Deep Learning with Free-Text Rationale for Cryptic Crosswords

William Hill and **Steven James**
School of Computer Science and Applied Mathematics
University of the Witwatersrand, Johannesburg

Abstract

Cryptic Crosswords are puzzles characterised by the need to overcome extreme ambiguity in clues, to the point where even humans struggle to solve them. Thus they provide a ready-made environment for learning and evaluating the ability of Language Models to learn the nuances of natural language and perform complex disambiguation tasks. Owing to the recency of work concerning Cryptic Crosswords, no investigations into the effect of rationales on the answer accuracy have taken place, although they have had significant success on other NLP problems. We examine the use of a self-rationalising model, which simultaneously predicts the answer to a cryptic clue and an associated free-text rationale, on the answer accuracy. We use T5 as our base model and include Curriculum Learning in our training process. We find that there is additional predictive ability in the free-text rationales, but that our language models are unable to learn to produce the rationales with good-enough quality to exploit it. Thus, in certain cases, even though the overall accuracy remains low, the use of a self-rationalising model does lead to slight improvement[1].

Introduction

Cryptic Crosswords are puzzles, appearing in many major newspapers such as the *Times*, whereby a natural language clue is given, which has a single answer comprising of one or more words (i.e. a word or phrase). "Cryptic" refers to the fact that the clue's surface-level meaning is equivocal and requires one or more clever wordplays and world knowledge in order to discover the true answer, as demonstrated in Figure 1. Wordplay denotes any valid natural language disambiguation task, including phonetic, syntactic, and character-level manipulation (e.g. homophones, anagrams, synonyms).

Consequently, any Machine Learning model aiming to solve these clues requires robust knowledge of the English language and its nuances — just as any human attempting this challenge would need.

A significant amount of research has focused on storing and retrieving knowledge within the parameters of a Language Model via pretraining and achieving state-of-the-art results on common Natural Language Processing (NLP) tasks (Petroni et al. 2019; Talmor et al. 2020; Roberts, Raffel, and Shazeer 2020). The intricacies of the required knowledge for solving Cryptic Crosswords necessitate additional techniques, such as Curriculum Learning (Rozner, Potts, and Mahowald 2021), in order to acquire said knowledge more reliably.

However, solutions using current best practices (using a pretrained T5 model (Raffel et al. 2020), fine-tuned on Cryptic Crossword datasets) yielded inadequate state-of-the-art results for predicting the answers (Efrat et al. 2021; Rozner, Potts, and Mahowald 2021). Thus further work is needed on complex disambiguation tasks.

Specifically, previous work has shown that when additionally predicting the rationale behind a model's predicted label, the state-of-the-art label accuracy was improved upon or matched for common NLP tasks (Narang et al. 2020; Kumar and Talukdar 2020), suggesting that simultaneously predicting rationale can improve label prediction. This train of thought has not been attempted when considering Cryptic Crosswords and it is this topic on which we focus, with an emphasis on Free-Text Rationale[2].

Firstly, we address the question of whether simultaneously producing free-text rationale (with the answer prediction) actually improves the model's ability to correctly answer cryptic clues. We find that in the majority of experiments, simultaneously producing the rationale did not increase the model's accuracy. In fact, the accuracy decreased and, in some cases, quite significantly. However, the experiments where the accuracy increased still had a low accuracy.

Secondly, we investigate the quality of the rationales produced by examining the additional predictive ability the rationales yield over the cryptic clues (Wiegreffe, Marasović, and Smith 2021). We find the quality of our predicted rationale to be quite dismal, but the quality of the ground-truth rationale was significantly higher.

This culminates in the suggestion that there is additional predictive power in the rationale, but that our models are just unable to learn to produce free-text rationale with good-enough quality to exploit it.

[1] We release the code to reproduce the results in this paper at https://github.com/WillHill257/cryptic-crossword-rationale.

[2] "Free-Text" and "Natural Language" are used interchangeably to describe free-form text sequences

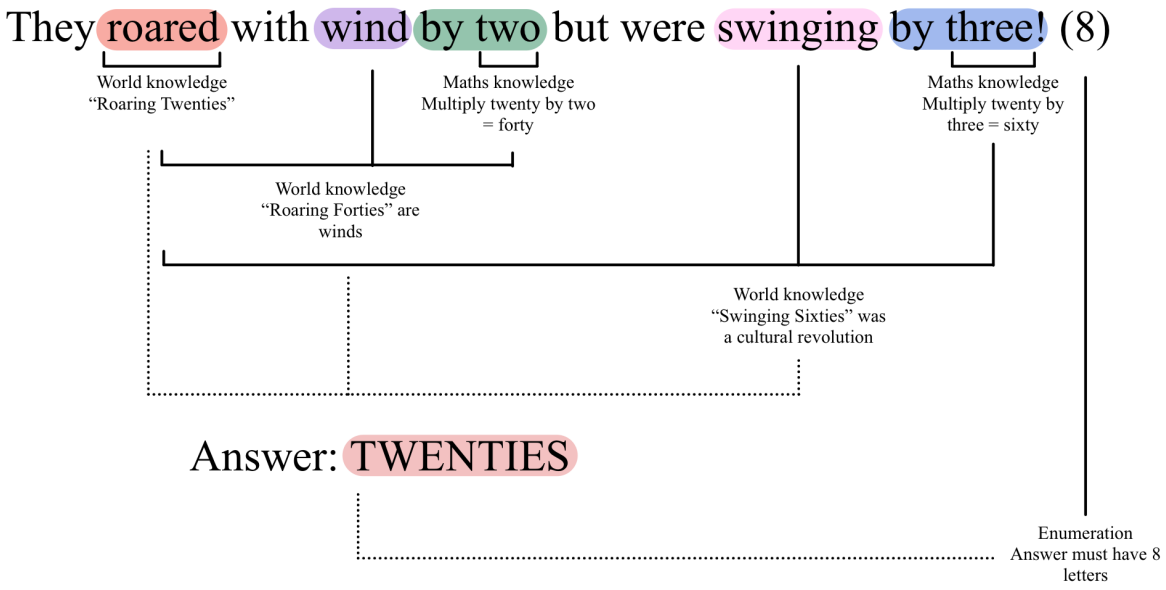

Figure 1: Example of how clever wordplays are required to decipher a cryptic clue. For this clue, world and maths knowledge is required.

Background and Related Work

Language Models

A Language Model is a probability distribution over sequences of words, which gives the ability to predict those words (i.e. construct meaningful sentences and bodies of text). At its simplest, a Language Model can be used to predict the next word in a given sequence of text by taking the option with the highest probability. A Neural Language Model uses a neural network to capture the complexities of natural language.

Language Models are generally trained with unsupervised (or self-supervised) learning, taking in a sequence of text and learning to reproduce that sequence of text. A common method is Masked Language Modelling (MLM), where the input text sequence of text is corrupted and the model taught to generate the missing segment. For example, given the sentence *"the sun is shining so it must be daytime"*, a corrupted input string could be *"the sun is shining so it must be [MASK]"* and the model has to learn that the missing part should be *"daytime"*.

Acquiring Knowledge

One of the main attractions of Cryptic Crosswords is the requirement for the solver to possess (1) crystallized knowledge[3], (2) an acute understanding and appreciation for the nuances of language (English to be exact), and (3) high fluid intelligence[4] (Friedlander and Fine 2016).

[3]the accumulation of knowledge and skills through experience

[4]the ability to think and reason abstractly

Accordingly, to build a Neural Language Model to solve Cryptic Crosswords, we need to store as much knowledge as possible in its parameters: this allows the model to gain general-purpose knowledge that allows it to intrinsically understand language and apply it to a downstream task (Roberts, Raffel, and Shazeer 2020; Raffel et al. 2020). The currently-accepted technique for doing so is pretraining the model, which in NLP is done on a data-rich task, generally via unsupervised learning on unlabelled data, and has led to state-of-the-art results on many NLP benchmarks (Raffel et al. 2020; Devlin et al. 2019; Yang et al. 2019; Dong et al. 2019; Liu et al. 2019; Lan et al. 2020).

Raffel et al. (2020) collated the "Colossal Clean Crawled Corpus" (C4), which consists of hundreds of gigabytes of English language data scraped from the Web, and used this to pretrain their T5 model (the parameters for various-sized models have been released). These pretrained parameters are used as the initialisation for the model when fine-tuning on downstream tasks. Furthermore, updating all of the pretrained parameters when fine-tuning produces better performance (Raffel et al. 2020).

Free-Text Rationale

One of the main issues with Neural Networks is that they operate as black boxes: we know what the given input is and what the produced output is, but we have no way of knowing how the network arrived at the answer. Comparably, this is also how humans operate: if you ask someone a question and they give you an answer, the only way to understand their thought process is to ask the person to explain it. Therefore,

it is a natural extension to ask the model to explain itself as well.

Multiple streams of research have focused on making Neural Language Models more interpretable: for example, interpretability by construction, using extractive rationales, and using free-text rationales. Interpretability *by construction* is performed by introducing architectural modifications such that intermediate steps in the model's decision making process can be observed and thus provide some insight into the final decision (Andreas et al. 2016; Jiang et al. 2019). For example, we combine component neural networks to achieve the overall goals. Interpretability *using extractive rationales* is performed by requiring the model to additionally output the span (substring) of the input text which motivated its answer, thus revealing what the model perceived as the most informative part(s) of the input (Lei, Barzilay, and Jaakkola 2016; DeYoung et al. 2020). Interpretability *using free-text rationales* is performed by requiring that the model generates a free-form, natural language explanation for its answer. The advantage of free-text rationales is that they allow the model to produce an explanation where its decision making process includes information not immediately present in the input (and so extractive rationales would be insufficient) (Camburu et al. 2018; Rajani et al. 2019).

In addition to providing insight into a model's decision making process, rationales have also been shown to improve the accuracy of the label prediction itself for common NLP tasks, such as Commonsense Question Answering (Rajani et al. 2019) and NLI (Kumar and Talukdar 2020). As of yet, the effect of rationales in the context of Cryptic Crosswords has not been investigated.

Furthermore, owing to the ambiguous and complex nature of Cryptic Crosswords, free-text rationales will be the form of interpretation and explanation required to provide reasoning over-and-above information found in the clue itself. A logical step would be to use a pipeline — there are two separate models, one mapping inputs to rationales, and another independently mapping the rationales to outputs. However, Wiegreffe, Marasović, and Smith (2021) demonstrate that pipelines are insufficient for free-text rationales: the final label predictions suffer from compounding errors; and rationales alone are subject to a loss of context (i.e. free-text rationales do not provide enough information alone to predict a label with sufficient accuracy). Instead, they present a *self-rationalising* model where the label and rationale are predicted simultaneously: this way, the model is forced to reason about the answer as it is producing the answer, which remedies both the issues raised. A comparison of these architectures is displayed in Figure 2.

Fine-Tuning

Given that we possess a pretrained neural language model, T5, we need to consider the methods for fine-tuning it on the downstream task of simultaneously predicting the answer to a cryptic clue and producing an associated free-text rationale.

$$(a) \qquad I \xrightarrow{M_1} R \xrightarrow{M_2} O$$

$$(b) \qquad I \xrightarrow{M_3} OR$$

Figure 2: Difference between a pipeline and self-rationalising model. I is the cryptic clue (as input), R is the free-text rationale, and O is the answer (as output). M_i is a distinct model. (a) is a pipeline model, using the clue to produce a rationale, and then using the rationale to produce the answer. These two stages are handled by separate models. (b) is a self-rationalising model, using the clue to produce both the answer and rationale in a single stage, using a single model.

Multitask Learning Fine-tuning can be performed in a number of ways, depending on the task. If there are multiple tasks, or the task can be broken down into multiple subproblems, then multitask learning is required, which can be performed either implicitly or explicitly. Implicit multitask learning is performed when the model learns how to perform smaller tasks as a byproduct of learning to perform a more complex task (this generally occurs during unsupervised learning) (Radford et al. 2019). This type of learning is exploited during our pretraining process to teach the model to perform language-general tasks. For example, if the pretraining corpus includes content from a forum, then in addition to the content learnt, the model may also know how to answer questions. Explicit multitask learning is performed when an instruction of the task to perform is included in the input (this generally occurs during supervised learning). In NLP, this instruction is generally a prompt added to the input text (Sanh et al. 2022). For example, given input text, we can prepend *"sentiment: "* to perform Sentiment Analysis, or we can prepend *"summary: "* to summarise the text. Sanh et al. (2022) also demonstrated that enough variability in natural language prompts (i.e. similar to how prompts occur in human conversation) makes the model robust to variations in held-out prompts, although the need for resilience to this variability should not be present in our Cryptic Crossword domain.

Sanh et al. (2022) showed that explicit multitask learning is the more effective of the two if we require our model to perform multiple tasks. In our Cryptic Crossword setting, we can utilise explicit multitask learning to differentiate between when our model needs to predict a rationale with the answer or not.

Curriculum Learning (CL) When humans learn to perform a complex task, they generally start with simpler concepts and move on to progressively harder ones at a later stage, and typically use the simpler concepts to aid the learning of the advanced ones. Our models will have to go through this same process. Simply asking a person, or our model, to learn the complex task outright leads to degraded performance since trying to learn too many new things at

once hinders the ability to learn them (Rozner, Potts, and Mahowald 2021).

Bengio et al. (2009) introduced the idea of applying *Curriculum Learning* to machine learning models. A survey by Soviany et al. (2022) provides multiple examples where Curriculum Learning outperforms conventional training methods. Rozner, Potts, and Mahowald (2021) demonstrate that Curriculum Learning in a Cryptic Crossword context dramatically improves results (over Efrat et al. (2021)) and use explicit multitask learning to do so: first train the model on progressively harder subproblems (e.g. definition lookup, then word descrambling etc.), and finally train the model on the problem of Cryptic Crossword clue deciphering.

However, two major problems exist with Curriculum Learning: how do we decide which tasks are simpler and which are more complex, and the occurrence of catastrophic forgetting.

For the former, prior work has considered multiple different methods for sorting tasks by difficulty, such as rule-based methods (Bengio et al. 2009), self-paced learning, manual annotation by domain experts, and using domain-dependent difficulty measures (Soviany et al. 2022). Soviany et al. (2022) conjecture that the best method for designing a curriculum is one where we model our own human learning experience, a method which Rozner, Potts, and Mahowald (2021) used with favourable results. Additionally, when separating the examples by difficulty, we need to ensure that each difficulty level has enough diversity (in terms of labels) to ensure we are still able to generalise effectively.

For the latter problem, catastrophic forgetting refers to the phenomenon where a cognitive system has previous information completely erased as a result of learning new information (French 1999). Typically, all natural cognitive systems gradually forget old information as new information is learnt. However, the better the system is able to generalise, the more likely it is to catastrophically forget (French 1999). Neural Networks are an example of such a cognitive system: the very features which allow the model to generalise are what will cause the model to forget the "simpler" tasks when it learns the more advanced tasks. Rozner, Potts, and Mahowald (2021) deal with this issue by periodically feeding the model an earlier task (although, this only decreases the likelihood of catastrophic forgetting occurring, and does not eradicate it entirely).

Conclusion

We have thus determined the following points:

1. The best way to embed knowledge into our models is via a pretrained model which had an unsupervised objective.
2. Free-Text Rationale is the type of rationale best suited to a problem of our variability.
3. An explicit multitask approach, with prompts, is the optimal method of fine-tuning on our downstream task of solving cryptic clues. Additionally, Curriculum Learning is a promising approach to more reliably acquire clue-solving skills.

Problem Representation

Cryptic Crosswords have existed since the 1920s and have become a popular enigma. As such, there are magnitudes of clue-answer samples to use as data sets, such as those collated in Efrat et al. (2021) and Rozner, Potts, and Mahowald (2021), but there are far fewer explanations given for why an answer is in fact the answer.

Because we are going to simultaneously predict a rationale and the answer, ideally we would want data where each clue-answer pair is explained which, practically, is not the case. However, we still want to use all the available data for training, to gain a more robust model. Fortunately, Narang et al. (2020) demonstrated that good-quality rationales can still be produced with a limited number of explained examples, although the quality does increase with more explained examples.

Consequently, we require a model that is able to predict a label either with or without an associated rationale. Raffel et al. (2020) presented a Unified Text-To-Text Transformer, T5, which can be used to accomplish this — it takes, as input, a sequence of text, and produces, as output, another sequence of text. They argue that any problem can be formulated in this format: for example, instead of a classification problem producing an ID for the class, it will produce the class name itself. Consequently, T5 can be used to address any task. Following Narang et al. (2020), prompts can be used to differentiate the task we want the model to perform: if we want the model to produce a rationale, we will prepend *explain* to the clue for the input text, and append *explanation* and the given explanation to the label for the target output text; if we do not want the model to predict a rationale because, for example, the sample does not have an associated explanation, then we do not include the prompts and an explanation. This allows us to utilise the entire collated data sets to train a single model which can simultaneously produce the label and a rationale. Importantly, this allows us to utilise an existing architecture proven to be excellent (Raffel et al. 2020; Narang et al. 2020; Sanh et al. 2022), without having to add any special architectural modifications, which is complex and may or may not work with the same efficacy.

Data

There is a multitude of clue-answer samples which can be drawn from newspapers. However, only a small portion of these have explanations attached (mostly when used to provide examples of how to solve cryptic clues). Nevertheless, there are many blogs which are focused on Cryptic Crosswords, and the authors of these blogs offer annotations and explanations for the clues and associated answers.

George Ho[5] has collated cryptic clues and answers from a variety of sources, as well as clue-answer pairs with associated annotations/explanations from various blogs[6].

Additionally, for our Curricular dataset, we reuse the dataset collated by Rozner, Potts, and Mahowald (2021).

[5]View his website at https://www.georgeho.org and Cryptics datasheet at https://cryptics.georgeho.org/datasheet.

[6]Dataset available on request.

Preparation

There are similarities between the answers of cryptic clues which prevent the model from learning how to solve the clue: it rather exploits lexical similarity between clues with the same answer (it becomes a case of "these two clues are similar, so their answers must be similar" instead of solving each clue outright).

Therefore, following Rozner, Potts, and Mahowald (2021), we will train and test our models on three different methods of splitting the data, using a train/validation/test split of 60/20/20:

1. Random Split: the dataset will be randomly split into training, validation, and test splits.
2. Naive Disjoint Split: the dataset will be split into training, validation and test splits such that all clues with the same answer appear exclusively in one of the splits.
3. Word-initial Disjoint Split: the dataset will be split into training, validation and test splits such that all clues which have answers starting with the same two letters will appear exclusively in one of the splits.

Each split gets progressively more restrictive, where the last split is designed to overcome T5's robustness to inflection (Rozner, Potts, and Mahowald 2021) — derivatives of words are lexically similar to each other, so much so that T5 could perceive them as having negligible difference.

Thus, the Word-initial Disjoint split offers the fairest measure of generalisability to new clues.

Methodology

Aims

The aims of this paper are to determine: (1) the effect of simultaneously producing a free-text rationale with the cryptic clue answer on the answer accuracy (this is our main aim); (2) the quality of the rationale produced; and (3) the effect of model size on answer accuracy and rationale quality.

Models

In order to achieve these aims, we need to define three model types (following Wiegreffe, Marasović, and Smith (2021)):

1. **I→O:** this model takes only the cryptic clue (**I**) as input and produces only the predicted answer (**O**) as output.
2. **I→OR:** this model takes only the cryptic clue (**I**) as input and produces both the predicted answer (**O**) and the associated free-text rationale (**R**) as output.
3. **IR→O:** this model takes both the cryptic clue (**I**) and an associated free-text rationale (**R**) as input and produces the predicted answer (**O**) as output. The rationale can either be the predicted ones (from the I→OR model), or the ground-truth rationale from the dataset.

As mentioned in the *Free-Text Rationale* subsection, a pipeline model (i.e. an I→R→O model) is insufficient when free-text rationales are involved.

Following the discussion in the *Problem Representation* section, we use T5 models and a multitasking approach to differentiate between their types during training and inference.

A T5 model is a text-to-text transformer and, as such, all the input and output are included as a single sentence, respectively. To this end, we use prompts to instruct the model as to the task to perform as well as include prompts to distinguish the different parts of the sentence. Specifically, the input and output strings take the following form (per model), where the prompts are in bold (an example is given in Table 1):

1. **I→O:**

   input: **clue:** clue

   output: answer
2. **I→OR:**

   input: **explain clue:** clue

   output: answer **explanation:** rationale
3. **IR→O:**

   input: **clue:** clue **explanation:** rationale

   output: answer

Metrics

We evaluate the models according to two streams (as applicable): the accuracy of the answers, and the quality of the rationales.

Firstly, the accuracy is simply the percentage of answers that the model correctly predicted ($\in [0, 100]$). A correct answer is one that exactly matches the true answer, regardless of letter case.

Secondly, the Rationale Quality is determined using an automated simulatability score (Wiegreffe, Marasović, and Smith 2021), which accounts for the variation that can arise in natural language. To calculate this, we train an I→O model, an I→OR model, and a IR→O model (using the predicted rationales from the I→OR model). We then compute the difference between the I→O and IR→O models' accuracies, which yields the additional predictive ability of the rationales. Specifically, we compute:

$$(IR{\rightarrow}O) - (I{\rightarrow}O)$$

The size of a positive score indicates just how good the quality is, whilst a negative score indicates a rationale of terrible quality — it hinders instead of benefits performance.

Curriculum Learning

Based on the findings of Rozner, Potts, and Mahowald (2021), we train models based on two curricular tasks:

1. **Definition Lookup:** Given a definition and the number of letters in the answer, determine the word corresponding to the definition. For example, given the input *"Litigator's group (3)"*, the corresponding answer is *"aba"*.
2. **Word Descramble:** Given a definition, the number of letters in the answer and an anagram of the answer, determine the answer. The anagram is randomly added to

Model Type	Input String	Output String
I→O	*clue:* School with American vowel sound? (5)	SCHWA
I→OR	*explain clue:* School with American vowel sound? (5)	SCHWA *explanation:* SCH (school) W (with) A (American)
IR→O	*clue:* School with American vowel sound? (5) *explanation:* SCH (school) W (with) A (American)	SCHWA

Table 1: A table showing an example of the formats of the input and output strings for the different model types. The given clue is "School with American vowel sound? (5)" and has answer "SCHWA" with ground-truth rationale "SCH (school) W (with) A (American)". In English, a "schwa" is a vowel sound produced when the lips, tongue and jaw are completely relaxed. For example, the *a* in *about* is a schwa.

the beginning or end of the input string to ensure robustness to either format. For example, given the input string *"baa Litigator's group (3)"*, the corresponding answer is *"aba"*.

Additionally, we also use multitask prompting to instruct our model which task to perform: for definition lookup, we prepend the prompt *"phrase:"*; and for the descrambling task, we prepend the prompt *"descramble:"* (an example is given in Table 2).

We trained the curricular models with a pipeline approach: first, the model was trained on the Definition Lookup task for a fixed number of epochs and then, using the same dataset, trained on the Word Descrambling task for a fixed number of epochs. We did not periodically revisit a Definition Lookup task whilst performing the Word Descrambling training and found the results to not be adversely affected (further discussed in the Baseline comparison in the Appendix).

Transfer Learning

As discussed in the *Acquiring Knowledge* subsection, it is of the utmost importance to use transfer learning to embed general-purpose knowledge into our models.

Raffel et al. (2020) have pretrained various-sized T5 models on a Masked Language Modelling (MLM) task using the C4 dataset.

Consequently, for the initialisation of our Curriculum Learning models, we use these pretrained MLM weights. For the initialisation of our I→O, I→OR, and IR→O models, we can either use the pretrained MLM weights, or the weights of the trained Curriculum Learning models.

In order to determine our aims, we use both the T5-Small and T5-Large models (which have 60 million and 770 million parameters, respectively) as the base for each of our I→O, I→OR, and IR→O model types. We run experiments across both these sizes as well as across each of the three data splits (as defined in the *Data* section), which will give us an accurate indication of the generalisability of our models.

Experiments and Results

Baselines

We replicate the works of Efrat et al. (2021) and Rozner, Potts, and Mahowald (2021) to verify that our implementation is correct. Refer to the Appendix for more information.

1. Effect of Self-Rationalisation

Purpose In order to address the main question of this paper, we compare the accuracy of models where we do and do not simultaneously predict the free-text rationale with the cryptic clue answer. Here, we do not use Curriculum Learning to pretrain our models, but rather rely on the MLM pretraining from Raffel et al. (2020).

Result The results are presented in the first two (numerical) columns of Table 3.

Notably, on all but the *Word Initial Disjoint* split, the accuracy of the I→O models were higher than the accuracy of the I→OR models. This is indicative of the model learning "shortcuts" to solve the cryptic clue instead of learning the underlying skills required to solve them in general. When we consider the *Word Initial Disjoint* split, which was designed to be the fairest representation of generalisability, the accuracies are extremely low, but there is a slight improvement due to self-rationalism.

Additionally, for each data split, the accuracy of both the I→O and I→OR models increases as the model gets larger.

2. Effect of Self-Rationalisation (with Curriculum Learning for clue solving)

Purpose Following Rozner, Potts, and Mahowald (2021), we aim to allow our models to more robustly acquire the knowledge and skills required to solve cryptic clues by using a Curriculum Learning Approach. We repeat the previous experiment, except we now first apply Curriculum Learning to our models (i.e. initialise them with a model trained on our curriculum) before fine-tuning them on our downstream task of answering cryptic clues (both with and without the simultaneous production of rationale).

Result The results are presented in the last two columns of Table 3.

Here, the I→O accuracy is better than the I→OR on all but the largest model trained on the *Word Initial Disjoint* split. This is also indicative of the model learning "shortcuts" to solve the problems instead of the underlying skills. However, even the small model trained on the *Word Initial Disjoint* split had a better I→O accuracy than I→OR accuracy. Seeing as this split is meant to increase the models' ability to generalise, this result suggests that simultaneously predicting the rationale may hurt performance overall. Experiments 3 and 4 further investigate this point.

Additionally, for each data split, the accuracy of both the I→O and I→OR models increases as the model gets larger.

Curricular Task	Input String	Output String
Definition Lookup	*phrase:* Litigator's group (3)	aba
Word Descrambling	*descramble:* baa Litigator's group (3)	aba

Table 2: A table showing an example of the formats of the input and output strings for the different Curriculum Learning tasks. The given input is "Litigator's group (3)" and has answer "aba". In the Word Descrambling task, we additionally randomly scramble the letters of the answer and either prepend or append this to the input string.

Data Split	Model Size	No CL		With CL	
		I→O	I→OR	I→O	I→OR
Random	t5-small	0.8088	0.3808	**1.2892**	0.4812
	t5-large	10.8745	6.4310	**13.1491**	7.3741
Naive Disjoint	t5-small	0.1802	0.1002	**0.3663**	0.1827
	t5-large	2.8210	2.6442	**3.3187**	3.2210
Word Initial Disjoint	t5-small	0.0025	0.0099	**0.0230**	0.0131
	t5-large	0.1553	0.3139	0.1578	**0.3451**

Table 3: A table showing the accuracy ($\in [0, 100]$) of various models across each of our data splits and model sizes. The first two results columns are based on models **with no** Curriculum Learning applied and the final two results columns are based on models **with** Curriculum Learning applied. Going down, the data splits increase in restrictiveness, where the first is most prone to "shortcuts" and the last the best indication of generalisability. No Beam Search was used during inference. The highest accuracy in each row is in bold (higher is better).

Comparison to no CL Evidently, across all data splits and model sizes, the results with Curriculum Learning are better than the corresponding results with no Curriculum Learning applied. Consequently, for all future experiments, we only use models initialised with a model trained on our curriculum.

3. Performance of our rationale-producing models on *Cryptonite*

Purpose As support to Experiment 2's suggestion that rationale may hinder performance, we use models pretrained (on each of our three data splits) to produce rationale and evaluate them on the "official" *Cryptonite* dataset. This will allow us to compare the baseline performance of a model that does not explicitly know how to produce a rationale to our models, which do know how to produce rationales.

Result The results are presented in Table 4.

Focusing on the last column (i.e. the I→OR models), the accuracy when simultaneously producing the rationale is lower for the models that knew how to produce rationale than the model which didn't know how to produce rationale (i.e. the I→OR model with the *None* Data Split was asked to produce rationales despite never being explicitly trained to do so). It is possible that this *None* Data Split model learnt how to produce a rationale as a byproduct of the unsupervised MLM pretraining Raffel et al. (2020) applied. Consequently, when we force the models to explicitly reason about the rationale, it reduces their capacity to learn how to answer the primary task of deciphering and answering the cryptic clues, hence the drop in accuracy.

It appears that the models based on our data splits perform better than the *None* model. However, this deduction is relatively meaningless considering that the models based on our data splits were trained on a completely different dataset. There may have been an overlap between *Cryptonite* and our data source (since both sources are based on the Cryptic Crosswords which appear in newspapers and ours just has further annotations). This means we may have leakage from our training set into the *Cryptonite* test set.

The more important take-away is that the rationale hurt performance.

Data Split	Number of Beams	I→O	I→OR
None	1	**12.5358**	9.2289
	5	**13.1093**	9.6953
Random	1	**13.7936**	6.9350
	5	**20.2699**	12.3982
Naive Disjoint	1	**13.0443**	6.3539
	5	**17.1656**	10.8690
Word Initial Disjoint	1	**13.4763**	6.4916
	5	**17.7582**	11.4119

Table 4: Performance of T5-Large-based models on the "official" *Cryptonite* dataset with a varying number of beams used during inference. The Data Split column refers to the way the model was initialised — *None* means we only used CL pretraining and the others refer to the resultant model of training (with CL) on that split. Notably, the I→OR values for Data Split *None* are the only I→OR values determined from a model not explicitly trained to produce rationale. The highest accuracy in each row is in bold (higher is better).

4. Rationale Quality

Purpose Experiments 2 and 3 suggest that simultaneously producing rationale degrades accuracy instead of improving it. However, without investigating the quality of the produced rationale, there is no way to ascertain for certain whether that deduction is correct or not — it may be the case that instead of rationale production hurting performance, our models may simply fail to produce good enough rationale and that trying to learn this dominates the training process, thereby hindering the solving of cryptic clues.

Hence, we train models which additionally take the rationale as input (i.e. IR→O models) and compare their accuracies against the models which do not take this additional rationale (i.e. the I→O models previously trained). This provides us with a quantity representing the collective rationale quality.

Result The results are presented in Table 5.

The Quality of our Predicted Rationales is dismal, with 4 of the 6 tests having negative values. This means that instead of the rationale granting additional predictive ability, their mere inclusion in the problem detracts from the information presented in the cryptic clues.

We additionally examined the Quality of the True Rationales and found it to be significantly better. This means that the information contained in the ground-truth rationales far outweighs the information contained in the rationales that our models learnt to predict.

Consequently, it is clear that our models and training process are unable to learn the task of producing meaningful free-text rationales. It is quite possible that better rationales could impact the accuracy in a different manner, so we cannot state that the inclusion of rationales in the prediction decreases the accuracy of the models.

Potential reasons for discrepancy There are various reasons why our models might be unable to learn to produce meaningful free-text rationales. One such conjecture could be similar to the argument presented by Rozner, Potts, and Mahowald (2021) in that the rationales are too complex to learn to produce using only fine-tuning on our downstream task. Coupled with the already-complex task of learning to decipher cryptic clues, the task we present to our models is simply too hard to learn without extra help.

Discussion and Future Work

Following the discovery of Experiment 4, it is clear our models are incapable of producing rationales with a good enough quality to potentially have meaningful information.

Consequently, future avenues of exploration could include finding better methods to produce the free-text rationale. Additionally, it is worthwhile to determine the correlation between how good the Rationale Quality is and the accuracy of the model when solving cryptic clues and, based on this, determine what the threshold quality would be so that simultaneously producing rationale improves accuracy.

As mentioned in the results of Experiment 4, a possible reason for our inability to produce rationales of good quality is to build on the idea presented in Rozner, Potts, and Mahowald (2021) — that the model is trying to learn to do too much at once. They aimed to reduce the amount of skill the model had to learn in order to solve the cryptic clues by introducing Curriculum Learning. We also explore the use of Curriculum Learning in solving the clues and saw improved results. However, the additional task of producing a free-text rationale is quite complex. Therefore, we may be falling into a similar issue where the task of learning to solve both problems simultaneously is too complicated and requires a more refined approach, such as Curriculum Learning on the rationales as well.

A number of cherry-picked examples of our predicted rationales are displayed in Table 6. In this table, we compare the rationales produced by I→OR models trained on our Random and Word Initial Disjoint (WID) data splits. It is notable that the WID rationales are more meaningful and plausible and do relate to the associated predicted answers. The same cannot be said for the Random split's rationales, which are often duplicates of the predicted answer and, as such, offer no additional predictive ability. This is supported by the difference in Rationale Quality in Table 5. The fact that the WID model gets the answer wrong despite having a more meaningful rationale correlates with the low Rationale Quality in Table 5 — the rationales are just not good enough.

Conclusion

Our initial results suggest that simultaneously producing the free-text rationale whilst predicting the answer to the cryptic clue only slightly improves performance under certain circumstances. However, we found that we were predicting low-quality rationale — the ground-truth rationales are of much better quality and yield significant additional predictive power over our predicted rationales.

Therefore, better ways of producing the rationale should be explored in order to gain a more robust conclusion.

Appendix

Baselines

We replicated two prior works to establish baselines.

Efrat et al. (2021) Here, the authors presented the dataset *Cryptonite*, a collation of cryptic clues and answers. They additionally trained a T5-Large model (initialised with the same pretrained MLM weights we intend to use) on an "official" split, which is equivalent to our naive disjoint split.

We are able to reproduce their results (shown in Table 7) with 5 beams (in the inference Beam Search). We also determine the results for a single beam — this sets a baseline against which we can compare the performance of our I→OR models later.

Rozner, Potts, and Mahowald (2021) Here, the authors investigated a Curriculum Learning approach to solving cryptic crosswords. They collated their own dataset of cryptic clues for this task, but did not release it due to copyright concerns. Hence, we cannot replicate their results. They did, however, create their own Curricular dataset and it is this data that we use for our Curriculum Learning.

Data Split	Model Size	I→O	IR→O				
			Accuracy (Predicted Rationale)	Predicted Rationale Quality	Accuracy (True Rationale)	True Rationale Quality	Quality Difference (True − Predicted)
Random	**t5-small**	1.2892	1.1854	-0.1038	26.6544	25.3652	25.4690
	t5-large	13.1491	13.0503	-0.0988	45.6696	32.5206	32.6193
Naive Disjoint	**t5-small**	0.3663	0.3267	-0.0396	23.2503	22.8840	22.9236
	t5-large	3.3187	3.7920	0.4733	36.5007	33.1820	32.7088
Word Initial Disjoint	**t5-small**	0.0230	0.0205	-0.0025	11.5066	11.4836	11.4861
	t5-large	0.1578	0.4922	0.3345	19.0972	18.9394	18.6049

Table 5: A table showing the Rationale Quality for both Predicted and Ground-Truth Free-Text Rationale and their difference across the three data splits and two model sizes. All results were attained on models trained with Curriculum Learning. The third column (I→O) is the same as the second last column in Table 3 and represents the accuracy when no rationales are considered at all. The two Accuracy columns are based on the accuracy of the IR→O model when trained on Predicted Rationale (from the associated I→OR model) and Ground-Truth Rationale, respectively. The two Rationale Quality columns contain the automated simulatability scores for their respective models. The final column represents the difference between the Qualities of the Predicted and True Rationale. Higher accuracies are better; a negative Rationale Quality signifies the rationale degrading performance.

Clue	True Answer	True Rationale	Data Split	Predicted Answer	Predicted Rationale
"Butt call" in storm (7)	RAMPAGE	RAM (butt) + PAGE (call)	Random	CROSSBY	CROSSBY
			Word Initial Disjoint	CROSSBY	CROSS (butt) + BY (call)
vegetable said to be languishing in wings of RMH (6)	RADISH	An anagram (to be languishing) of SAID in the outer letters (wings) of RMH. Another Australian abbreviation that would mean nothing to a UK solver	Random	RADISH	RADISH
			Word Initial Disjoint	TURNIP	TURN (said to be languishing) in RMH (wings of RMH)
Dregs that remain around university (7)	RESIDUE	RESIDUE	Random	RESIDUE	RESIDUE around U
			Word Initial Disjoint	RUINSAU	RUINS (remains) around U (university)

Table 6: The predicted answers and rationales of T5-Large-based I→OR models trained, using Curriculum Learning, on our Random and Word Initial Disjoint (WID) data splits. Even though we predict the incorrect answer for the first clue, we can see a more meaningful rationale for the WID split. This aligns with the higher quality we found (in Table 5). For the second and third clues, the Random model gets the answer right despite not having a meaningful rationale, which shows that the rationale is not really utilised when answering the clue. Similarly, the WID model gets the answer wrong despite having a more meaningful rationale, reinforcing that our rationales do not provide additional predictive ability. The WID model for these two clues also shows that we are somewhat able to produce rationales (even if they are low quality) that are related to the predicted answer, signifying that we have made progress with rationale production.

Number of Beams	Efrat et al.	Our replication of Efrat et al.	Rozner, Potts, and Mahowald	Our replication of Rozner, Potts, and Mahowald
1	—	6.7	—	12.5
5	7.6	6.9	10.9	13.1

Table 7: A table showing the accuracy ($\in [0, 100]$) of various T5-Large models for the Cryptonite dataset. We present a replication of Efrat et al. and Rozner, Potts, and Mahowald (using 5 beams) and also show our evaluations (for the I→O model type) using a single beam. Note that the Efrat et al. accuracy values are based on an I→O model with no CL and the Rozner, Potts, and Mahowald accuracy values are based on an I→O model with CL. Higher is better.

The authors also recreated the results of Efrat et al. (2021) and further determined the performance on the *Cryptonite* dataset using a Curriculum Learning approach. It is these results against which we compare our implementations.

As shown in Table 7, our results (with CL) are slightly worse. However, this can easily be attributed to the fact that during our Curriculum Learning, we just increased the curriculum difficulty without periodically introducing an easier problem again. Therefore, we are susceptible to a small amount of catastrophic forgetting (in agreement with Rozner, Potts, and Mahowald's (2021) discussion), but all-in-all, the results are similar.

Consequently, for our experiments, we use this method of Curriculum Learning (without periodic lookup).

References

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 39–48.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 41–48. New York, NY, USA: Association for Computing Machinery.

Camburu, O.-M.; Rocktäschel, T.; Lukasiewicz, T.; and Blunsom, P. 2018. e-snli: Natural language inference with

natural language explanations. *Advances in Neural Information Processing Systems* 31.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

DeYoung, J.; Jain, S.; Rajani, N. F.; Lehman, E.; Xiong, C.; Socher, R.; and Wallace, B. C. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4443–4458. Online: Association for Computational Linguistics.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified language model pre-training for natural language understanding and generation. *ArXiv* abs/1905.03197.

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

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* 3(4):128–135.

Friedlander, K. J., and Fine, P. A. 2016. The grounded expertise components approach in the novel area of cryptic crossword solving. *Frontiers in Psychology* 7.

Jiang, Y.; Joshi, N.; Chen, Y.-C.; and Bansal, M. 2019. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2714–2725. Florence, Italy: Association for Computational Linguistics.

Kumar, S., and Talukdar, P. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8730–8742. Online: Association for Computational Linguistics.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv* abs/1909.11942.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 107–117. Austin, Texas: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv* abs/1907.11692.

Narang, S.; Raffel, C.; Lee, K.; Roberts, A.; Fiedel, N.; and Malkan, K. 2020. WT5?! Training Text-to-Text Models to Explain their Predictions. In *arXiv preprint arXiv:2004.14546*.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 2019. Language models as knowledge bases? 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)*, 2463–2473. Hong Kong, China: Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Raffel, C.; Shazeer, N. M.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv* abs/1910.10683.

Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4932–4942. Florence, Italy: Association for Computational Linguistics.

Roberts, A.; Raffel, C.; and Shazeer, N. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5418–5426. Online: Association for Computational Linguistics.

Rozner, J.; Potts, C.; and Mahowald, K. 2021. Decrypting Cryptic Crosswords: Semantically Complex Wordplay Puzzles as a Target for NLP. In *Advances in Neural Information Processing Systems*, volume 34.

Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T. L.; Raja, A.; et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *The Tenth International Conference on Learning Representations*.

Soviany, P.; Ionescu, R. T.; Rota, P.; and Sebe, N. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*.

Talmor, A.; Elazar, Y.; Goldberg, Y.; and Berant, J. 2020. oLMpics-On What Language Model Pre-training Captures. *Transactions of the Association for Computational Linguistics* 8:743–758.

Wiegreffe, S.; Marasović, A.; and Smith, N. A. 2021. Measuring Association Between Labels and Free-Text Rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10266–10284. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J. G.; Salakhutdinov, R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.