# Crossword Space: Latent Manifold Learning for Italian Crosswords and Beyond

Cristiano Ciaccio[1], Gabriele Sarti[2], Alessio Miaschi[1] and Felice Dell'Orletta[1]

[1]ItaliaNLP Lab, Istituto di Linguistica Computazionale "A. Zampolli" (CNR-ILC), Pisa, Italy
[2]Center for Language and Cognition (CLCG), University of Groningen, The Netherlands

**Abstract**

Answering crossword puzzle clues presents a challenging retrieval task that requires matching linguistically rich and often ambiguous clues with appropriate solutions. While traditional retrieval-based strategies can commonly be used to address this issue, wordplays and other lateral thinking strategies limit the effectiveness of conventional lexical and semantic approaches. In this work, we address the clue answering task as an information retrieval problem exploiting the potential of encoder-based Transformer models to learn a shared latent space between clues and solutions. In particular, we propose for the first time a collection of siamese and asymmetric dual encoder architectures trained to capture the complex properties and relation characterizing crossword clues and their solutions for the Italian language. After comparing various architectures for this task, we show that the strong retrieval capabilities of these systems extend to neologisms and dictionary terms, suggesting their potential use in linguistic analyses beyond the scope of language games.

**Keywords**

Language Games, Crosswords, Semantic Similarity, Embeddings, Natural Language Processing, Information Retrieval

## 1. Introduction and Background

Language games have emerged as compelling benchmarks for evaluating the reasoning capabilities of language models (LMs), offering structured challenges that require diverse cognitive skills including wordplay comprehension, lateral thinking, and cultural knowledge integration [2, 3, 4, 5]. Among popular language games, crossword puzzles stand out as particularly challenging, demanding not only linguistic competence but also extensive world knowledge, cultural awareness, and lateral thinking skills [6, 7, 8, 9]. While recent advances in Large Language Models have shown impressive performance on many natural language understanding tasks, their effectiveness on language games remains constrained by fundamental limitations in accessing linguistic and culturally-relevant knowledge, in particular for less-resourced non-English languages [5].

Before the advent of modern language models, most approaches to crossword solving relied on retrieval-based methods and shallow lexical and semantic features to identify relevant information [10, 11]. For example, [12] proposed a retrieval model that exploited lexical resources and similarity metrics to match clues to candidate answers in Italian. In a subsequent work, [13] introduced SACRY, a system that incorporated syntactic information and ranking strategies to improve clue-answer matching. Importantly, fill-in-the-blank clues and clues representing anagrams or linguistic games are often omitted. While these traditional retrieval systems typically relied on surface-level features - such as lexical overlap, part-of-speech patterns, and predefined similarity measures — the identification of viable crossword solutions often involves more nuanced interpretations, including the use of wordplay, homophones and other unusual elements. For example, the clue *"Producono con procedimenti lenti"* plays on the polysemanticity of *lenti* (in Italian, either "slow" MASC. PLUR., or "lenses"), and could have *ottici* (opticians) as a valid solution. These kinds of subtle connections hinder the viability of tra-

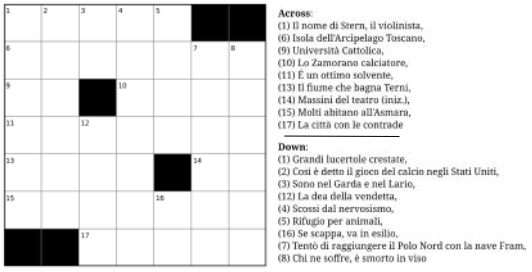


**Figure 1:** An example of symmetric-style crossword puzzle. The grid was populated using clues taken from the test set. The correct solution, which was autonomously found leveraging our system, is in Appendix A.

*CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy [1]*
cristiano.ciaccio@ilc.cnr.it (C. Ciaccio); g.sarti@rug.nl (G. Sarti); alessio.miaschi@ilc.cnr.it (A. Miaschi); felice.dellorletta@ilc.cnr.it (F. Dell'Orletta)
https://gsarti.com (G. Sarti); https://alemiaschi.github.io/ (A. Miaschi); http://www.italianlp.it/people/felice-dellorletta/ (F. Dell'Orletta)
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)

ditional retrieval systems in the context of crossword games.

Recent advances in cross-modal learning, particularly in vision-language models such as CLIP [14, 15], have demonstrated the effectiveness of dual encoder architectures in learning shared representations across different modalities. These approaches typically employ separate encoders for each modality, training them to project inputs into a common latent space where semantically related items cluster together. Inspired by these successes, we propose adapting this paradigm to the domain of language games, specifically focusing on the relationship between crossword clues and their solutions[1].

In this work, we evaluate several dual encoder architectures designed to learn effective representations for crossword puzzle elements (see Figure 1 for an example of a crossword puzzle). Our approaches treat clues and solutions as distinct "modalities" that can be embedded to a shared latent space. The clue encoder must understand various forms of wordplay, cultural references, and linguistic devices, while the solution encoder must represent semantic, lexical and grammatical characteristics of the words. By training these encoders jointly with a contrastive objective, we create a retrieval system specifically optimized for the complexities of crossword puzzles. Our contributions are threefold: (1) We formalize the problem of specialized retrieval for language games and demonstrate the limitations of generic retrieval approaches in this domain; (2) We introduce and evaluate multiple dual encoder architectures tailored for Italian crossword puzzles, exploring different design choices and training strategies; (3) We demonstrate the utility of our learned representations for solution ranking and explore their generalization capabilities to neologisms. Our experimental results show that domain-specific models significantly outperform generic alternatives, suggesting that specialized retrieval mechanisms are essential for effectively ranking plausible alternatives in this domain.

# 2. Our Approach

Our approach formalizes crossword's clues answering as an information retrieval problem. Given a clue $c_i$ from the set $\mathscr{C} = \{c_1, \dots, c_n\}$ and a matching solution $s_i$ from the finite set of all available solution words $\mathscr{S} = \{s_1, \dots, s_n\}$, our system scores the similarity of a subset of candidates $\mathscr{S}^* \in \mathscr{S}$ with $c_i$ to produce a similarity-based ranking. Inspired by CLIP's approach [14], we opted for a a dual encoder architecture [16], composed of two pre-trained transformers encoders [17] —referred to as *towers*— which are fine-tuned on clue-solution pairs with a contrastive learning objective to learn a joint embedding space between clues and words.

In the following sections, we describe in detail the architecture of our model (Section 2.1), the datasets used for the experiments (Section 2.2), the encoder models employed (Section 2.3), the experimental setting (Section 2.4), the evaluation strategy adopted to assess the system's performance (Section 2.5).

## 2.1. Model's Architecture

To explore the effectiveness of our approach, we experiment with different encoder-based models for initializing the encoder towers, each fine-tuned and tested on a dataset of Italian crossword clues. As shown by Dong et al. [18], to effectively learn a shared parameter space using a dual encoder, there are two main architectural options: (a) the **Siamese Dual Encoder (SDE)** and (b) the **Asymmetric Dual Encoder (ADE)** with a shared linear projection. Both consist of two pre-trained Transformers encoders, in our case, a clue-encoder $f_1$ and solution-encoder $f_2$, trained to produce representations $\mathbf{c}_i = f_1(c_i)$ and $\mathbf{s}_i = f_2(s_i)$ by average pooling, where both $\mathbf{c}_i, \mathbf{s}_i \in \mathbb{R}^m$. These are linearly projected into a shared feature space $C \in \mathbb{R}^n$ in order to maximize the cosine similarity between positive pairs $(\mathbf{c}_i, \mathbf{s}_i^+)$ and minimize it for negative ones $(\mathbf{c}_i, \mathbf{s}_i^-)$. The distinction between SDE and ADE lies in the parameter sharing: while in SDE the two encoders $f_1$ and $f_2$ have tied parameters ($\theta_{f1} = \theta_{f2}$), in ADE the two encoder towers have untied parameters ($\theta_{f1} \neq \theta_{f2}$) but share a final layer norm and the linear transformation $T : \mathbb{R}^m \to \mathbb{R}^n$, which is essential to achieve an effectively shared space. Having separate encoders can be advantageous when modeling different modalities and distributions since it allows the two encoders to specialize independently on the specific nuances of the input types they process. To assess which of the two architectures is better suited for our task, we conduct preliminary experiments on both and compare their results in Section 3.1.

## 2.2. Dataset

For training our dual encoders, we employ the ItaCW crossword dataset [19], containing 125k unique definition-word pairs. We expand this collection with additional clue-solution pairs found on the web, and deduplicate the resulting set of entries, obtaining a total of 416,407 samples.

In addition to the original crossword dataset, to evaluate the out-of-distribution performances of our system we also consider word–definition pairs automatically extracted from the Italian Wiktionary, neologisms from the ONLI (Osservatorio Neologico della Lingua Italiana[2]) and

[1]Code, models and datasets are released at: https://github.com/snizio/Crossword-Space.

[2]https://www.iliesi.cnr.it/ONLI/.

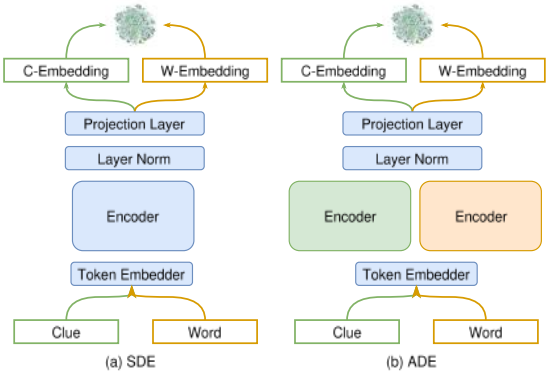


**Figure 2:** The two architectures tested: (a) Siamese Dual Encoder (SDE), (b) Asymmetric Dual Encoder with shared linear projection (ADE). Final clue (C) and solution (S) embeddings are projected to a shared latent space in both architectures. Blue modules are shared.

a set of 100 recently lexicalized neologisms [20]. Since some word-definitions pairs maintain the same inferential relation that occurs for most clue-solution pairs (excluding nuanced and specific crossword cases), augmenting the dataset with these specific resources allows us to assess the performance variations and generalization to different linguistic settings that exhibit the same input-output structure of crosswords, offering a natural extension to the main dataset. Specifically, the usage of dictionary data is twofold: (a) to understand whether augmenting the train set with word-definition pairs can enhance downstream performance on the crossword data; (b) to assess the extent to which models trained on word-clue pairs can be used to answer dictionary definitions. On the other hand, the ONLI and the 100-neologisms dataset will be used to test the robustness and generalization of our systems, therefore simulating a scenario where a novel term appears in a crossword, as is often the case. The ONLI covers a wide range of neologisms appearing on national and local newspapers, thus strictly related to the Italian culture, including newly coined or derived formations, internationalisms, foreignerisms, technical terms and some authorial neologisms until 2019; while the 100-neologism dataset consists of lemmas extracted from various online dictionaries (lexicalized after 2020) that focus mostly on politics, COVID-19 social dynamics and contain several foreignerisms.

## 2.3. Models

As backbone models, we choose several pre-trained encoders available for the Italian language, varying in parameter size and pre-training approaches. Specifically, we picked the encoders of `IT5-small` (35M) and `IT5-base` (110M) from the IT5 family [21] of encoder-decoders pre-trained on the Italian cleaned split of the MC4 [22]; `Italian-ModernBERT-base`[3] (135M) and `Italian-ModernBERT-base-embed-mmarco-triplet`[4] (135M), both based on the ModernBERT architecture [23] and pretrained on Italian with the latter being finetuned in a sentence-transformer fashion [24] on the mMARCO dataset [25]; lastly, we employed `paraphrase-multilingual-mpnet-base-v2`[5] [26] (278M), a multilingual model based on XLM-RoBERTa already tuned as a sentence embedder.

## 2.4. Experimental setting

We begin by comparing ADE and SDE architectures to assess the optimal approach for our clues answering task. Subsequently, each model is trained across two dataset configurations: the first one consists of using only a subset of the crossword dataset as the training set, the second one introduces also a split of the Italian Wiktionary in the training data. On the other hand, the evaluation is always performed on an held-out test set composed of crosswords clues, dictionary [6], ONLI and the 100-neologisms definitions. After merging all the data sources we split the resulting dataset into 90% train, 5% validation and 5% test (see Table 1).

We train our SDE and ADE architectures to minimize the symmetric InfoNCE loss used in CLIP [14] with in-batch negatives. During training, for each step, we mine for $(B-1)*r$ hard negatives that have the highest similarity to the positive target, where $B$ is the batch size and $r \in [0,1]$ is a fraction that determines how many of the hardest negatives are kept [27]. Formally, let $\mathbf{c}_i \in \mathbb{R}^m$ be the normalized embedding of the $i$-th clue, and $\mathbf{s}_j \in \mathbb{R}^m$ the normalized embedding of the $j$-th solution word. Let $\tau = \exp(t)$ be a learnable temperature parameter, and let $\mathcal{N}_i$ denote the indices of the top-$k$ hardest negatives. The *clue-to-solution* contrastive loss is defined as $\mathscr{L}_{c\to s}$:

$$\frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp\left(\tau \cdot \cos(\mathbf{c}_i, \mathbf{s}_i)\right)}{\exp\left(\tau \cdot \cos(\mathbf{c}_i, \mathbf{s}_i)\right) + \sum_{j\in\mathcal{N}_i} \exp\left(\tau \cdot \cos(\mathbf{c}_i, \mathbf{s}_j)\right)}$$

Similarly, the *solution-to-clue* loss is $\mathscr{L}_{s\to c}$:

$$\frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp\left(\tau \cdot \cos(\mathbf{s}_i, \mathbf{c}_i)\right)}{\exp\left(\tau \cdot \cos(\mathbf{s}_i, \mathbf{c}_i)\right) + \sum_{j\in\mathcal{N}_i} \exp\left(\tau \cdot \cos(\mathbf{s}_i, \mathbf{c}_j)\right)}$$

The final symmetric contrastive loss is the average of the two losses:

$$\mathscr{L} = \frac{1}{2}\left(\mathscr{L}_{c\to s} + \mathscr{L}_{s\to c}\right)$$

[3] DeepMount00/Italian-ModernBERT-base.
[4] nickprock/Italian-ModernBERT-base-embed-mmarco-triplet.
[5] sentence-transformers/paraphrase-multilingual-mpnet-base-v2.
[6] When augmenting the dataset with dictionary definitions, all inflected forms are dropped.

| | Train | Val. | Test |
|---|---|---|---|
| **Crosswords** (Cross.) | 374,713 | 20,853 | 20,841 |
| **Dictionary** (Dict.) | 78,103 | 4,303 | 4,316 |
| **ONLI** | - | - | 2,986 |
| **Neologisms** (Neo.) | - | - | 100 |
| **Tot.** | 452816 | 25156 | 28213 |

**Table 1**
Train, validation and test split sizes for the tested datasets.

The training setup is the same across all models, architectures and dataset configurations. Each model is trained for a maximum of six epochs with a batch size $B$ of 256 using AdamW [28] with a linearly decaying learning rate. The hard negatives fraction decays linearly during training from 0.8 to 0.05 (for detailed hyperparameter see Appendix B).

Before the test phase, all available solution words $\mathcal{S}$ are encoded into their relative embeddings, normalized and stored into a vector database. During inference, for a normalized clue embedding $\mathbf{c}_i$, the retrieval is performed leveraging the FAISS library [29] by inner product on the stored embedding matrix $\mathbf{E}_{|\mathcal{S}|\times m}$, where $|\mathcal{S}| = 106,988$ is the cardinality of the finite set of available solution words and $m$ is the embeddings dimension.

**Baselines** In order to further assess the performance of our models, we include and compare several baselines based on two main approaches: (a) **clues to clues** (**c2c**), where, given an input clue, the most similar clues and their corresponding solutions are retrieved from the training set, as commonly done in the crossword solving literature [13, 30, 31]; and (b) **clues to solutions** (**c2s**), where solutions are retrieved by directly comparing the given clue against the set of all possible solutions. For c2c we computed the similarity scores between clues using (1) Levenshtein distance (c2c-lev), (2) BM25 (c2c-BM25) and (3) the cosine similarities between clues representations obtained with `paraphrase-multilingual-mpnet-base-v2` (c2c-MPNet) as a standalone sentence embedder and without any finetuning. For the c2s baseline, we rank the answers by cosine similarity between the clue and all solutions using, as before mentioned, the `paraphrase-multilingual-mpnet-base-v2` (c2s-MPNet). To ensure a fair comparison between models and baselines, the c2c retrieval is conducted against the clues in the training set, augmented with dictionary definitions.

## 2.5. Evaluation

To evaluate the retrieval performance of our trained models, we adopt the following standard metrics:

**Accuracy@1/10/100/1000** is the accuracy in retrieving the correct solution word given the corresponding clue, considering the top 1/10/100/1000 most similar words retrieved by our system as valid.

| | Arch. | Accuracy@ | | | | MRR |
|---|---|---|---|---|---|---|
| | | 1 | 10 | 100 | 1000 | |
| **Cross.** | ADE | **.33** | **.63** | **.80** | .90 | **.43** |
| | SDE | .20 | .58 | **.80** | **.91** | .33 |
| **Dict.** | ADE | .07 | .22 | .42 | .65 | .12 |
| | SDE | **.10** | **.28** | **.47** | **.67** | **.16** |
| **ONLI** | ADE | .07 | .21 | .45 | .70 | .12 |
| | SDE | **.13** | **.32** | **.54** | **.74** | **.20** |
| **Neo.** | ADE | .05 | .11 | .25 | .63 | .07 |
| | SDE | **.09** | **.22** | **.39** | **.64** | **.14** |

**Table 2**
Test results for ADE and SDE architectures across the four tested domains. Top scores per dataset are marked in **bold**.

**Mean Reciprocal Rank (MRR)** represents how well a system ranks the first relevant result by averaging the reciprocal ranks of the first relevant item across all queries.

To simulate a more realistic crossword puzzle solving scenario, we also report metrics for candidate words retrieved from the filtered set $S_\ell \subseteq S$ containing only words with the same character length $\ell$ as the target word $s_{target}$, formally $S_\ell = \{s \in S \mid \text{len}(s) = \ell\}$. We append an asterisk when reporting metrics that include this filtering process (e.g. Acc@10* or MRR*).

# 3. Results

We begin by comparing the two architectures under evaluation, SDE and ADE, and then report the performance of all tested models for all datasets using the best-performing architecture.

## 3.1. Siamese vs. Asymmetric Encoders

Table 2 reports our test results for the `paraphrase-multilingual-mpnet-base-v2` model, the largest we trained, which guided our choice between the siamese and asymmetric architecture variants. Interestingly, **the asymmetric architecture shows a substantial gain in performance only for crossword clues** and especially in ranking terms (Acc@1 +13%, MRR +10%), while being outperformed by SDE in all other linguistic settings, although with a narrower gap. We hypothesize that due to the peculiar inference links that relate clues and target words, an asymmetric architecture could be better at enriching representations with input/output nuances separately, rather than jointly as in ADE models. Indeed, many puzzles feature clues with wordplay intended to

| | | Accuracy@ | | | | | | | | MRR | MRR* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 1* | 10 | 10* | 100 | 100* | 1000 | 1000* | | |
| **Cross.** | c2c-lev | .20 | .30 | .39 | .50 | .53 | .63 | .63 | .74 | .27 | .37 |
| | c2c-BM25 | .25 | .38 | .49 | .60 | .63 | .69 | .69 | .76 | .33 | .46 |
| | c2c-MPNet | .27 | .39 | .45 | .59 | .61 | .73 | .74 | .84 | .33 | .46 |
| | c2s-MPNet | .003 | .03 | .02 | .10 | .09 | .23 | .18 | .50 | .01 | .05 |
| | IT5-small | $.08_{+.00}$ | $.26_{+.01}$ | $.29_{+.02}$ | $.57_{+.01}$ | $.57_{+.03}$ | $.81_{+.01}$ | $.81_{+.02}$ | $.95_{+.00}$ | $.15_{+.01}$ | $.36_{+.01}$ |
| | IT5-base | $.15_{+.01}$ | $.41_{+.01}$ | $.48_{+.02}$ | $.74_{+.01}$ | $.75_{+.02}$ | $\mathbf{.91}_{+.00}$ | $\mathbf{.90}_{+.01}$ | $\mathbf{.98}_{+.00}$ | $.26_{+.01}$ | $.52_{+.01}$ |
| | ModernSBert | $.25_{+.01}$ | $.45_{+.01}$ | $.52_{+.02}$ | $.72_{+.01}$ | $.73_{+.02}$ | $.87_{+.01}$ | $.86_{+.02}$ | $.96_{+.01}$ | $.34_{+.01}$ | $.55_{+.01}$ |
| | ModernBert | $.09_{+.01}$ | $.27_{+.01}$ | $.30_{+.04}$ | $.57_{+.03}$ | $.58_{+.04}$ | $.78_{+.05}$ | $.81_{+.03}$ | $.81_{+.15}$ | $.16_{+.02}$ | $.37_{+.02}$ |
| | MPNet-base | $\mathbf{.33}_{-.01}$ | $\mathbf{.54}_{-.00}$ | $\mathbf{.63}_{+.00}$ | $\mathbf{.80}_{+.00}$ | $\mathbf{.80}_{+.01}$ | $\mathbf{.90}_{+.01}$ | $\mathbf{.90}_{+.01}$ | $.97_{+.00}$ | $\mathbf{.43}_{-.01}$ | $\mathbf{.64}_{-.00}$ |
| **Dict.** | c2c-lev | .04 | .06 | .07 | .10 | .10 | .17 | .16 | .30 | .05 | .07 |
| | c2c-BM25 | .05 | .09 | .10 | .17 | .18 | .27 | .26 | .39 | .07 | .12 |
| | c2c-MPNet | .06 | .12 | .13 | .25 | .25 | .39 | .38 | .54 | .08 | .16 |
| | c2s-MPNet | .05 | .13 | .15 | .30 | .30 | .46 | .45 | .67 | .08 | .19 |
| | IT5-small | $.05_{+.06}$ | $.14_{+.10}$ | $.15_{+.12}$ | $.35_{+.14}$ | $.33_{+.15}$ | $.59_{+.13}$ | $.58_{+.15}$ | $.85_{+.08}$ | $.08_{+.08}$ | $.21_{+.12}$ |
| | IT5-base | $\mathbf{.09}_{+.08}$ | $\mathbf{.24}_{+.11}$ | $\mathbf{.27}_{+.13}$ | $\mathbf{.49}_{+.13}$ | $\mathbf{.48}_{+.14}$ | $\mathbf{.71}_{+.11}$ | $\mathbf{.70}_{+.13}$ | $\mathbf{.91}_{+.05}$ | $\mathbf{.15}_{+.10}$ | $\mathbf{.33}_{+.12}$ |
| | ModernSBert | $.04_{+.07}$ | $.13_{+.15}$ | $.14_{+.18}$ | $.32_{+.23}$ | $.31_{+.25}$ | $.57_{+.20}$ | $.56_{+.22}$ | $.83_{+.12}$ | $.07_{+.11}$ | $.19_{+.18}$ |
| | ModernBert | $.03_{+.07}$ | $.12_{+.12}$ | $.13_{+.13}$ | $.32_{+.18}$ | $.31_{+.19}$ | $.53_{+.20}$ | $.56_{+.19}$ | $.56_{+.37}$ | $.07_{+.09}$ | $.19_{+.14}$ |
| | MPNet-base | $\mathbf{.07}_{+.11}$ | $\mathbf{.19}_{+.17}$ | $\mathbf{.22}_{+.19}$ | $\mathbf{.43}_{+.20}$ | $\mathbf{.42}_{+.21}$ | $\mathbf{.66}_{+.17}$ | $\mathbf{.65}_{+.18}$ | $\mathbf{.87}_{+.09}$ | $.12_{+.13}$ | $\mathbf{.27}_{+.18}$ |
| **ONLI** | c2c-lev | .01 | .02 | .02 | .03 | .03 | .04 | .03 | .07 | .01 | .02 |
| | c2c-BM25 | .01 | .02 | .02 | .04 | .04 | .07 | .06 | .09 | .01 | .03 |
| | c2c-MPNet | .04 | .07 | .07 | 0.1 | .09 | .12 | .11 | .13 | .05 | .08 |
| | c2s-MPNet | .11 | .30 | .30 | .52 | .49 | .68 | .65 | .84 | .18 | .38 |
| | IT5-small | $.08_{+.04}$ | $.23_{+.09}$ | $.23_{+.10}$ | $.48_{+.11}$ | $.44_{+.11}$ | $.71_{+.09}$ | $.67_{+.11}$ | $.91_{+.04}$ | $.13_{+.06}$ | $.31_{+.10}$ |
| | IT5-base | $\mathbf{.16}_{+.07}$ | $\mathbf{.41}_{+.10}$ | $\mathbf{.42}_{+.10}$ | $\mathbf{.68}_{+.08}$ | $\mathbf{.65}_{+.09}$ | $\mathbf{.85}_{+.05}$ | $\mathbf{.83}_{+.07}$ | $\mathbf{.96}_{+.02}$ | $\mathbf{.25}_{+.08}$ | $\mathbf{.50}_{+.09}$ |
| | ModernSBert | $.05_{+.03}$ | $.18_{+.10}$ | $.16_{+.09}$ | $.41_{+.14}$ | $.36_{+.14}$ | $.69_{+.10}$ | $.64_{+.11}$ | $.90_{+.04}$ | $.09_{+.05}$ | $.26_{+.11}$ |
| | ModernBert | $.06_{+.02}$ | $.20_{+.07}$ | $.18_{+.07}$ | $.43_{+.09}$ | $.40_{+.08}$ | $.63_{+.11}$ | $.64_{+.06}$ | $.64_{+.27}$ | $.10_{+.04}$ | $.28_{+.07}$ |
| | MPNet-base | $.07_{+.05}$ | $.24_{+.12}$ | $.21_{+.14}$ | $.50_{+.15}$ | $.45_{+.16}$ | $.74_{+.11}$ | $.70_{+.12}$ | $.92_{+.04}$ | $.12_{+.08}$ | $.33_{+.13}$ |
| **Neo.** | c2c-lev | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 |
| | c2c-BM25 | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 |
| | c2c-MPNet | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 | .01 |
| | c2s-MPNet | **.15** | **.31** | **.29** | **.50** | .47 | .67 | .67 | .85 | **.20** | **.36** |
| | IT5-small | $.03_{+.01}$ | $.12_{+.03}$ | $.09_{+.06}$ | $.23_{+.16}$ | $.24_{+.13}$ | $.58_{+.08}$ | $.58_{+.08}$ | $.90_{+.04}$ | $.05_{+.02}$ | $.17_{+.06}$ |
| | IT5-base | $.09_{+.02}$ | $\mathbf{.24}_{+.07}$ | $\mathbf{.19}_{+.17}$ | $.47_{+.09}$ | $\mathbf{.48}_{+.06}$ | $\mathbf{.74}_{+.08}$ | $\mathbf{.71}_{+.08}$ | $\mathbf{.95}_{+.01}$ | $.13_{+.06}$ | $\mathbf{.32}_{+.09}$ |
| | ModernSBert | $.03_{+.03}$ | $.10_{+.05}$ | $.08_{+.06}$ | $.26_{+.15}$ | $.24_{+.16}$ | $.56_{+.14}$ | $.54_{+.14}$ | $.84_{+.07}$ | $.05_{+.04}$ | $.16_{+.08}$ |
| | ModernBert | $.01_{+.05}$ | $.13_{+.03}$ | $.08_{+.10}$ | $.31_{+.12}$ | $.31_{+.10}$ | $.54_{+.12}$ | $.54_{+.11}$ | $.54_{+.35}$ | $.05_{+.06}$ | $.19_{+.06}$ |
| | MPNet-base | $.05_{+.04}$ | $.16_{+.12}$ | $.11_{+.14}$ | $\mathbf{.30}_{+.30}$ | $\mathbf{.25}_{+.34}$ | $\mathbf{.62}_{+.22}$ | $\mathbf{.63}_{+.19}$ | $.94_{+.00}$ | $.07_{+.07}$ | $.21_{+.18}$ |

**Table 3**
Model performances across test sets for each dataset, with **bold** values indicating best performances within each dataset. Footer values show the difference in performance after augmenting the original crosswords-only training set with dictionary definitions (Dict.).

be taken metaphorically or in other non-literal senses. For example, a correct answer for the clue "half a dance" might be *can* (half of the dance named *cancan*). In this setting, an encoder specialized in enriching the representation of the clue with dance names might be necessary to achieve good performances. On the other hand, for dictionary-like entries, there is no sufficient need to develop uniquely independent representations (as shown by the ADE performance drop) since word-definition pairs are typically symmetric in meaning and structure. In these settings, the same encoder can effectively capture both sides of the pair, benefiting from shared parameters that reinforce semantic alignment. Given that our primary interest in this work lies in crosswords, we adopt the ADE architecture with a shared linear projection for subsequent evaluations.

## 3.2. Main results

Table 3 shows the results of all models across the various test sets:

**Crosswords** MPNet-base, ModernSBert and IT5-base strongly outperform all baselines, especially at higher candidate sizes and when applying length filtering ("*"). Overall, the MPNet-base yields the best result, suggesting that model size has a positive effect on improving task performance. In terms of MRR, ModernSBert is the second-best performer, substantially outperforming its only pre-trained counterpart, ModernBert, underscoring the additional value of using models that have already undergone a sentence finetuning phase for boosting retrieval performance. All baselines leveraging the c2c approach are superior when confronted with IT5-small and ModernBert, especially in terms of MRR. Interestingly, incorporating dictionary data into the training set yields only moderate overall gains and does not significantly

| | Query | c2c-BM25 | c2c-MPNet | IT5-base | MPNet-base | Target |
|---|---|---|---|---|---|---|
| **Cross.** | Il numero di chi comanda | direzione, autorità | dieci, fili | numero romano, numero | **uno**, centouno | uno |
| | ____ urrà! | laura, liv | incantesimo, toccaferro | pelu, miki | **hip**, ip | hip |
| | Lido senza pari | arenile, ostia | bti, cv | vl, dl | dd, **ld** | ld |
| | Colore monosillabo | tinta, si | toni, pallore | indaco, **blu** | si, ma | blu |
| **Dict.** | infiammazione acuta o cronica di un nervo | epididimite, endocardite | linfadenopatia, linfadenopatico | tendinite, flogosi | **nevrite**, spondilite | nevrite |
| | navigare seguendo la linea di costa | cabotare, piaggiare | cabotare, litoranea | navigare, **costeggiare** | **costeggiare**, circumnavigare | costeggiare |
| **ONLI** | Terrorismo di matrice anarchica. | diagonalizzabile, conio | eversivo, terrorista | **anarcoterrorismo**, anarcoinsurrezionalismo | fascismo, hitlerismo | anarcoterrorismo |
| | Il potere delle mafie. | stampa, plenipotenza | cupola, dia | malaffare, **mafiocrazia** | direttorio, establishment | mafiocrazia |
| **Neos.** | Chi pratica hacking con lo scopo di divulgare slogan nazisti. | sport, autostop | hacker, pirateria | **nazi-hacker,** hacker | cyberpirata, sabotatore | nazi-hacker, |
| | Lavoro da remoto, svolto in prossimità della propria abitazione | rincasare, vicina | domestici, computer | telelavoro, smartworking | masserizia, trasferta | nearworking |

**Table 4**
Some examples of retrieved answers across baselines, models and test sets.

impact the results, further emphasizing that definitions and crossword clues originate from different linguistic distributions.

**Dictionary** All models, and especially baselines, severely drop in performance when dealing with dictionary data. Furthermore, the rank changes: IT5-base obtains higher results than the multilingual MPNet-base, despite having half of the parameters. As expected, enhancing the training set with dictionary samples yields substantial gains across all models; especially, the MPNet-base increases results-wise more than the IT5-base, resulting in similar scores for both models.

**ONLI** For ONLI neologisms, all c2c baselines continue to decline while c2s-MPNet gains significantly w.r.t. crossword clues and dictionary definitions. IT5-base achieves the best results, with a substantial gap from the MPNet-base. As in the dictionary setting, augmenting the dataset with dictionary definitions yields improvements, although more moderate. ONLI neologisms are retrieved better than dictionary words, even when augmenting the dataset. One hypothesis for this phenomenon is that crossword clues are more aligned with the definitions of neologisms, as they may reflect similar linguistic strategies. Both crossword clues, particularly those involving wordplay, and journalistic neologism definitions often rely on compositionality. For example, clues such as "half a dance" or "prefix meaning new" require the decomposition and reinterpretation of word parts, similarly to many neologisms in ONLI are defined through transparent compounds or affix-based constructions (e.g., *mafiocracy* = *mafia* + *-cracy*). This shared reliance on compositionality may partially explain why models trained on crossword clues generalize better to ONLI neologisms than to standard dictionary definitions, which are often more rigid and semantically grounded.

**Neos.** Models perform poorly in this setting. However, they still widely outperform all c2c baselines, which are almost fully incapable of retrieving correct answers. Interestingly, the simple c2s-MPNet approach yields strong results, achieving top Acc@1 and Acc@1* scores. Overall, IT5-base achieves the best results, beating the c2s-baseline from Acc@10, followed by the multilingual MPNet-base. As for ONLI and Dict., all models benefit importantly from training on dictionary definitions and, especially, the MPNet-base in this configuration becomes the top performer in terms of Acc@10*, Acc@100, Acc@100* and Acc@1000.

### 3.2.1. Discussion

Overall, we observe an interesting trend concerning baselines: while all c2c (clues to clues) approaches perform reasonably well on crosswords, their performance drastically drops when dealing with dictionary terms and neologisms. On the other hand, the c2s-MPNet baseline, which directly confronts clues and solutions during retrieval, exhibits an inverse trend, performing better with definition-like clues than with crossword clues. These results further corroborate the hypothesis that clues and definitions have a different relation to target words: **words and definitions are more semantically aligned, from a distributional point of view, than crossword clues and solutions**. Furthermore, the extremely low performance of c2c-baselines on neologisms

**Across**: (1) Un fuoco acceso in segno di gioia, (5) Così sono certe illusioni, (8) Affettato in società, (12) Monte biblico, (13) Varia secondo il pesce, (14) Un terribile male (sigla), (15) Sovrintendenti dei Carabinieri, (19) Un'ipotesi che fa dubitare, (20) Basta uno spavento per mutarlo, (21) Uno pesa cento grammi, (23) Il lago di Cleveland, (24) Il rumore di un crollo, (28) Lo è colei che dice... ormai!, (31) In fondo al treno, (32) La droga di Caienna, (35) Il successore di Sansone, (36) dio primordiale nella mitologia greco-romana, (38) Così è l'eccezione, (39) La Tanzi dello schermo, (40) La produce un baco

**Down**: (12) Una temuta malattia (sigla), (32) Cambiano la mela in pera, (1) Lancia fasci di luce, (33) Le iniziali della Aulin, (2) Capace e... arruolato, (34) Poco prevedibile, (3) Un flusso precipitoso di parole, (4) Un tipo di media calcolata per la velocità, (16) Motore alimentato a gasolio, (17) Due lettere per l'Italia., (29) La società dei linguisti italiani (sigla), (18) Sono opposti nella bussola, (30) Un grido cui si faceva eco, (5) Breve paragrafo, (24) Targa perugina, (6) Tutt' altro che sommi, (25) Ancona sulle targhe, (7) In mezzo e in centro, (21) Il Beta amico di Archimede, (19) percezione di sé come cittadino, (8) Mandare lampi e fulmini, (9) La risposta degli incerti, (22) Quasi adesso..., (37) Attenzione all'inizio, (10) Modesti meno mesti, (26) La Netrebko celebre soprano, (11) Sigla di Brescia, (27) Bella isola del Dodecaneso

**Figure 3:** An autonomously solved crossword puzzle. Clues taken from the test set were answered by using our system to retrieve the fifty closest answers, and the complete grid was filled using the Z3 SAT solver.

confirms that clues-to-clues mappings are insufficient to handle lexical innovation in crossword puzzles. This supports our initial motivation for a joint latent space that leverages rich distributed representations, enabling the modeling of unseen clues and solutions for the task of crossword retrieval. Finally, the **majority of our trained systems achieved better results than baselines on crossword clues** with the biggest and multilingual model, MPNet-base, achieving the best results, closely followed by the IT5-base. For neologisms in particular, the better performances of the monolingual IT5-base encoder despite its smaller parameter count suggest that **language-specific training might benefit retrieval in domains heavily influenced by culture and language-specific lexical innovation dynamics**.

## 4. Analysis and Applications

This section provides further explorations in applications and properties of our crossword embeddings systems.

**Examples Analysis** Table 4 reports some examples of the Top2 retrieved answers across baselines, models and test sets. For this purpose, we manually selected cases showing the limitations of traditional baselines, e.g. crossword clues carrying a non-literal meaning. For example, the cryptic-style clue "Lido senza pari" (transl. *Beach without even*) requires interpreting *even* as referring to the characters in even positions inside the word *lido*. Baselines do not capture this meaning nuance, while some of our models arrive at the correct solution, despite the well-known problem of character awareness in character-blind models [32, 33]. Another interesting case involves neologisms: baselines are unable to retrieve the correct answers since they represent a fringe minority in the available pool of definitions and solutions. On the other hand, our models, especially the monolingual IT5, show signs of generalization and were able to retrieve the correct answers despite not being trained on them.

**Automated Crossword Solving** Despite not being the main focus of this article, we tried to leverage our system to automatically solve crossword puzzles as a concrete application of clues answeringcrossword. Figures 1 and 3 show an example of a crossword puzzle, built entirely from clues in the test sets, automatically filled using the Z3 SMT (Satisfiability Modulo Theories) solver [34][7], leveraging candidates retrieved by the MPNet-base model. Specifically, by treating crossword puzzles as a satisfiability problem, we can define a set of first-order logical constraints that must be satisfied across all variables (grid cells) to find valid solutions: each clue corresponds to a sequence of grid variables constrained to match one of its candidate answers, forming a disjunctive (OR) group. These candidate-level constraints are then combined conjunctively (AND) across all clues. Additionally, for intersecting cells, equality constraints are enforced to ensure character consistency between overlapping horizontal and vertical words. The final formula, composed of these conjunctive and disjunctive logical statements, is passed to the solver, which searches for a globally consistent solution that satisfies all constraints simultaneously. Despite the complexity of this approach, which requires that each candidate set contains the correct solution, our biggest model, MPNet-base, was able to solve entirely some small-medium grids using a candidate size $10 \leq k \leq 50$, confirming the effectiveness of our system. We posit that a strategy iterating Z3 solving attempts over progressively larger candidate sizes could provide a strong baseline for crossword solving systems with a given computational budget, and we leave such assessment to future work.

## 5. Conclusion and Future Work

In this work, we introduced and evaluated dual encoder architectures for retrieving solutions of Italian crossword clues by learning a shared latent space between clues and solutions. Our experiments demonstrated that the

[7]We partially modified the implementation found at https://github.com/pncnmnp/Crossword-Solver.

Asymmetric Dual Encoder (ADE) architecture, with its independent encoders for clues and solutions, outperformed the Siamese Dual Encoder (SDE) in handling the nuanced and often non-literal relationships characteristic of crossword puzzles. Our results also highlighted the limitations of traditional retrieval-based approaches (e.g., clues-to-clues methods), particularly when testing their generalization towards neologisms' definitions. In contrast, our dual encoder-based models, especially the larger and multilingual MPNet-base and the monolingual IT5-base, exhibited signs of generalization across diverse linguistic settings, including newly coined terms and culturally specific references. This underscores the importance of leveraging rich distributed representations to model the complex interplay between clues and solutions.

In future work, it could be interesting to explore ensemble methods that combine traditional information retrieval approaches with dual encoder models, including clues-to-clues retrieval techniques, to leverage their complementary strengths. Training a cross-encoder reranker on top of retrieved candidate solutions may also prove beneficial, as it would enable the exploitation of contextual relationships between clues and solutions, an approach that is standard in retrieval-based systems. Moreover, conducting a detailed linguistic analysis of clues, examining categories, frequency distributions, and other properties, could provide deeper insights into their characteristics. Finally, extending the methodology toward an automatic completion system for crossword puzzle grids represents a promising direction for supporting full puzzle solving.

## Acknowledgments

This work has been supported by the FAIR - Future AI Research (PE00000013) project under the NRRP MUR program funded by the NextGenerationEU, the PRIN PNRR 2022 Project EKEEL - Empowering Knowledge Extraction to Empower Learners (P20227PEPK) and the XAI-CARE-PNRR-MAD-2022-12376692 project under the NRRP MUR program funded by the NextGenerationEU. 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.

## References

[1] C. Bosco, E. Ježek, M. Polignano, M. Sanguinetti, Preface to the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), in: Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), 2025.

[2] P. Basile, M. de Gemmis, P. Lops, G. Semeraro, Solving a complex language game by using knowledge-based word associations discovery, IEEE Transactions on Computational Intelligence and AI in Games 8 (2016) 13–26. doi:10.1109/TCIAIG.2014.2355859.

[3] R. Manna, M. P. di Buono, J. Monti, Riddle me this: Evaluating large language models in solving word-based games, in: C. Madge, J. Chamberlain, K. Fort, U. Kruschwitz, S. Lukin (Eds.), Proceedings of the 10th Workshop on Games and Natural Language Processing @ LREC-COLING 2024, ELRA and ICCL, Torino, Italia, 2024, pp. 97–106. URL: https://aclanthology.org/2024.games-1.11.

[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] 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/.

[6] 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.

[7] 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.

[8] 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/.

[9] 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/.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 8748–8763. URL: https://proceedings.mlr.press/v139/radford21a.html.

[15] F. Bianchi, G. Attanasio, R. Pisoni, S. Terragni, G. Sarti, D. Balestri, Contrastive language-image pre-training for the italian language, in: F. Boschetti, G. E. Lebani, B. Magnini, N. Novielli (Eds.), Proceedings of the 9th Italian Conference on Computational Linguistics, Venice, Italy, November 30 - December 2, 2023, volume 3596 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3596/paper9.pdf.

[16] D. Gillick, A. Presta, G. S. Tomar, End-to-end retrieval in continuous space, arXiv preprint arXiv:1811.08008 (2018).

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.

[18] Z. Dong, J. Ni, D. Bikel, E. Alfonseca, Y. Wang, C. Qu, I. Zitouni, Exploring dual encoder architectures for question answering, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 9414–9419. URL: https://aclanthology.org/2022.emnlp-main.640/. doi:10.18653/v1/2022.emnlp-main.640.

[19] 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.

[20] C. Ciaccio, A. Miaschi, F. Dell'Orletta, Evaluating lexical proficiency in neural language models, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025.

[21] 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/.

[22] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mT5: A massively multilingual pre-trained text-to-text transformer, in: Proceedings of the 2021 Con-

ference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 483–498. URL: https://aclanthology.org/2021.naacl-main.41. doi:10.18653/v1/2021.naacl-main.41.

[23] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, et al., Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, arXiv preprint arXiv:2412.13663 (2024).

[24] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410/. doi:10.18653/v1/D19-1410.

[25] L. Bonifacio, I. Campiotti, R. de Alencar Lotufo, R. F. Nogueira, mmarco: A multilingual version of MS MARCO passage ranking dataset, CoRR abs/2108.13897 (2021). URL: https://arxiv.org/abs/2108.13897. arXiv:2108.13897.

[26] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 4512–4525. URL: https://aclanthology.org/2020.emnlp-main.365/. doi:10.18653/v1/2020.emnlp-main.365.

[27] J. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, International Conference on Learning Representations (2021).

[28] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[29] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library, arXiv preprint arXiv:2401.08281 (2024).

[30] A. Zugarini, M. Ernandes, A multi-strategy approach to crossword clue answer retrieval and ranking, in: CLiC-it, 2021.

[31] A. Severyn, M. Nicosia, G. Barlacchi, A. Moschitti, Distributional neural networks for automatic resolution of crossword puzzles, in: C. Zong, M. Strube (Eds.), Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 199–204. URL: https://aclanthology.org/P15-2033/. doi:10.3115/v1/P15-2033.

[32] L. Edman, H. Schmid, A. Fraser, CUTE: Measuring LLMs' understanding of their tokens, 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. 3017–3026. URL: https://aclanthology.org/2024.emnlp-main.177/. doi:10.18653/v1/2024.emnlp-main.177.

[33] C. Ciaccio, M. Sartor, A. Miaschi, F. Dell'Orletta, Beyond the spelling miracle: Investigating substring awareness in character-blind language models, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025.

[34] L. de Moura, N. Bjørner, Z3: An efficient smt solver, in: C. R. Ramakrishnan, J. Rehof (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 337–340.

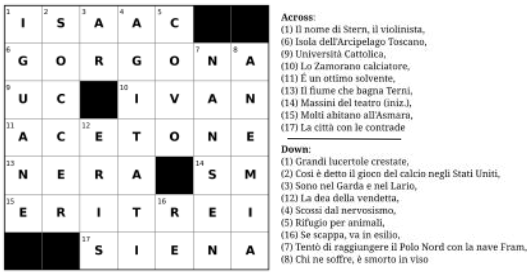


**Figure 4:** Solution for the autonomously solved crossword puzzle in Figure 1.

# A. Solved crossword puzzle

Figure 4 report the solution of the crossword presented in Figure 1.

# B. Further details on the hyperparameters

Both the siamese and asymmetric architectures were designed using PyTorch and the training was conducted on two Nvidia GeForce RTX 4090 GPUs. For the asymmetric architecture we leverage parallelization by assigning each encoder to a different GPU. Each model was trained

to produce representations of dimensionality equals to 768. We used the default betas and $\epsilon$ AdamW parameters. Table 5 reports the specific hyperparameters used with each model. Due to limited computational resources, we did not perform an extensive hyperparamters optimization, rather, we relied on the configurations suggested by the models creators. The maximum token length of the clues and solutions were set to respectively 64 and 16. The learnable temperature parameter $\tau$ was initialized to the equivalent of 0.07 from and clipped as done in CLIP paper. During batch generation, in order to avoid false negatives during hard batch mining, each batch cannot contain the same solution two or more times.

| Model | lr | weight decay |
|---|---|---|
| IT5-small | 5e-4 | 1e-3 |
| IT5-base | 5e-4 | 1e-3 |
| ModernBert | 2e-5 | 0.0 |
| ModernSBert | 2e-4 | 1e-3 |
| MPNet-base | 2e-4 | 1e-3 |

**Table 5**
Models specific hyperparameters.

During training, we kept track of the model's performance on the validation dataset and we picked the checkpoint with lowest validation loss.