically disambiguated only when embedded within a crossword grid (e.g., generic clues such as *"Città francese"* or *"Nome d'uomo"*[10]), a contextual constraint only available in Subtask 2.

6.2. Subtask 2: Grid Solving

Table 2 (**b**) reports the leaderboard of systems participating in Subtask 2. All systems achieved substantially higher results than the baseline thanks to a combination of stronger clue-answering experts and specifically tailored grid-solving algorithms. Both teams employed a similar pipeline, leveraging the systems developed for subtask 1 to generate candidate answers for each clue, then applying a search algorithm to fill the grid while enforcing crossing constraints. Hence, no team exploited the training and validation datasets released for subtask 2.

The FFT-UNIBA runs achieved by far the best results, reaching a character-level accuracy of 92% and correctly solving 34% of the grids in the test set. Given the higher performance obtained in Subtask 1, we hypothesize that the strength of the FFT-UNIBA clue-answering system, which acts as the main semantic bottleneck in the solving pipeline, plays a major role in the observed gap with UNIBA. Moreover, their multiple-seed strategy with high-confidence ranking, backtracking, and a constrained depth-first search

[9] An excerpt of the identified clusters, along with clue examples and the top terms extracted with TF-IDF are reported in Appendix A.

[10] Transl. "French city", "male name".

Systems	5×5			7×7			9×9			11×11			13×13		
	CA	WA	FM	CA	WA	FM	CA	WA	FM	CA	WA	FM	CA	WA	FM
FFT-UniBa_c1000_2	1.00	1.00	1.00	0.94	0.88	0.60	0.92	0.83	0.10	0.91	0.82	0.00	0.85	0.73	0.00
FFT-UniBa_c1000NODICT_2	1.00	0.98	0.90	0.95	0.90	0.60	0.92	0.85	0.10	0.91	0.81	0.00	0.84	0.73	0.00
FFT-UniBa_c1000_1	0.98	0.94	0.70	0.94	0.86	0.50	0.94	0.86	0.20	0.89	0.79	0.00	0.87	0.76	0.00
FFT-UniBa_c1000NODICT_1	0.97	0.90	0.70	0.91	0.82	0.30	0.93	0.84	0.10	0.91	0.81	0.00	0.85	0.74	0.00
UNIBA RUN2	0.89	0.77	0.40	0.85	0.72	0.30	0.83	0.68	0.10	0.80	0.65	0.00	0.70	0.52	0.00
UNIBA RUN1	0.86	0.72	0.40	0.86	0.72	0.30	0.85	0.69	0.10	0.80	0.65	0.00	0.71	0.52	0.00
Baseline	0.85	0.71	0.40	0.68	0.49	0.00	0.74	0.62	0.00	0.66	0.51	0.00	0.73	0.59	0.00

Table 3
Cruciverb-IT Subtask 2 results reported separately for each grid size (5×5, 7×7, 9×9, 11×11, 13×13).

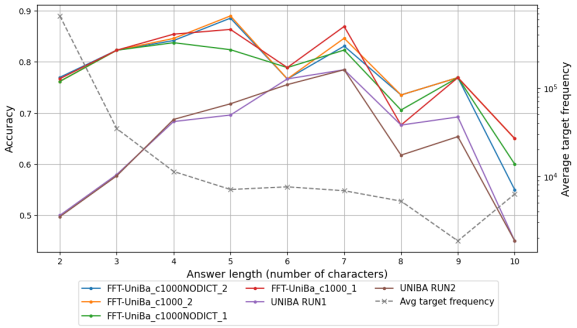

Figure 3: WA accuracies of each run across answer lengths (number of characters); the gray dashed-line marks the average target frequency.

System	Intersection number			
	1	2	3	4
FFT-UniBa_c1000_1	0.73	0.89	0.92	0.95
FFT-UniBa_c1000_2	0.74	0.88	0.92	0.95
FFT-UniBa_c1000NODICT_2	0.71	0.87	0.92	0.95
FFT-UniBa_c1000NODICT_1	0.72	0.87	0.91	0.95
UNIBA RUN2	0.60	0.70	0.81	0.88
UNIBA RUN1	0.60	0.72	0.81	0.89
Baseline	0.53	0.68	0.74	0.75

Table 4: Character accuracy by number of surrounding non-black cells (columns). Systems here are ranked by the overall CA.

approach proved effective in mitigating potentially incorrect early placements and, overall, appears well suited to the combinatorial nature of crossword solving, suggesting that explicit constraint propagation is crucial for grid-level reasoning.

A clear pattern emerges when analyzing performance across grid sizes in Table 3: while systems achieve near-perfect accuracy on 5×5 grids (up to 100% FM for the best system), performance degrades steeply as grid size increases, with the best model dropping from 100% to 60% FM when moving from 5×5 to 7×7 grids. No system achieved a complete solution on 11×11 or 13×13 grids, highlighting the exponential growth in complexity as the number of interdependent constraints increases.

Influencing Factors. For Subtask 2, the influence of lexical and structural factors on system performance appears less pronounced than in Subtask 1. In particular, the relationship between FullMatch accuracy and answer length is weaker. As shown in Figure 3, accuracy generally increases from answers of length 2 up to length 6, followed by a moderate decrease for longer answers, without the marked downward trend observed in Subtask 1. Overall, the drop in performance for longer words is noticeably less severe, and the curves across systems exhibit a smoother behavior. A similar pattern emerges when considering answer frequency. While average target frequency decreases across length bins, accuracy does not show a strong monotonic decline. Instead, performance initially increases and only decreases for the lowest-frequency bins, with a substantially milder effect compared to Subtask 1. A plausible explanation for these trends lies in the specific configuration of Subtask 2, which integrates a crossword grid solver into the prediction pipeline. In this setting, the final predictions do not solely reflect the behavior of the underlying neural models, but rather the interaction between the models and the solver. As a consequence, the solver may act as a filtering and re-ranking component, partially mitigating the impact of both lexical frequency and answer length, and thereby smoothing the correlations observed in Figure 2 (**b**). Moreover, the absence of a sharp performance drop for longer answers can be attributed to the presence of multiple grid constraints: although longer words are generally harder to predict in isolation, their instantiation within a crossword grid provides additional crossing letters and structural cues, which can facilitate their recovery with respect to Subtask 1.

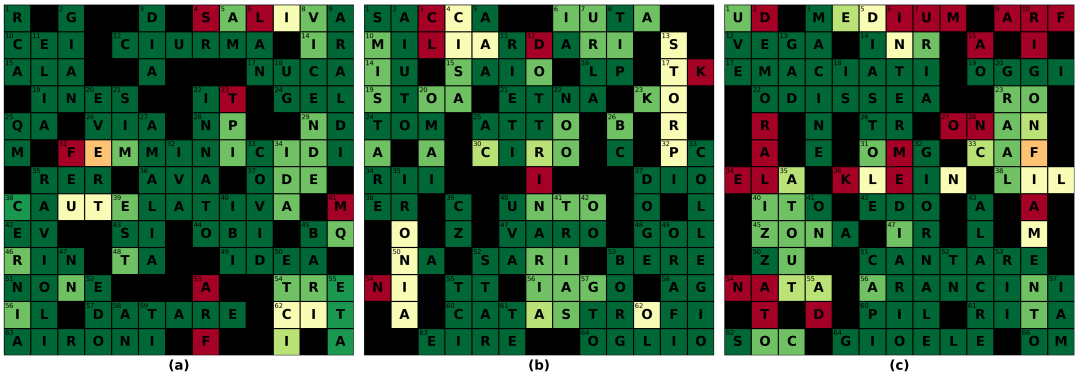

Figure 4: Example grids taken from the test set of subtask 2. The grids (see Appendix B for the corresponding clues) are populated with the correct answers and each cell is colored (green to red) by the fraction of systems that answered correctly (green cell = all systems predicted the correct character in that cell; red cell = no system predicted the correct character for that cell).

The Role of Intersections. Table 4 provides insight into why constraint-based approaches succeed: character accuracy increases monotonically with the number of intersecting words per cell (from 0.73 to 0.95 for FFT-UniBa). Cells with more intersections benefit from additional constraints that help disambiguate among candidate answers, effectively allowing the system to cross-check predictions. This finding aligns with human solving strategies, where solvers often rely on crossing words to confirm or reject candidate answers. Focusing in particular on the FFT-UniBa system, its multiple configurations reveal that larger candidate pools generally improve Full Match scores by increasing the likelihood of including the correct answer in the search space. Dictionary augmentation also provides modest but consistent improvements, particularly for rare words that may not appear in the model's top predictions. We deem the trade-off between the gains from these approaches and the search and computational complexity an interesting topic for future research on efficient automatic crossword solvers.

Figure 4 shows three 13×13 crossword grids taken from the test set and populated by gold solutions with each cell colored by the fraction of systems that correctly predicted the corresponding character. These examples provide qualitative visual evidence supporting the trend reported in Table 4: systems tend to produce incorrect answers more frequently for isolated cells, particularly those corresponding to short entries or located near grid borders. For example, in Figure 4 crossword **(c)**, no system was able to correctly fill numerous isolated cells such as the ones for clues 9, 15, 27, 34, 36, 54, 55 originating, often, regions of high density errors. Similar patterns are notable, also, in grids **(a)** and **(b)**. This common trend highlights the performance advantage of constraint-based approaches when structural redundancy is available for disambiguation, while also underscoring the importance of a robust clue-answering expert for cases with limited or no intersection hints.

6.3. Human and Artificial Solvers

Looking at the core challenges encountered by the models in the Cruciverb-IT tasks, we have wondered whether such challenges are consistent with the difficulties perceived by human solvers. We have also asked ourselves whether the automatically created grids are somewhat in line with grids created by humans, and how they would be received by human creators and solvers. While running a full study with human subjects was not feasible at this time (although it might be considered in future extensions of this work), consultation with Italian crossword expert Stefano Bartezzaghi yielded some interesting considerations. We report them below.

Length At the level of the underlying models, longer solutions are generally harder to predict, with the exception of two-letter words, which often correspond to abbreviations, initials, or wordplay phenomena. This behaviour is consistent with the trends observed in Subtask 1, where predictions are produced in isolation. However, when considering the complete systems used in Subtask 2, i.e. the

combination of neural models and the grid solver, the role of answer length becomes less clear-cut. In this setting, the negative impact of longer words is attenuated, and performance tends to increase up to medium-length answers, followed by only a mild decrease for longer ones. This suggests that the constraints imposed by the crossword grid and the re-ranking performed by the solver partially compensate for the intrinsic difficulty of predicting longer strings in isolation. In particular, longer answers benefit from a higher number of crossings, which provide additional character-level constraints and can facilitate their recovery within the grid. Interestingly, this behaviour brings artificial systems closer to human solving strategies. For human solvers, longer words can in fact be easier to retrieve, due to the larger number of constraints and to the presence of standard bound morphemes, such as *-zione*, *post-*, *sub-*, *-abile*, *-mento*, which further restrict the set of plausible candidates and often make the solution more predictable. Moreover, longer and morphologically transparent words can help fill the grid more efficiently by constraining neighbouring entries.A similar distinction emerges for clue length. At the model level, longer clues tend to be more difficult to handle. From a human perspective, instead, difficulty is primarily determined by the degree of focus and ambiguity of the clue rather than by its length.[11] In fact, longer clues are often more informative and less ambiguous than short and highly compact ones, and can therefore be perceived as easier to solve.

Frequency At the level of the neural models, lexical frequency plays a major role: more frequent answers are generally more likely to be predicted correctly. This behaviour clearly emerges in Subtask 1, where systems operate without grid-level constraints. When considering the full systems employed in Subtask 2, however, the effect of frequency becomes weaker and less monotonic. Although frequent words still tend to be favoured, accuracy does not sharply decrease for low-frequency answers, and the overall trend appears substantially smoother. As in the case of answer length, this attenuation can be attributed to the presence of the grid solver, which integrates the predictions of the models with structural constraints derived from the crossword grid. As a result, the final output reflects the interaction between lexical preferences learned by the models and the combinatorial constraints enforced by the solver, rather than lexical frequency alone. This behaviour partially aligns artificial systems with human solving strategies. For human solvers, common words are in general easier to retrieve, but the specificity and ambiguity of the clue often play a more decisive role than raw lexical frequency. For instance, even a very frequent word such as *"albero"* (transl. "tree") can become difficult to recover when the clue is vague (e.g. "vegetation"), semantically ambiguous (e.g. "Maestro di barca"), or relies on an idiosyncratic contextualization, that is, a clue grounded in a highly specific and unconventional frame rather than in a direct lexical relation (e.g. *"La sequoia lo è del mammut"*[12]). In this respect, the solver-based setting of Subtask 2 reduces the dominance of frequency-driven behaviour observed at the model level, and yields a performance profile that is closer to the human perception of difficulty.

Grid features While for models larger grids are very difficult, with no system being able to solve 11x11 and 13x13 crosswords in Cruciverb-IT, for human solvers the grid size per se does not play a direct role in the complexity of the task. One aspect instead which appears to be consistent across artificial and human solvers is cell isolation: the more neighbors a cell has, and therefore the more constraints, the easier it is to fill that with the correct character.

Do the grids feel "human"? As a general point, we have been wondering to what extent the generated grids resemble good examples of human-crafted crosswords and if they do come across as artificial. Apparently, the gap is there: in general, the grids we have generated would not be particularly

[11] For instance, the solution *Torino* can be clued with increasingly specific definitions, such as *"città italiana"* (transl. "Italian city"), *"capoluogo di regione italiana"* (transl. "capital of an Italian region"), and *"capoluogo del Piemonte"* (transl. "capital of Piedmont"), where more detailed, and thus often longer definitions progressively reduce ambiguity and are typically perceived as easier by human solvers.

[12] Transl. "The sequoia is the mammoth's one". The clue relies on an implicit historical context: sequoias are extremely long-lived trees, and some specimens were already alive when mammoths still existed. The solution is therefore *"albero"*, which can only be retrieved by reconstructing this implicit and unconventional contextual relation.

appealing to a human solver, since they lack what crossword experts would call *rhythmic breadth*. This is due to the fact that the generated grids do not offer elaborate crossings and do not form large *squares*, except in very limited extensions.

7. Conclusion

We presented Cruciverb-IT, the first shared task on Italian crossword solving, held at EVALITA 2026. The task attracted five participating teams who submitted a total of 17 system runs across two subtasks: clue answering and full grid solving. We released a dataset of approximately 410,000 Italian clue-answer pairs and 600 automatically generated crosswords of varying sizes, providing a new benchmark for evaluating language understanding and reasoning capabilities in Italian.

Our evaluation reveals that modern NLP approaches achieve promising results in individual-clue answering, with the best system reaching 69% accuracy at rank 1 via retrieval-augmented LLM prompting. For grid solving, constraint-satisfaction methods combined with fine-tuned language models are most effective, achieving up to 92% character accuracy and solving 34% of grids completely. However, a clear scalability challenge emerges: while systems reliably solve smaller grids (5×5), the steep performance decrease in larger grids indicates that crossword solving at realistic scales remains an open problem.

In addition, the qualitative analysis reported in Section 6.3, based on a consultation with Italian crossword expert Stefano Bartezzaghi, highlights a systematic gap between human solving strategies and the behavior of current neural models, as well as a more nuanced picture when considering full crossword-solving systems. In particular, our observations show that factors such as morphological predictability, semantic focusing, ambiguity, and idiosyncratic contextualization of clues play a central role for human solvers. At the same time, model-level predictions are largely driven by surface-level properties such as answer length and lexical frequency, whereas system-level configurations that integrate a grid solver partially mitigate these effects by exploiting structural constraints induced by the crossword grid. This comparison suggests that future crossword-solving systems should explicitly account for higher-level linguistic and pragmatic properties of clues, and not only for grid-level constraints, in order to better approximate human strategies.

The findings suggest several directions for future research. First, hybrid architectures that combine the semantic understanding of Large Language Models with explicit constraint reasoning may be required to effectively improve grid-level performance, beyond solving individual clues. Second, the strong correlation between intersection density and accuracy suggests that iterative refinement strategies could be effective for progressively constraining the solution space. Finally, the resource and evaluation framework introduced in this task can serve as a blueprint for exploring crossword solving in other languages and for investigating the broader question of how language models handle linguistic puzzles at the intersection of cultural knowledge and logical reasoning.

Acknowledgments

We would like to thank Stefano Bartezzaghi for carefully analyzing our results and observations and for providing valuable insights that greatly helped us better interpret the linguistic and pragmatic aspects of the task. The authors acknowledge the support of the PNRR MUR project PE0000013-FAIR. Partial support was also received by the project "Understanding and Enhancing Preference Alignment in Large Language Models Through Controlled Text Generation" (IsCc8_ALIGNLLM), funded by CINECA under the ISCRA initiative, for the availability of HPC resources and support.

Declaration on Generative AI

During the preparation of this work, the author used GPT-5 and Grammarly to conduct grammar and spelling checking. The author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] S. C. Shapiro, Artificial intelligence, in: S. C. Shapiro (Ed.), Encyclopedia of Artificial Intelligence, 2nd ed., John Wiley & Sons, Inc., New York, 1992, pp. 54–57.

[2] M. L. Littman, Review: Computer language games, in: T. Marsland, I. Frank (Eds.), Computers and Games, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 396–404.

[3] M. Ernandes, G. Angelini, M. Gori, Webcrow: A web-based system for crossword solving, in: AAAI Conference on Artificial Intelligence, 2005. URL: https://link.springer.com/chapter/10.1007/11590323_37.

[4] 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.

[5] 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.

[6] 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/.

[7] 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/.

[8] G. Angelini, M. Ernandes, M. Gori, Solving italian crosswords using the web, in: International Conference of the Italian Association for Artificial Intelligence, 2005. URL: https://link.springer.com/chapter/10.1007/11558590_40.

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

[10] A. Moschitti, M. Nicosia, G. Barlacchi, SACRY: Syntax-based automatic crossword puzzle resolution sYstem, in: H.-H. Chen, K. Markert (Eds.), Proceedings of ACL-IJCNLP 2015 System Demonstrations, Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Beijing, China, 2015, pp. 79–84. URL: https://aclanthology.org/P15-4014/. doi:10.3115/v1/P15-4014.

[11] 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 10th 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/.

[12] G. Sarti, T. Caselli, A. Bisazza, M. Nissim, EurekaRebus - verbalized rebus solving with LLMs: A CALAMITA challenge, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 1202–1208. URL: https://aclanthology.org/2024.clicit-1.132/.

[13] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell'Orletta, Crossword space: Latent manifold learning for italian crosswords and beyond, in: Proceedings of the 11th Italian Conference on Computational

Linguistics (CLiC-it 2025), 2025.
[14] 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.
[15] M. L. Ginsberg, Dr. fill: Crosswords and an implemented solver for singly weighted csps, Journal of Artificial Intelligence Research 42 (2011) 851–886.
[16] K. Zeinalipour, T. Iaquinta, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, M. Gori, Italian crossword generator: Enhancing education through interactive word puzzles, in: Proceedings of the 9th Italian Conference on Computational Linguistics (CLiC-it 2023), 2023. URL: https://ceur-ws.org/Vol-3596.
[17] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, volume 4, Now Publishers Inc, 2009.
[18] 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.
[19] L. De Moura, N. Bjørner, Z3: An efficient smt solver, in: International conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2008, pp. 337–340.
[20] A. Yassine, C. Savelli, D. Napolitano, G. Gallipoli, L. Cagliero, E. Baralis, Ac/dg at cruciverb-it: Retrieval-based approaches for italian crossword clue answering, 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.
[21] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., Qwen3 technical report, arXiv preprint arXiv:2505.09388 (2025).
[22] A. Porcelli, F. Di Gravina, E. Fontana, M. Curri, F. D. Di Gregorio, Fft-uniba at cruciverb-it: Special length tokens and csp for italian crossword solving, 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.
[23] G. Sarti, M. Nissim, IT5: Text-to-text pretraining for Italian language understanding and generation, 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. 9422–9433. URL: https://aclanthology.org/2024.lrec-main.823/.
[24] F. Giobergia, Minds at cruciverb-it: Solving italian crossword clues with masked language models and candidate pooling, 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.
[25] P. Basile, Uniba at cruciverb-it: Solving crosswords with encoder-decoder models and beam search, 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.
[26] A. Shcherbakov, D. Croce, R. Basili, Unitor at cruciverb-it: Retrieval-augmented two-step reasoning for italian crossword clue answering, 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.
[27] M.-L. M.-F. Multi-Granularity, M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.
[28] A. Zeng, X. Lv, Q. Zheng, Z. Hou, B. Chen, C. Xie, C. Wang, D. Yin, H. Zeng, J. Zhang, et al., Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, arXiv preprint arXiv:2508.06471 (2025).

A. Influencing Factors

Figure 5 shows the distribution of test instances across different system agreement levels, split according to whether the answer word appears in the training set. Table 5, instead, reports some examples of the clues and the corresponding top TF-IDF terms extracted by applying K-Means clustering to the set of instances incorrectly predicted by all the systems.

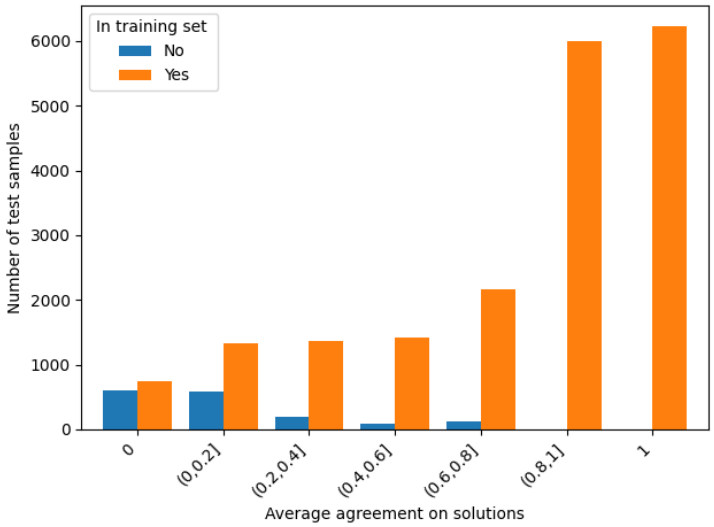

Figure 5: Distribution of test instances across system agreement levels, measured as the percentage of systems that correctly predicted the gold answer, split on whether the answer word appeared in the training set.

Cluster	Top terms	Clue examples
1	italiana, centro, tipo, re, successo, gioco, personaggio, noto, esserlo, piccola	La zona da cui nasce il Po; Malvagia; Si torna agli antichi; Tiranno... spagnolo; Condizione di chi ha due capi
2	famoso, scrittore, cantante, televisivo, storico, italiano, successo, soneria, soneria personaggio, tipo	Famoso dramma di Agatha Christie; Giornalista conduttore di un famoso Processo televisivo; Famoso scrittore; É famoso quello da Procida; Famoso cantante italiano
3	nome, femminile, dà, famosa, tipo, televisivo, successo, scrittore, soneria, soneria personaggio	Nome d'uomo; Altro nome dei ricci di mare; Nome di molti spagnoli, messicani e argentini; Altro nome del martin pescatore; Danno nome a una famosa sonata di Beethoven
4	film, famoso, noto, personaggio, nome, televisivo, tipo, scrittore, soneria, soneria personaggio	Un famoso film a episodi; La musica che fa da sottofondo al film; É "onorario" quello d'un film con Richard Gere; Un film americano d' autore; Un noto film del regista Joseph Losey
5	celebre, televisivo, tipo, storico, soneria personaggio, soneria, successo, scrittore, roma, provincia	È diventato celebre quello di notte; La casa editrice di un celebre dizionarista; Un celebre Scipione; Una celebre massima latina relativa alla salute; Un celebre idillio di Manzoni
6	francese, famoso, tipo, storico, soneria personaggio, televisivo, successo, scrittore,	Città francese; Pierre, regista cinematografico francese; La Svizzera... francese; Regione francese; Guillaume-Léon, politico francese
7	donna, nome donna, nome, storico, tipo, televisivo, successo, scrittore, soneria, soneria	Una donna generosa; Donna ricaduta nel peccato, per la teologia; Attraente come una donna; Donna che non ci vede più; Donna di casa
8	provincia, comune provincia, comune, storico, tipo, televisivo, successo, scrittore, soneria, soneria personaggio	La località misteriosa di oggi (provincia di Bergamo); Comune in provincia di Salerno; Comune in provincia di Lecco; Comune in provincia di Verbania; Lombardi di provincia

Table 5
Examples of clusters obtained by applying TF-IDF + K-Means (K = 10) to the set of instances incorrectly predicted by all systems. Each row reports representative clue examples and the corresponding top TF-IDF terms characterizing the cluster.

B. Crossword Clues

We report here the crossword clues associated with the grids discussed in Section 6.2.

ACROSS	**DOWN**
4. Il secreto di certe ghiandole	1. Assicurazione per chi guida
10. Una Pina del teatro	2. Il primo nome di Volontè
12. L'equipaggio di una nave pirata	3. Indaga sulla mafia
14. Centro di mira	4. Iniziali dell'attore Rea
15. Tornante nel calcio	5. Il rimpianto Moro (iniz.)
17. Fra capo e collo	6. Divisione amministrativa svedese
19. Agnese a Toledo	8. Si succedono nella vita
22. Il centro di Positano	9. Leggevano in pubblico gli editti
24. Permette di fissare e modellare i capelli	11. Le sue gesta sono narrate nei due libri dei Re della
25. Quality Assurance	18. Ugo Gregoretti
26. Quella di mezzo è preferibile	20. Fu per anni la grande rivale della Navratilova
28. Le separa la O	21. Comprano e vendono azioni
29. Nel cuore delle Ande	22. Insito, connaturato
31. Si susseguono spaventosamente	23. Tribunale Penale Internazionale
35. Metropolitana parigina	25. Quantità Massima
36. __, come lava! diceva Calimero	27. La Rodriguez, regina del fado
37. La si scrive per elogiare	31. Fuoco francese
38. La misura che si prende per tutelarsi	32. Una voce in fattura
42. Le aveva in testa Eva	33. Il 19 responsabile della pandemia
43. Lo dice chi accetta	34. L'attrice Di Benedetto
44. Una cintura per arti marziali	35. Hanno un gustoso ripieno
45. Le consonanti dell'ubiquo	38. Grosso pesce da tana
46. Precede... Tin-Tin	39. La Cina gli è vicina
48. Coda di gusta	40. In quel luogo per Livio
49. Quella fissa viene e non se ne va	41. Un'abbreviazione per... aree
51. Lo dicono i Romani... negando	45. Una ruota del Lotto
54. Tante erano le Caravelle di Colombo	47. Le gemelle... di Ennio
56. Articolo per muratore	50. Uno... starnuto
57. Mettere giorno, mese e anno	52. Il nome del ministro Ronchi
62. Piccolo a Torino	53. Il fumetto di un cane
63. Gli uccelli che possono essere cinerini	55. Da tenera diventa avanzata
	58. Il Nobel che ha dato il premio e non l'ha ricevuto (iniz.)
	59. Segue erre ed esse

Table 6
Definitions and solutions for Grid **(a)**.

ACROSS

1. Si porta sulle spalle
6. Si mette nel sacco
10. Sono pieni di quattrini
14. Il centro del Friuli
15. Abito di penitenza
16. Hanno preceduto i CD
17. Non si toccano in Fort Knox
19. L'...aula di Zenone
21. Alto vulcano siciliano
23. Lo può subire il pugile
24. Il nome dell'attore Selleck
25. Si può far di presenza
30. Un famoso re di Persia
32. Iniziali della Cruz
34. Canaletti veneziani
37. Ogni religione ha il suo
38. Simbolo dell'erbio
40. Cola sulla leccarda
47. Il battesimo della nave
48. Fa gridare i tifosi
50. Una patria della canzone italiana popolare (targa)
52. Lo indossa l'indiana
53. Un piacere da moderare
54. Venite in centro
55. La coppia in lotta
56. Recita un credo ateo
58. Grasso della critica (iniz.)
60. Eventi disastrosi

DOWN

1. Suddividere la posta
2. Si accorre per prestarlo
3. Il nucleo del nucleo
4. Noto passo appenninico
5. Precede le prime inserzioni dell'elenco
6. La rima della fattoria
7. Le lancia l'atterrito
8. Con "tap" forma un ballo
11. Provincia del Lazio
12. Così è l'adesione acritica a dogmi
13. C'è chi vorrebbe raddrizzar loro le gambe!
20. A questo punto... degli antichi vati
22. Prefisso per intelletto
25. Un po' acerbo
26. Le iniziali di Colleoni
33. La scuola con gli interni
35. Si apprezza nell'umorista
37. A lui spettava il governo della Repubblica di Venezia
39. Catanzaro
40. Il frutto che si mangia a chicco a chicco
41. Periodo geologico
42. Un risultato calcistico
49. Creano costosi monili
52. Un artista molto noto
53. Il famoso Björn del tennis
55. Pubblica guide (sigla)
57. Global Trade Organization
61. Si beve nel pomeriggio

Table 7
Definitions and solutions for Grid **(b)**.

ACROSS

1. Le hanno il custode e la guardia
3. Sensitivo
9. Il fumetto di un cane
12. La più splendente stella della Lira
14. International Normalised Ratio (sigla)
17. Molto magri, denutriti
19. Era domani fino a ieri
22. Capolavoro ricordato con l'Iliade
23. La "erre" dei Greci
26. Nel centro di Montreal
27. Ha dato il nome a un "vizio" (che peraltro non aveva)
31. Orribili Mostriciattoli Genetici
33. Sportello di consulenza fiscale
34. Uno dei figli del biblico Caleb
36. Il pittore del blu
38. La Hardin pianista
40. Andato via
42. Il nome del ministro Ronchi
45. Porzione di territorio
47. Fanno di maggio un miraggio
50. Mezzo Zulù
51. Verbo... canoro
54. La neo... dal fiocco rosa
56. Manicaretti siciliani
60. Indice della ricchezza nazionale
61. L'attrice Tushingham
62. Società... in breve
64. Dix che non dipinge
66. Pari in forma

DOWN

1. Ci sono da tavola
2. Scoraggiato, avvilito
3. Ruote da mulino
5. Dilatazione delle cavità cardiache
6. Gli spetta una provvigione per quel che ha... combinato
7. Uccello oceanico delle zone artiche
10. Ingrossamento
13. Il Lerner dell'Infedele
15. Targa del Nord-Ovest
18. È bellissimo... in mezzo
20. Il sacro calice ricercato da Parsifal
28. In mezzo al cancan
32. Si compra in edicola
35. Una carta favorevole
41. Il titolo del deputato (abbreviazione)
43. Solerte e laborioso
51. Girano sui colli...
52. Iniziali di Nuvolari
53. Li percorrono le gondole
55. Preposizione articolata
57. Già latino, sono inglese

Table 8
Definitions and solutions for Grid **(c)**.