# What Makes Cryptic Crosswords Challenging for LLMs?

**Abdelrahman Sadallah** **Daria Kotova** **Ekaterina Kochmar**
Department of Natural Language Processing, MBZUAI
{abdelrahman.sadallah, daria.kotova, ekaterina.kochmar}@mbzuai.ac.ae

## Abstract

Cryptic crosswords are puzzles that rely on general knowledge and the solver's ability to manipulate language on different levels, dealing with various types of wordplay. Previous research suggests that solving such puzzles is challenging even for modern NLP models, including Large Language Models (LLMs). However, there is little to no research on the reasons for their poor performance on this task. In this paper, we establish the benchmark results for three popular LLMs: `Gemma2`, `LLaMA3` and `ChatGPT`, showing that their performance on this task is still significantly below that of humans. We also investigate why these models struggle to achieve superior performance. We release our code and introduced datasets at https://github.com/bodasadallah/decrypting-crosswords.

## 1 Introduction

A cryptic crossword is a type of crossword puzzle known for its enigmatic clues (Friedlander and Fine, 2016). Unlike standard crossword puzzles, where clues are straightforward definitions or synonyms of the answers, cryptic crosswords involve wordplay, riddles, and cleverly disguised hints that make solving them more challenging (Moorey, 2018). Figure 1 shows an example of such a puzzle.

To solve a cryptic clue, one must not only apply generic rules in the specific context of the clue but also use domain-specific knowledge to produce a reasonable answer. Therefore, tackling cryptic crosswords with modern NLP methods provides a novel and interesting challenge. It has been shown that NLP models' performance is far from that of humans: Rozner et al. (2021) and Efrat et al. (2021) report an accuracy of 7.3% and 8.6% for rule- and transformer-based models. Sadallah et al. (2024) and Saha et al. (2024) show similarly low results for LLMs. In contrast, expert human solvers achieve 99% accuracy and self-proclaimed amateurs reach 74% (Friedlander and Fine, 2009, 2020), however, there are still no official statistics for average human performance.

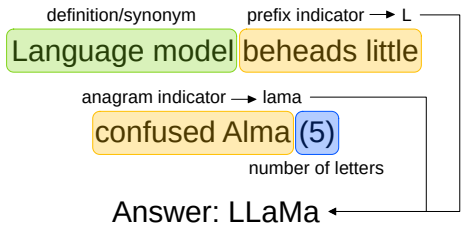


Figure 1: An example of a cryptic clue: number 5 at the end of the clue denotes the number of characters in the answer and is called **enumeration**. The **definition** part here is *language model*, with the rest being the **wordplay** part. *Beheads* or similar words point to the first letters of the next word, while *confused* (as well as *mixed up*, etc.) is likely to indicate an anagram. As we should look for a language model's name that starts with the letter *l* plus an anagram of *Alma* and consists of 5 letters, the answer here is *LLaMA*.

Typically, a cryptic clue can be divided into two parts: the **definition** and the **wordplay** (see Figure 1). The definition consists of one or more words in the clue that can be used interchangeably with the answer, and it usually appears either at the beginning or at the end of the clue. The wordplay can take many forms: the most popular ones include *anagrams*, *hidden words*, and *double definitions*, among others (see Table B1 for popular wordplay types and their examples).

Past approaches to solving cryptic clues range from rule-based models to traditional machine learning models like KNN (Rozner et al., 2021) and transformers like T5 (Rozner et al., 2021; Efrat et al., 2021). However, all these models achieve only modest accuracy on the task (see Section 2). The fact that LLMs can develop emergent capabilities (Wei et al., 2022) suggests that they may be able to solve cryptic puzzles if not on a par with human solvers, then at least somewhat successfully,



*Proceedings of the 31st International Conference on Computational Linguistics*, pages 5102–5114
January 19–24, 2025. 

however, our preliminary investigation shows that a zero-shot, naive approach to evaluating LLMs yields very low accuracy. Recently, Sadallah et al. (2024) and Saha et al. (2024) have evaluated modern LLMs on the task and showed that using certain prompting techniques can help push the limits of LLMs on this task, yet they are still far behind human experts' performance.

In this work, we focus on the interpretability of the LLM performance on the task of cryptic crossword solving and analyze which aspects of the task cause models to struggle most. Therefore, we evaluate LLMs solving one cryptic clue at a time rather than in a grid-eliminating information flow from the rest of the grid, in contrast to previous work like that of Saha et al. (2024). We focus on three main areas of the models' reasoning: (1) we explore whether they can extract the definition part of the clue; (2) we test the models' ability to identify the wordplay type in prompts containing varying amount of information; and (3) we test the models' internal reasoning by asking them to explain how they arrived at the answer.

Our main contributions are as follows: **(1)** We explore the general abilities of LLMs on the challenging task of solving cryptic crosswords using simple prompting strategies, each with a different amount of information embedded into the prompts; **(2)** We investigate models' understanding of the task by addressing three auxiliary tasks; **(3)** To facilitate reproducibility of our results and follow-up experiments, we introduce a small new dataset annotated with wordplay type labels,[1] and release all our data and code.[2]

## 2 Related Work

Although prior work looked into wordplay (Luo et al., 2019; He et al., 2019; Ermakova et al., 2023) and traditional crosswords (Littman et al., 2002; Zugarini et al., 2023), much less attention has been paid to cryptic crosswords. The early work of Deits (2015) achieved 7.3% accuracy on the task with a rule-based solver,[3] which applied hand-crafted probabilistic context-free grammar to generate all possible syntactic structures for clue words. Following this, Efrat et al. (2021) introduced `Cryptonite`, a dataset of 523,114 cryptic clues collected from *The Times* and *The Telegraph*. They fine-tuned a T5 (Raffel et al., 2023) model, which helped set the benchmark accuracy for Transformer models at 7.6%. Similarly, Rozner et al. (2021) introduced a dataset extracted from *The Guardian* and used a curriculum learning approach (Soviany et al., 2021), which involved training a model on simpler tasks before progressing to more complex compositional clues. This increased the performance to 21.8%.

Recently, Sadallah et al. (2024) and Saha et al. (2024) have evaluated multiple LLMs on the task of solving cryptic crossword puzzles using a range of prompting techniques, including zero and few-shot learning. While Sadallah et al. (2024) also explicitly fine-tune open-source LLMs for this task, Saha et al. (2024) use a combination of chain-of-thought (CoT) (Wei et al., 2023) and self-consistency (SC) (Wang et al., 2023) techniques, and achieve an accuracy score of 20.85% with GPT4-turbo (OpenAI et al., 2024). Both conclude that LLMs' performance on this task is still far from that of human experts. However, neither work further analyzes the models' behavior or why they struggle with this task.

## 3 Data

### 3.1 *The Guardian* dataset

In our experiments, we primarily use the dataset introduced by Rozner et al. (2021), which was extracted from *The Guardian*. Most previous models were tested on this dataset, so we have chosen it for comparison purposes as well. In total, the dataset contains 142,380 clues. Rozner et al. (2021) introduced two different splits for it: *naive (random)* and *word-initial disjoint*. We evaluate our models on the test subset of 28,476 examples from the *naive (random)* split, as it has more diverse examples than the other split.

### 3.2 *Times for the Times* dataset

To test models' performance across datasets, we use the data collected by George Ho,[4] where every clue has a marked definition. The original dataset contains around 600k clues from many sources, which would result in extremely expensive experimentation with LLMs. For our experiments, we have sampled 1,000 representative examples[5] col-

[1] https://huggingface.co/datasets/boda/small_explanatory_dataset

[2] https://github.com/bodasadallah/decrypting-crosswords

[3] https://github.com/rdeits/cryptics

[4] https://cryptics.georgeho.org/

[5] https://huggingface.co/datasets/boda/times_for_the_times_sampled

lected from the *Times for the Times* blog.[6] We ensure that the distribution of these examples, with respect to the number of words in the definition and their position in the clue, is similar to the full dataset and rely on the available definitions to estimate how well our models understand what the definition is. Additionally, this information helps investigate whether including the definition explicitly aids the models in solving the clues.

### 3.3 Small explanatory dataset

Unfortunately, there is no large-scale dataset that contains information about the wordplay types of the clues. To investigate whether our models can detect wordplay types, we have annotated 200 examples from the additional dataset (see Section 3.2), including 40 clues for each major wordplay type (*anagram*, *assemblage*, *container*, *hidden word*, and *double definition* – see Table B1 for examples).

## 4 Methodology

### 4.1 Zero-shot setup

**Base prompt** We begin by defining a simple prompt (see Figure E1) that only includes the minimal information required to solve the task. We include the line "you are a cryptic crosswords expert", as it has been shown that this phrase can help the model performance (Xu et al., 2023).

**All-inclusive prompt** In this prompt, we combine general information about cryptic crossword solving without adding examples or CoT (Wei et al., 2023) (see Figure E2). We include information about clue parts and their meanings. We also add information about the typical position of the definition in the clue. Finally, our preliminary experiments suggest that LLMs often struggle to understand the constraints of the answer length mentioned in the clue, so we explicitly tell the model that the number of letters in the answer is indicated in parentheses at the end of the clue. In addition, we experiment with solving a cryptic clue using the definition provided.

### 4.2 Dividing solution process into sub-tasks

Next, we investigate why the models struggle to solve the task. To do that, we design experiments to test the models' ability to (1) extract definition word(s) from the clue, (2) detect the wordplay type with varying levels of information, and (3) explain the solution process given the clue and the answer.

## 5 Experiments and Discussion

We choose two of the most recent and popular open-source LLMs, `Gemma2` (Gemma et al., 2024) and `LLaMA3` (Grattafiori et al., 2024), and one closed-source model, `ChatGPT` (OpenAI, 2021). The details are provided in Appendix A, and the results in Table 1.

### 5.1 Cryptic clue solving

The first four rows of Table 1 show the models' accuracy in solving cryptic clues on two different datasets for two different prompts. We can see that `ChatGPT` outperforms the open-source models. Also, we can conclude that providing the models with the definition improves their performance. To put these results into perspective, in Table 2, we compare our results with those obtained by Rozner et al. (2021). We do not compare to the results from Saha et al. (2024) because they are reported on a different subset of the dataset from Rozner et al. (2021). We observe that using `ChatGPT` in a zero-shot setting achieves results comparable to (but still lower than) those of T5 fine-tuning. One important thing to note is that Rozner et al. (2021) explicitly fine-tuned models on the task, while the models we used are general LLMs that were pre-trained on the generic language modeling task.

### 5.2 Understanding various aspects of the task

#### 5.2.1 Definition extraction

We ask the models to extract the definition part of the clue with the prompt illustrated in Figure E3. We specify that the definition should be a synonym for the answer but do not indicate that the definition usually appears at the beginning or end of the clue. All models show higher results in the definition extraction task. One reason for this could be that the definition is explicitly included in the clue, making the task a matter of repeating part of the clue, which is generally easier than generating new words as an answer.

#### 5.2.2 Wordplay detection

**Determining the wordplay type** We identify five major types of wordplay listed in Table B1. Then we investigate if our models could identify the wordplay type from the clues. Usually, professional solvers note indicator words that relate the clue to one type or another: for example, *confused,*

[6] `https://times-xwd-times.livejournal.com/`

| Task | Number of examples | Info / Prompt | Accuracy LLaMA3 | ChatGPT | Gemma2 |
|---|---|---|---|---|---|
| Cryptic Clue Solution | 28476 | base prompt | 2.2 | 10.9 | 4.8 |
| Cryptic Clue Solution | 28476 | all inclusive prompt | 2.1 | **11.4** | 2.4 |
| Cryptic Clue Solution | 1000 | all inclusive prompt | 3.3 | 13.4 | 5.3 |
| Cryptic Clue Solution | 1000 | ~ + definition | 3.8 | **16.2** | 7.0 |
| Definition Extraction | 1000 | definition extraction | 19.3 | **41.2** | 21.8 |
| Wordplay Type Detection | 200 | wordplay types | 20.0 | 42.5 | 33.5 |
| Wordplay Type Detection | 200 | ~ + explanation + ex. | 23.0 | 43.5 | 39.0 |
| Wordplay Type Detection | 200 | ~ + clue answer | 23.5 | **44.5** | 43.5 |

Table 1: The summary of the results obtained in our experiments on the `naive (random)` (first two rows), `Times for the Times` (rows 3-5), and `small explanatory` (last three rows) datasets. Best results are highlighted in bold.

*mixed up*, and *mad* usually indicate anagrams. To test the models' ability to identify the wordplay type, we design three experiments that gradually add information to the prompt. The specific design of the experiments is described in the Appendix C.

The results show that adding the definition for the wordplay and providing a model with the answer do not significantly improve the model's ability to extract the wordplay type except for Gemma, which has a performance increase of 10%. LLaMA3 only predicted one wordplay type (*hidden word*) using the 'wordplay types' prompt (see Figure E4), but providing more information in the other prompts helped the model predict other types. We hypothesize that a potential reason for LLaMA3's behavior is that the model seems to attend more to the task prompt than the clue itself.

We acknowledge that the small dataset size might constrain our ability to draw definitive conclusions. However, an important observation is that all 3 models over-predict some types (*anagram* and *hidden word*) while under-predicting others (*assemblage*). We include the full analysis with the models' confusion matrices on the most informative prompt shown in Figure E6 in Appendix C.

### 5.2.3 Explanation extraction

Finally, we ask the models to explain the solution, given the clue and the answer. Our analysis of the models' answers shows that: (1) All the models follow some kind of structure in their explanations, breaking the clue into parts of one to three words; however, this separation often does not seem to make sense, as it may combine both definition and wordplay parts together or use words that do not interact with each other. (2) LLaMA3 does not mention any wordplay operations and only works at a synonym level, which is insufficient for solving the clues. (3) Gemma shows the knowledge of some operation types (such as anagram and even homophones-related operations) but applies it incorrectly. (4) ChatGPT recognizes that something should be done with the characters and words in the clue and sometimes even gets it right, for example, suggesting taking an anagram of a given word or putting together words in an assemblage clue; however, it does not properly "understand" the procedure. For instance, one of the ChatGPT's outputs is: *rearranging the letters of "pan" and adding "to cook cheese" results in "parmesan"*. This statement is incorrect, as one cannot get "parmesan" from the letters in "pan" and "to cook cheese." (5) The easiest type to generate sensible explanations for are clues for the *double definition* type, where both parts of the clue are synonymous with the answer – this aligns with how base LLMs were trained.

| Model | Accuracy |
|---|---|
| LLaMA3 (best) | 2.2 |
| Gemma2 (best) | 4.8 |
| ChatGPT (best) | 11.4 |
| Rule-based | 7.3 |
| T5 fine-tuned | 16.3 |
| T5 fine-tuned + curriculum | 21.8 |

Table 2: Comparison with previous results: a rule-based method of Deits (2015) and the T5-based approach of Rozner et al. (2021).

## 6 Conclusions and Future Work

In this work, we have focused on studying the inner workings of LLMs while solving cryptic crosswords rather than trying to improve their performance on this task. We began by evaluating the models under a zero-shot setting and then tried to gain insights into their understanding of cryptic clues through auxiliary tasks. The results suggest that although `ChatGPT` model overall outperforms open-source LLMs, solving cryptic crosswords remains a very challenging task for all tested LLMs, with a significant room for improvement. In addition, we conclude that splitting the task into subtasks helps the models to some extent, which indicates that models cannot break down the composite task by themselves. The performance of the models on the chosen subtasks still remains unsatisfactory: the models struggle to identify the definition and the wordplay type.

We believe the performance can be improved in future work using several possible research directions. Firstly, promising avenues for research in this area are chain-of-thought (Wei et al., 2023) and tree-of-thought (Yao et al., 2023) techniques. This is motivated by our current results that suggest that splitting the task into simpler subtasks helps improve the model performance: specifically, CoT-based methods can teach models how to arrive at the solution step-by-step by splitting the original complex task into such multiple simpler subtasks. Secondly, given the considerable performance increase achieved using curriculum learning with T5 (Rozner et al., 2021), we consider this direction worth exploring with LLMs as well. Finally, approaches such as a mixture of experts (Jacobs et al., 1991; Gale et al., 2022) used to train open-source models like Mixtral (Jiang et al., 2024) can be applied to the task, as models may develop expert layers specializing in separate wordplay types.

## Limitations

**Limited set of LLMs experimented with** Experiments with an extensive set of state-of-the-art LLMs can get quite expensive. Due to budget limitations, we have been selective in terms of the LLMs that we use in this study. Specifically, we chose only a few of the most popular open-source and closed-source LLMs. We believe that the obtained results shed light on the current LLMs' capabilities on this task. However, we acknowledge that the set of LLMs we tested here is limited, and our results cannot be extrapolated to other LLMs. In addition, in many experiments, we have observed that minor changes in settings do not bring substantial improvement to the results. This motivated us to perform only a limited set of experiments with the chosen models, as elaborated in the paper.

**Limitations of the datasets size** Some datasets we used are not large in terms of the number of examples. The main reason for this is the lack of existing datasets with rich annotation, so we had to create one such dataset ourselves. We acknowledge that the results obtained on a larger dataset may be more reliable; however, we believe that the results reported here already provide us with useful insights.

**Closeness to the real-world scenario** In this work, we have focused on solving one clue at a time. In the real-world scenario, human solvers encounter twenty to thirty clues in one grid. Solving one clue usually reveals letters of the other answers, which can be quite helpful in the solution process. In contrast, our goal is to investigate LLMs' abilities in cryptic crossword clue interpretation, and we do not try to solve the whole grid.

**Dangers of data contamination** Finally, we observe in our experiments that `ChatGPT` outperforms the open-source models. We acknowledge that we lack information about its training setup, as `ChatGPT` is a proprietary model, and therefore, we cannot guarantee that this model's training data is uncontaminated; in other words, it is not entirely clear whether the model could have been exposed to any of the crossword clues during its training. However, we note that all LLMs still struggle to solve cryptic clues, showing that even if some contamination took place, the models do not seem to be able to memorize and simply reproduce the answers from previously seen clues. As a side note, human experts also get exposed to a lot of clues in their practice, and their performance on the task is still much higher than that of LLMs.

## Ethics Statement

We foresee no serious ethical implications from this study.

## Acknowledgments

We are grateful to the campus supercomputing center at MBZUAI for providing resources for this research.

## References


Tri Dao. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. *Preprint*, arXiv:2307.08691.

Robin Deits. 2015. Cryptics.

Avia Efrat, Uri Shaham, Dan Kilman, and Omer Levy. 2021. Cryptonite: A Cryptic Crossword Benchmark for Extreme Ambiguity in Language. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4186–4192, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liana Ermakova, Anne-Gwenn Bosser, Adam Jatowt, and Tristan Miller. 2023. The JOKER Corpus: English-French Parallel Data for Multilingual Wordplay Recognition. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2796–2806, New York, NY, USA. Association for Computing Machinery.

Kathryn J. Friedlander and Philip A. Fine. 2016. The grounded expertise components approach in the novel area of cryptic crossword solving. *Frontiers*.

Kathryn J. Friedlander and Philip A. Fine. 2020. Fluid Intelligence is Key to Successful Cryptic Crossword Solving. *Journal of Expertise*, 3(2):101–132.

KJ Friedlander and PA Fine. 2009. Expertise in cryptic crossword performance: an exploratory survey. In *Proceedings of the International Symposium on Performance Science, Auckland, eds A. Williamon, S. Pretty, and R. Buck (Utrecht: European Association of Conservatoires (AEC))*, pages 279–284.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. 2022. MegaBlocks: Efficient Sparse Training with Mixture-of-Experts. *Preprint*, arXiv:2211.15841.

Team Gemma, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open Models Based on Gemini Research and Technology. *Preprint*, arXiv:2403.08295.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Va-

sic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

He He, Nanyun Peng, and Percy Liang. 2019. Pun Generation with Surprise. In *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)*, pages 1734–1744, Minneapolis, Minnesota. Association for Computational Linguistics.

Robert Jacobs, Michael Jordan, Steven Nowlan, and Geoffrey Hinton. 1991. Adaptive Mixture of Local Experts. *Neural Computation*, 3:78–88.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *Preprint*, arXiv:2401.04088.

Michael L. Littman, Greg A. Keim, and Noam Shazeer. 2002. A probabilistic approach to solving crossword puzzles. *Artificial Intelligence*, 134(1):23–55.

Fuli Luo, Shunyao Li, Pengcheng Yang, Lei Li, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Pun-GAN: Generative Adversarial Network for Pun Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3388–3393, Hong Kong, China. Association for Computational Linguistics.

Tim Moorey. 2018. *How to Crack Cryptic Crosswords*. Collins Puzzles.

OpenAI. 2021. ChatGPT. Technical report, OpenAI.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Bar-

ret Zoph. 2024. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Preprint*, arXiv:1910.10683.

Josh Rozner, Christopher Potts, and Kyle Mahowald. 2021. Decrypting Cryptic Crosswords: Semantically Complex Wordplay Puzzles as a Target for NLP. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11409–11421.

Abdelrahman "Boda" Sadallah, Daria Kotova, and Ekaterina Kochmar. 2024. Are LLMs Good Cryptic Crossword Solvers? *Preprint*, arXiv:2403.12094.

Soumadeep Saha, Sutanoya Chakraborty, Saptarshi Saha, and Utpal Garain. 2024. Language Models are Crossword Solvers. *Preprint*, arXiv:2406.09043.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and N. Sebe. 2021. Curriculum Learning: A Survey. *International Journal of Computer Vision*, 130:1526 – 1565.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *Preprint*, arXiv:2206.07682.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Preprint*, arXiv:2201.11903.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. ExpertPrompting: Instructing Large Language Models to be Distinguished Experts. *Preprint*, arXiv:2305.14688.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *Preprint*, arXiv:2305.10601.

Andrea Zugarini, Thomas Röthenbacher, Kai Klede, Marco Ernandes, Björn Eskofier, and Dario Zanca. 2023. Die Rätselrevolution: Automated German Crossword Solving. In *Proceedings of the 9th Italian Conference on Computational Linguistics*.

## A Implementation details

For the open-source models, we chose to use the instruct-tuned versions (`gemma-2-9b-it` and `Meta-Llama-3-8B-Instruct`) as they give better results and are more suitable for the task. For all three models, we generated the outputs using zero temperature (greedy sampling) as it showed slightly better results in our preliminary experiments. To get the highest possible performance from the open-source models, we used the models in full precision, without doing any quantization. Finally, we used FlashAttention-2 (Dao, 2023) for faster inference.

We used `gpt3.5-turbo` with near-zero temperature and $top\_p$ of 1e-9. However, as it gave slightly different results across different runs, we ran it 3 times and reported the average of the results from these runs.

We do not use any post-processing of the models' answers.

## B Wordplay types

Common wordplay types are listed in Table B1 with examples[7] and explanations. We identify 5 main types: anagram, assemblage, container, hidden word, and double definition.

## C Worplay type detection experiments

In the first experiment, we give the models the names of the five different wordplay types and ask them to predict which wordplay type the given clue belongs to (see Figure E4). We notice that `LLaMA3` fails to understand the task and produces only one type for all examples, which suggests that the model does not analyze the given clues thoroughly. Next, we experiment with providing the models with the explanations and one example for each wordplay type (Figure E6). Finally, we add the answer for each clue to test whether the models can infer information about the wordplay types from the answer (Figure E7).

Next, we analyze the models' predictions using the most informative prompt (Figure E7). For `LLaMA3`, the most frequently predicted wordplay type is "hidden word" (100+ samples) and "container" (55 samples), and never predicted "double definition" or "assemblage." The confusion matrix is shown in Figure C1.

`Gemma` most frequently predicted "anagram" (101 samples) and "hidden word" (43 times) and never

[7] Examples are taken from `https://crypticshewrote.wordpress.com/explanations/`

| Type | Example Clue | Answer |
|---|---|---|
| **Anagram:** certain words or letters must be jumbled to form an entirely new term. | Never upset **a Sci Fi writer** (5) | Verne |
| **Assemblage:** the answer is broken into its component parts and the hint makes references to these in a sequence. | **Bitter** initially, but extremely enjoyable refreshment (4) | Beer |
| **Container:** the answer is broken down into different parts, with one part embedded within another. | **The family member** put us in the money (6) | Cousin |
| **Hidden word:** the answer will be hidden within one or multiple words within the provided phrase. | **Confront them** in the tobacco store (6) | Accost |
| **Double definition:** contains two meanings of the same word. | **In which you'd place the photo** of the **NZ author** (5) | Frame |

Table B1: Examples of common wordplay types. The definition part is bolded.

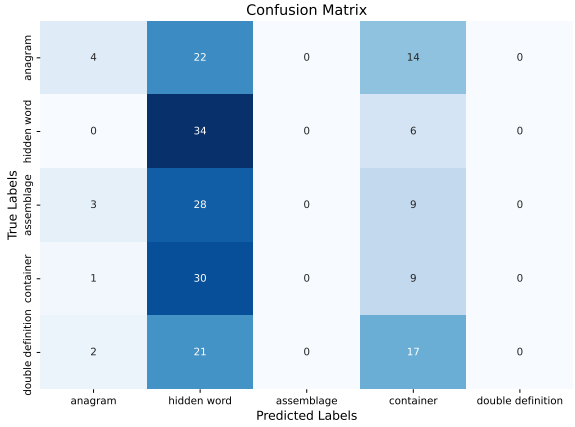


Figure C1: Confusion matrix for LLaMA3 on wordplay type prediction using the most informative prompt E7.

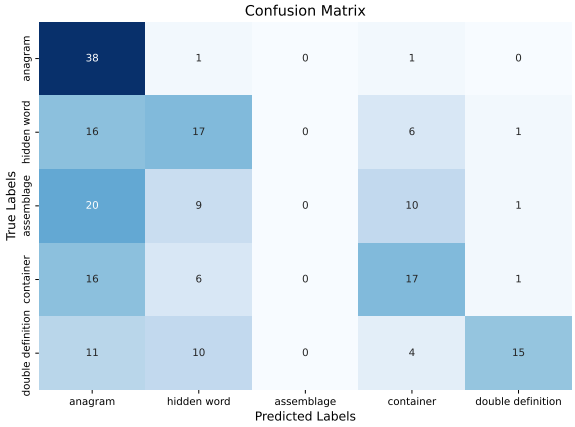


Figure C2: Confusion matrix for Gemma on wordplay type prediction using the most informative prompt E7.

predicted "assemblage." Its confusion matrix is shown in Figure C2.

ChatGPT most frequently predicted "container" (97 times) and "anagram" (46 times) and predicted "assemblage" only 3 times. Its confusion matrix is shown in Figure C3. What is interesting here is that the model sometimes predicted types different from the specified ones.

## D Data sources

In the text of the paper, we mention several sources of cryptic crosswords:

1. *The Times*[8]
2. *Telegraph*[9]

[8] https://www.thetimes.co.uk/puzzleclub/crosswordclub/home/crossword-cryptic

[9] https://puzzles.telegraph.co.uk/crossword-puzzles/cryptic-crossword

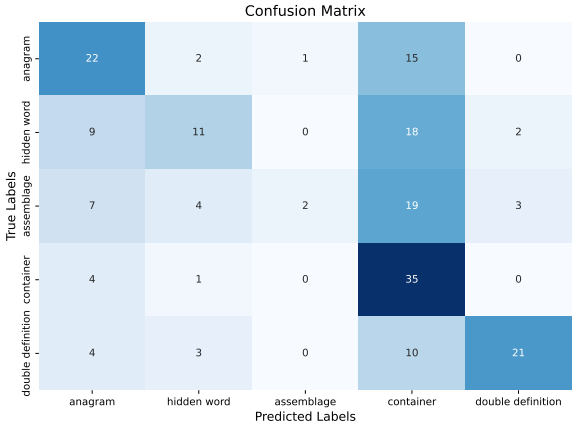


Figure C3: Confusion matrix for ChatGPT on wordplay type prediction using the most informative prompt E7.

3. *The Guardian*[10]
4. *Times for the Times* blog[11]

We do not parse their data specifically but rather use already prepared datasets or samples from them.

## E Prompts

We present all the prompts we used in this section: see Figures E1 to E7.

```
You are a cryptic crossword expert.
You are given a clue for a cryptic
crossword. Output only the answer.
clue:
{clue}
output:
{output}
```

Figure E1: Base prompt.

```
You are a cryptic crossword expert.
The cryptic clue consists of a
definition and a wordplay.
The definition is a synonym of the
answer and usually comes at the
beginning or the end of the clue.
The wordplay gives some instructions
on how to get to the answer in
another (less literal) way.
The number/s in the parentheses at
the end of the clue indicates the
number of letters in the answer.
Extract the definiton and the wordplay
in the clue, and use them to solve the
clue. Finally, output the answer on
this format:
Answer: <answer>,
Clue:
{clue}
```

Figure E2: All inclusive prompt.

```
You are a cryptic crossword expert. I
will give you a cryptic clue. Every clue
has two parts: a definition and a
wordplay. The definition is a synonym
of the clue's answer. Extract the
definition word/s from this clue. Only
output the definition.
Clue: {clue}
Definition:
```

Figure E3: Prompt for the definition extraction.

```
You are a cryptic crosswords expert. I
will give you a clue. Every clue has
two parts: a definition and wordplay.
Definition is a synonym of the answer.
Wordplay is the rest of the clue.
Please extract the wordplay type for
this clue.
Here is a list of all possible wordplay
types: anagram, hidden word, double
definition, container, assemblage.
Only output the wordplay type.
Clue: {clue}
Output:
```

Figure E4: Prompt for the wordplay type classification.

```
You are a cryptic crossword expert.
The cryptic clue consists of a
definition and a wordplay.
The definition is a synonym of the
answer and usually comes at the
beginning or the end of the clue.
The wordplay gives some instructions
on how to get to the answer in
another (less literal) way.
The number/s in the parentheses at
the end of the clue indicates the
number of letters in the answer.
Use the given definition, and extract
the wordplay in the clue, and use
them to solve the clue. Finally, output
the answer on this format:
Answer: <answer>,
Clue:
{clue}
Definition:
{definition}
```

Figure E5: All inclusive prompt with included definition.

[10] https://www.theguardian.com/crosswords/series/cryptic

[11] https://times-xwd-times.livejournal.com/

You are a cryptic crosswords expert. I will give you a clue. As you know, every clue has two parts: a definition and wordplay. Please extract the wordplay type from this clue.
Here is a list of all possible wordplay types, and their descriptions:
- anagram: An anagram is a word (or words) that, when rearranged, forms a different word or phrase.
  Example: Ms Reagan is upset by the executives (8)
  The answer: Managers

- hidden word: The answer is found in the clue itself, amongst other words.
  Example: Confront them in the tobacco store (6)
  The answer: Accost

- double definition: Clues contain two meanings of the same word. The words may be pronounced differently, but must be spelt the same.
  Example: Footwear for pack animals (5)
  The answer: Mules

- container: One word is placed inside another (or outside another) to get the answer.
  Example: Curse about the Maori jumper (7)
  The answer: Sweater

- assemblage: The answer is broken up into smaller parts and each syllable or part is given a separate clue. These separate clues are then put together into one clue.
  Example: Brash gets a Prime Minister employment, but it's drudgery (6,4)
  The answer: Donkey work
Only output the wordplay type.
Clue: {clue}
Output:

Figure E6: Prompt for the wordplay type classification with examples for each wordplay type.

You are a cryptic crosswords expert. I will give you a clue. As you know, every clue has two parts: a definition and wordplay. Please extract the wordplay type from this clue.
Here is a list of all possible wordplay types, and their descriptions:
- anagram: An anagram is a word (or words) that, when rearranged, forms a different word or phrase.
  Example: Ms Reagan is upset by the executives (8)
  The answer: Managers

- hidden word: The answer is found in the clue itself, amongst other words.
  Example: Confront them in the tobacco store (6)
  The answer: Accost

- double definition: Clues contain two meanings of the same word. The words may be pronounced differently, but must be spelt the same.
  Example: Footwear for pack animals (5)
  The answer: Mules

- container: One word is placed inside another (or outside another) to get the answer.
  Example: Curse about the Maori jumper (7)
  The answer: Sweater

- assemblage: The answer is broken up into smaller parts and each syllable or part is given a separate clue. These separate clues are then put together into one clue.
  Example: Brash gets a Prime Minister employment, but it's drudgery (6,4)
  The answer: Donkey work
Only output the wordplay type.
Clue: {clue}
The answer: {ans}
Output:

Figure E7: Prompt for the wordplay type classification with examples for each wordplay type. Here we also add the answer for the clue.