Decrypting Cryptic Crossword Clues

Mitanshu Gada Soon Song Cheok Prathyusha Mallela
Oregon State University
{gadam, cheoks, mallelap}@oregonstate.edu

Abstract

We present a new method for solving cryptic crosswords using a pre-trained T5 model with adapters is proposed. The problem addressed is the difficulty of solving cryptic crossword puzzles, which require a deep understanding of language and wordplay. The proposed approach adapts the T5 model for specific types of clues and achieves state-of-the-art performance on a benchmark dataset of cryptic crossword clues. The results demonstrate the potential of pre-trained language models with adapters for solving complex language-based problems. We used the cryptic clue dataset of 400k (cryptonite) provided by hugging face. We evaluated our model performance for various epochs, our train accuracy for classifier was a 14.38 percent on an average for all 20 epochs.The metrics we are using to measure model performance are top1, top10, correct length and correct word counts. The results for these metrics are 0.088% for top1, 0.0879% for top10, 4.959% for correct length, and 37.3896% for correct word counts.

1. INTRODUCTION

Our project Decrypting Cryptic Crossword Clues focuses on solving cryptic clues of various types, including anagram, charade, container, reversal, deletion, double definition, hidden word, and homophone. Cryptic clues are difficult for current NLP systems to solve as they consist of a definition and a wordplay cipher that requires character-level manipulation. Expert human solvers rely on linguistic, world, and domain knowledge to solve cryptics, making them a challenging benchmark for NLP systems. We were inspired by the paper titled ”Decrypting Cryptic Crosswords: Semantically Complex Wordplay Puzzles as a Target for NLP” and the authors’ use of a curriculum approach by J. Rozner et al. (November 2021) to improve NLP models’ performance on a dataset of cryptic clues. However, we believed that using a rule-based approach to classify the clues prior to fine-tuning T5 small model would enhance T5’s ability to solve the clue. We based our approach on the paper ”Cryptonite: A Cryptic Crossword Benchmark for Extreme Ambiguity in Language” (Efrat et al., 2021), which discusses the authors’ approach of training Hugging Face’s Cryptonite dataset (https://huggingface.co/datasets/cryptonite) and the accuracy of the T5 model in solving them compared to a rule-based system. Cryptonite is a linguistically complex and naturally sourced large-scale dataset of cryptic clues that requires disambiguating semantic, syntactic, and phonetic wordplays, as well as world knowledge. Even experienced solvers find cryptic clues challenging, and only top-tier experts can solve them with almost 100The anagram clue is the clue where a word or a phrase in the clue is expected to be rearranged. This is one of the most common clues in the dataset and this is visible in the classification of the clue types too. An example of this clue would be: Cook a chop another way (8) with the answer “Poach”. Here, a chop is the phrase that needs to be unscrambled as suggested by the pointer word “another”. The word must be synonymous with cook. The container clues are clues wherein either a word is placed within another word or a word encapsulates another word. Pointers to container clues are often the same as those for hidden word clues. Example of the container clue would be: Latest into the sack is severely reprimanded (7) with answer “blasted”, the indicator word is “into”, the synonym of latest is late, and late in sack would give late inside of bed; which would be “blasted”. The reversal clue: This type of clue requires you to reverse a word or group of letters to create the answer. For example: ”Doctor ordered a retreat” (reverse ”retreat” to get ”TERRIER”) ”Tall stories by child” (reverse ”child” to get ”DICED”) The deletion clue: This type of clue requires you to remove one or more letters from a word to create the answer. For example: ”Admission price without one Greek letter” (remove ”one” from ”price” and add ”beta” to get ”PRICEY”) ”Hide is almost cut” (remove the last letter of ”hide” to get ”HID”) The homophone clue: This type of clue requires you to find a word that sounds like the answer. For example: ”Musical instrument sounds like a snake” (the answer is ”BASS,” which sounds like ”viper’s base”) ”Group of actors sounds like a vegetable” (the answer is ”CAST,” which sounds like ”chard stalks”) The charade clue: This type of clue requires you to combine two or more words to create the answer. For example: ”A game played with a king and a horse” (combine ”chess” and ”knight” to get ”CHESTNUT”) ”Breed of dog from a river in Spain” (combine ”spaniel” and ”Ebro” to get ”SPRINGER”) The hidden word clue: This type of clue requires you to find a word hidden within a larger word or phrase. For example: ”No prize, but it’s hidden” (the answer is ”CONSOLATION,” which is hidden within the clue) ”You’ll find a plant in this address” (the answer is ”FLAT,” which is hidden within ”this address”) The double definition clue: This type of clue provides two separate definitions for the answer. For example: ”Famous boxer and container” (the answer is ”ALI,” which can refer to Muhammad Ali and also means a type of metal container) ”A type of tree and a ship’s pole” (the answer is ”MAST,” which can refer to a tree’s trunk and also a vertical pole on a ship) In our approach we have a rule-based clue classifier, which classifies the clues based on the indicators present in the indicator database/lists for each clue type. In the second approach, we are using a BiLSTM, which is a recurrent neural network architecture that processes input sequences bidirectionally, capturing both forward and backward context. This was chosen as one of the ways as a BiLSTM captures the preceding and succeeding words in the sequence which can help in text classification. The single layer BiLSTM classifier that was used creates a classifier.pt model which is further used in fine-tuning the T5 small state-of-art model for training purposes. After fine-tuning, we ran the metrics application to review our performance. The metrics we are using to measure model performance are top1, top10, correct length and correct word counts. The results for these metrics are 0.088

2. Related Work

Our project was inspired by two papers, ”Decrypting Cryptic Crosswords: Semantically Complex Wordplay Puzzles as a Target for NLP. 2021” [2] and ”Cryptonite: A Cryptic Crossword Benchmark for Extreme Ambiguity in Language” [1]. The authors of [2] proposed that natural language processing (NLP) can aid in solving cryptic crossword puzzles, which require the solver to decipher semantically complex wordplay clues. They suggested a set of NLP tasks that could be used to solve these clues and introduced a dataset of 7,000 cryptic crossword clues with their answers to demonstrate the feasibility of using NLP methods. The authors proposed that this work could lead to the development of more advanced crossword-solving tools and contribute to advancing NLP in understanding semantically complex language.

The authors of [1] proposed a deep learning method for solving cryptic crossword clues by fine-tuning a pre-trained T5 language model. They introduced a dataset of 30,000 cryptic crossword clues with their answers to train and evaluate their model, and compared its performance against human solvers and existing systems. The results showed that their model achieved high accuracy and outperformed the existing systems in solving cryptic crossword clues. However, a rule-based cryptic clue solver was found to be more efficient than their fine-tuned T5 model by a percent. Therefore, we decided to classify the clues using a rule-based classifier before training T5. Due to time constraints, we opted for a BiLSTM classifier to classify the data.

3. Methodology

We built two clue classification systems. The first is a rule-based classification system that checks indicators to classify the 400k cryptic clue naive split dataset from the cryptonite(https://huggingface.co/datasets/cryptonite) . This used a scoring mechanism which would predict the clue labels based on the highest indicator word match. This is the base version of the rule-based classifier that we chose to use. In our second approach to classifying the clues we used a single fully connected BiLSTM which would classify into 9 classes (outputs) of clues- anagram, reversal, deletion, homophone, container, double-definition, charade, hidden-word, and unclassified. This BiLSTM classifier is further used to train the adapters to fine-tune the T5 small model for our custom dataset.

Adapter Training: We chose to train adapters for different clue types as this would help us with fast adaptation to clues of a particular class. In short, different adapters perform different tasks. Overall, adapter design and training offer a flexible and efficient approach to adapting pre-trained models like T5 small to perform new tasks, while minimizing the need for additional training data and resources. We used prefix tuning adapters as it prepends weight vectors to the input, and T5 model is good at performing tasks with the task type is prepend to the sentence. For fine-tuning out T5 small we used the following steps: Adapter design - Adapter design involves identifying a small set of parameters in a pre-trained model (such as T5 small) that can be modified to perform a specific task. Instead of training a whole new model for a new task, the adapter approach enables the reuse of the pre-trained model and training only the small set of parameters that are task-specific. Here, we have identified the task to train clues of a particular class. Each of the classes has its own adapter. Adapter training - Adapter training is the process of training the adapter parameters using a small amount of task-specific data while keeping the rest of the pre-trained model frozen. This approach enables fast adaptation to new tasks with limited data and computational resources. Each of the clue types have their own adapters that are trained. Adapter tuning - Adapter tuning is the process of fine-tuning the entire pre-trained model, including the adapters, on the task-specific data. This approach allows the model to learn more about the specific task and potentially achieve even higher performance. We are using prefix tuning adapters on the clue dataset. The reason why we chose to use prefix tuning adapters is because the T5 model is very good at performing tasks that are pre-appended to the input task. To classify a clue type using a BiLSTM classifier, we first use the T5 tokenizer from Hugging Face to tokenize the input. This results in word IDs, which we then input into the classifier. The classifier was previously trained using labeled data and the model weights are stored in the classifier.pt file. By inputting the tokenized text into the BiLSTM classifier, we can classify the type of clue. Without this preprocessing and classification, the raw input would be difficult to interpret and analyze.

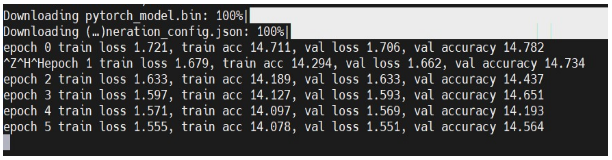
Metrics computation: In the overall implementation, once the BiLSTM classifier predicts the type of clue, the clue and its type are fed into the T5 model with adapters. The T5 model uses the clue type to activate the specific adapters that are relevant for that type of clue. The feed-forward of the T5 model will then include the parameters of the activated adapter. Once the T5 model generates the output, the T5 model tokenizers are used to decode the output. To check the performance of the T5 model, a metric is computed by generating the top ten beam search outputs for each input using the model. Then, the ten outputs for each input are combined and used to compute the metrics. This allows for a more robust evaluation of the model's performance.

The code is available at:

- BiLSTM, Adapter Training, and Metrics
- Rule-classifier with indicators
- Rule-based classified data
- BiLSTM Classifier
- Adapter Trained model (2000 epochs)
- Data used
- Algorithms used

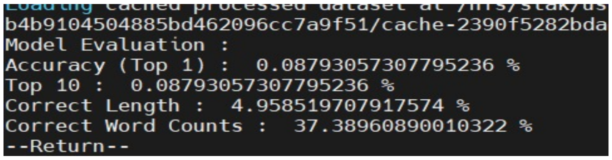
4. Results and Discussion

The low performance of the model could be attributed to various factors. One of the reasons could be an imbalance in the clue type data in the prepared dataset. Additionally, the rule engine and the BiLSTM classifier that labels the dataset may not be accurate, leading to incorrect labels in the training data. This can result in the classifier and adapters being trained with incorrect data, which ultimately leads to low performance. Lastly, the classifier may not be complex enough to learn how to accurately classify the clue type, leading to incorrect predictions and the use of wrong adapters to predict the clue answer.



```
Downloading pytorch_model.bin: 100%|
Downloading (…)neration_config.json: 100%|
epoch 0 train loss 1.721, train acc 14.711, val loss 1.706, val accuracy 14.782
^Z^H^Hepoch 1 train loss 1.679, train acc 14.294, val loss 1.662, val accuracy 14.734
epoch 2 train loss 1.633, train acc 14.189, val loss 1.633, val accuracy 14.437
epoch 3 train loss 1.597, train acc 14.127, val loss 1.593, val accuracy 14.651
epoch 4 train loss 1.571, train acc 14.097, val loss 1.569, val accuracy 14.193
epoch 5 train loss 1.555, train acc 14.078, val loss 1.551, val accuracy 14.564
```

Figure 1. BiLSTM Classifier Training



```
Loading cached processed dataset at /nfs/stak/us
b4b9104504885bd462096cc7a9f51/cache-2390f5282bda
Model Evaluation :
Accuracy (Top 1) :  0.08793057307795236 %
Top 10 :  0.08793057307795236 %
Correct Length :  4.958519707917574 %
Correct Word Counts :  37.38960890010322 %
--Return--
```

Figure 2. Metric Output

5. Limitations

The available compute resources were not enough to handle the requirements of the model, data, and metrics. Despite the advantages of using a High-Performance Computing (HPC) cluster for model training, the model itself occupied a significant amount of space. The code was frequently revised to enhance the performance of the trained model. The rule-based approach to clue classification was not utilized to train the T5 small model due to a lack of time. This was because the grammar, word-to-word relationships, and overall expression of the sentence needed to be comprehended in order to classify clues with expertise.

6. Future Scope

To improve the accuracy of our classifier for cryptic clues, we can explore more complex rule-based classification methods that take into account the grammar of the clues. We can then use T5 small's question and answer functionality to check whether our classification method is able to accurately predict the solutions to the clues. Additionally, we can experiment with using T5 base or T5 large models to further improve the accuracy of the output.

Another potential approach we can consider is adapter fusion, which involves combining multiple adapters together to improve performance on related tasks. While this is not currently incorporated in our system, we plan to explore this methodology in the future, as some cryptic clues may fall into one or more categories.

7. Acknowledgement

We would like to express our appreciation for the invaluable contributions of PyTorch, HuggingFace, and AdapterHub, without which our research would not have been possible. Our models were developed on top of these libraries, and we are grateful for their existence. We would also like to extend our thanks to the engineering department of Oregon State University for lending us GPU resources that enabled us to train, test, and store our models for future research. We are deeply grateful to Professor Stefan Lee for providing us with the opportunity to delve into NLP with deep learning concepts by helping us work on this project. Additionally, we would like to extend our special thanks to J. Rozner et al., Efrat et al., and Robin Deits for their research and work on developing systems that solve cryptic crossword clues, which inspired us to undertake this project.

8. Conclusion

We found that the performance of our model was not as high as we had anticipated, even though we decided to use a BiLSTM to improve the accuracy of classifying the cryptic crossword clues. It is important to note that we are still in the process of refining our approach, and this experiment provided us with valuable insights into the need for supervised learning to some extent in order to achieve success in the classification and solving of cryptic crossword clues.

9. References

[1] A. Efrat, U. Shaham, and D. K. O. Levy, “Cryptonite: A Cryptic Crossword Benchmark for Extreme Ambiguity in Language,” arXiv, Nov. 2021. doi: 10.48550/ARXIV.2103.01242

[2] Joshua Rozner, Christopher Potts, and Kyle Mahowald, Decrypting Cryptic Crosswords: Semantically Complex Wordplay Puzzles as a Target for NLP. 2021. [Online]. Available: //doi.org/10.48550/arXiv.2104.08620

[3] R. Deits, “A cryptic crossword clue solver ← - blog.robindeits.com,” A Cryptic Crossword Clue Solver. [Online]. Available: https://blog.robindeits.com/2013/02/11/a-cryptic-crossword-clue-solver/. [Accessed: 25-Mar-2023].