Transformers Can Compose Skills To Solve Novel Problems Without Finetuning

Anonymous ACL submission

Abstract

It is possible to achieve improved prediction performance with Transformers on unseen datasets by adding disparate new training tasks to an existing multitask training regime. We demonstrate that this can be attributed to a compositional mechanism rather than memorisation. Performance on DROP, DROP-CS and ROPES datasets can be improved by over 26 percent without finetuning through application of numerical reasoning tasks, while performance on seven other question-answering datasets that would not be expected to be improved remains essentially unchanged. By filtering our evaluation datasets to only those samples that have no answer overlap to similar training samples, and then further restricting to those samples which have the least semantic similarity with the training set, we show that improved performance after adding numerical reasoning tasks was not attributable to direct lookup. Our code and filtered datasets are available at https://github.com/anonymised.

1 Introduction

In this paper we present empirical findings on the ability of sequence-to-sequence Transformer models (a.k.a Transformers) to compositionally generalise in the domain of question answering, where both the input (question) and the label (answer) are expressed in natural language. We focus on the situation where the necessary composition is over disparate skills that must be learned over multiple training samples. To do so, we synthesise and extend several existing works, most notably the UnifiedQA multitask training environment and associated datasets (Khashabi et al., 2020), work on injecting numerical reasoning into Language Models (Geva et al., 2020) and research into evaluating similarity between training and test splits in the natural language domain (Lewis et al., 2021; Elangovan et al., 2021).

Over a forward pass through a Transformer, the high-dimensional vector (embedding) associated with a particular input token comes to incorporate information from other tokens in the input sequence (Vaswani et al., 2017; Manning et al., 2020; Russin et al., 2021). Resulting embeddings may encode the contextual meaning of words, syntactic grammatic structure (Manning et al., 2020), and mathematical structural rules (Russin et al., 2021).

Common practice in training Transformers, both in initial pretraining and subsequent training phases, is to allow weight updates to all layers of the model in the backward pass, including the initial embedding table from which subsequent training steps will retrieve updated embeddings (Devlin et al., 2019; Raffel et al., 2020).

The above two observations combine to the following uncontroversial conclusion; over the course of training, the embedding for a particular token will come to encode information not only from other tokens it has directly appeared in an input sequence with, but also indirectly from any token that has appeared in an input sequence with those tokens and so forth. Thus, at each step Transformers can be said to perform partial information propagation over a matrix of all vocabulary tokens against each other; or more broadly we can observe a mechanical and rather intuitive view of how a Transformer can compose information learned across its training history.

Compositional generalisation can be summarised as the ability to learn a set of atomic elements and to be able to generalise to an exponential number of valid novel combinations of those elements[1] (Fodor and Pylyshyn, 1988; Lake et al., 2017; Russin et al., 2020). This is significant in that it may provide a means for a model to generalise beyond its training distribution in a manner consistent with some models of human cognition (Baroni, 2020; Russin et al., 2020; Russin et al., 2019; Dankers et al., 2021). Many recent works evaluate and attempt to improve model performance on compositional generalisation, particularly in the context of semantic parsing (Lake and Baroni, 2018; Hupkes et al., 2020; Keysers et al., 2020; Furrer et al., 2020; Yin et al., 2021; Yanaka et al., 2021; Kim and Linzen, 2020). These works typically evaluate performance using non-i.i.d test splits where the test samples use elements seen in training, and where the labels are compositions derived from those elements but are different to those encountered in training.

However, empirical study of this phenomena in the context of natural language inputs with non-synthetic natural language outputs such as our question-answering domain is limited (Dankers et al., 2021). We take the liberty of suggesting that the compositional mechanism described above provides the vehicle for a Transformer to compositionally generalise in natural language. However, a conjecture that a Transformer could potentially exhibit this behaviour is different from a demonstration that a model actually does do this in any material way. We tested this through adapting the idea of using non-i.i.d test splits for natural language outputs. Starting by considering different datasets to those used in training as our test splits, we refine these further by only considering samples that have normalised answers (Rajpurkar et al., 2016) without word overlap with the normalised answer of the most semantically similar training example, the latter as measured using sentence embeddings (Reimers and Gurevych, 2019). In other words, those samples that have answers involving unlikely word compositions relative to similar training samples.

There is not a consensus on the degree to which Transformers and other neural models are able to generalise beyond their training distribution (Bahdanau et al., 2019; Hupkes et al., 2020; Dankers et al., 2021). For example Lewis et al (2021) shows that when considering three open-domain question-answering datasets, after eliminating test questions that are the same as those encountered in training, a BART (Lewis et al., 2020) model performs extremely poorly. The authors suggest it may hence only be capable of memorising[2] highly similar training examples. More broadly, various works (Lake and Baroni, 2018; Bahdanau et al., 2019; Russin et al., 2020) note poor generalisation for unlikely compositions of known elements. On the other hand, a number of papers (Kim et al., 2021; Furrer et al., 2020; Ontañón et al., 2021) propose approaches to enhancing the ability of neural models to compositionally generalise, in some cases demonstrating performance to an impressive extent. In a relevant study to our work (Dasgupta et al., 2020), it was initially observed that sentence embeddings produced by training on SNLI (Bowman et al., 2015) generalised poorly to predictions made on the Comparisons dataset. The authors say this requires encoding of systematic rules rather than dataset-specific heuristics. After training in a multitask fashion on both SNLI and Comparisons, good performance on both datasets was observed suggesting that the resulting embeddings now encoded systematic information. Another study (Hendrycks et al., 2021), considers challenging test datasets which contain unlikely samples relative to their training data. It is noteworthy that the UnifiedQA-trained version of the T5 model (Raffel et al., 2020) outperforms the much larger GPT3 (Brown et al., 2020) model on these datasets.

Our contributions can be summarised as: (1) A demonstration that a general-purpose Transformer can usefully compose disparate information learned across the training history to answer novel questions in the natural language domain and that composition and not memorisation is responsible for improved performance. (2) We illustrate a method of identifying evaluation samples that are unlikely to have memorisable answers. (3) We provide an environment for further study on the compositional effects of adding disparate tasks to a multitask training regime.

[1] Our usage of the compositional generalisation term is more literal than that by some authors in that we use it to describe a capability rather than a mechanism such as systematicity for instantiating the capability.

[2] We adopt this terminology of memorisation as the ability to directly derive an answer from a materially similar training sample.

2 Related Work

The UnifiedQA project (Khashabi et al., 2020) demonstrates that it is possible to attain good performance on unseen evaluation datasets (those that have not been involved in either pretraining or finetuning) after further training of a pretrained sequence-to-sequence Transformer on a variety of question-answering datasets in a multitask fashion. However two datasets that still have relatively poor performance are DROP (Dua et al., 2019) and DROP-CS (Gardner et al., 2020). These datasets offer the particular characteristic that some simple mathematical literacy (e.g. simple addition or ability to select the second highest element from a list) is helpful in order to correctly answer a question. Geva et al (2020) demonstrated significant performance improvement on DROP by pretraining on two datasets (TD and ND), that they designed to instill simple mathematical skills. This is followed by finetuning on DROP. Our work extends this idea by adding TD and ND to our existing multitask training mixture and analysing the impact (without finetuning) on DROP, DROP-CS, ROPES and on seven other question-answering datasets that we would not expect to benefit from the addition of these tasks. Lacking the resources to train the larger T5 models (Raffel et al., 2020), we empirically determined that the much smaller BART (Lewis et al., 2020) model gave us slightly better results than T5-base. Hence, we use BART for all our experiments while expecting that our results will generally be much lower than those reported in the UnifiedQA paper against the larger T5 models.

As noted, Hendrycks (2021) developed a number of challenging evaluation-only datasets. We combined four of their mathematics-focused datasets[3] into a single evaluation dataset which we call MMLU-M. The ability of sequence-to-sequence Transformers to learn simple mathematics is demonstrated by Nogueira et al (2021) and we experimented with their numerical representation format. In common with our work, Russin et al (2021) provide evidence for compositionality in contrast to memorisation of training data, in this case in the purely mathematical domain. They outline a method for probing embeddings to illuminate the compositional processing mechanism Transformers employ in the math domain and suggest that with sufficient training data Transformers can learn to compose to an extent while also describing their limitations.

It is challenging to measure the extent of Train-Test data leakage in natural language question-answering. In the area of open-domain question answering, Lewis et al (2021) identify training samples that have essentially the same normalised answers as an evaluation sample. For those, samples with questions that semantically match the evaluation question are manually identified. Noting that this approach focuses on identifying memorisable question-answers and lacking resources to perform manual annotation, we instead focus on identifying evaluation samples that cannot be memorised from any training example and find that it is possible to do so in a mostly automated fashion. Also considering the question of train-test overlap, Elangovan et al (2021) performs an analysis using cosine similarity of bag-of-words vectors as the similarity function. We initially adopted this approach but found that it does not work well for our numerical datasets where each individual number needs to be treated as a separate word, leading to an excessively large bag-of-words vector size.

For brevity, here we omit works on compositional generalisation already discussed in the introduction.

Our work has some commonality with a variety of work that focus on improving compositionality through training data enhancement. For example, Kim et al (2021) performs task-specific annotation of the training data with good results, and a number of works observe that compositional generalisation improves with variability in training data either through adding additional primitives to the SCAN training set (Kagitha, 2020), data augmentation (Andreas, 2020), or through the application of masked language pretraining (Furrer et al., 2020; Gontier et al., 2020).

3 Experimental Setup

3.1 Training Datasets

We extended the UnifiedQA multitask training environment (Khashabi et al., 2020) to incorporate

[3] MMLU-M is comprised of the elementary, high school and college mathematics datasets plus the high school statistics dataset.

Evaluation Dataset	Count	Filtered Count	Eval Type	Benefit From +TDND?
DROP (Dua et al., 2019)	8734	3102	F1	Y
DROP-CS (Gardner et al., 2020)	945	326	F1	Y
MMLU-M (Hendrycks et al., 2021)	963	485	MC (4)	N
Physical IQA (PIQA) (Bisk et al., 2020)	1838	722	MC (2)	N
Social IQA (SIQA) (Sap et al., 2019)	1935	753	MC (3)	N
CommonsenseQA (CQA) (Talmor et al., 2019)	1221	408	MC (5)	N
QASC (Khot et al., 2020)	926	345	MC (8)	N
QASC with IR (QASC-IR) (Khot et al., 2020)	926	338	MC (8)	N
ROPES (Lin et al., 2019)	1688	461	F1	Y
NEWSQA (Trischler et al., 2017)	4341	1944	F1	N

Table 1 Evaluation Datasets. Number of multi-choice options in brackets. +TDND refers to the addition of the two numerical literacy tasks to training. Note that MMLU-M obviously could benefit from numerical literacy but does not contain a significant number of examples that can benefit from the kind of simple mathematical skills imparted by TD or ND.

arbitrary training mixtures and with extensive instrumentation to facilitate comparative analysis of prediction performance of the same evaluation datasets against BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) models trained using different training mixtures. The baseline training datasets (collectively referred to as UQA) are: SQUAD 1.1 (Rajpurkar et al., 2016), SQUAD 2 (Rajpurkar et al., 2018), NarrativeQA (Kočiský et al., 2018), RACE (Lai et al., 2017), ARC (Clark et al., 2018), Regents (Clark et al., 2019b), OpenbookQA (Mihaylov et al., 2018), MCTest (Richardson et al., 2013), and BoolQ (Clark et al., 2019a).

We reformatted the two numerical reasoning datasets from Geva et al. (2020) into UnifiedQA-like format as follows:

'Numerical' Dataset (ND): `question \\n<tab>answer`

'Textual' Dataset (TD): `question \\n context paragraph<tab>answer`

These two datasets were added singly and together to the baseline UQA mixture to form UQA+ND, UQA+TD and UQA+TDND mixtures.

Training hyperpameters are listed in Appendix A.

3.2 Evaluation Datasets

At the expense of large performance gains, we did not finetune for evaluation datasets as this would remove our ability to determine what was causing any change in performance. This also enabled us to measure the effect of different training mixtures on each evaluation dataset from the same trained model checkpoint.

Following standard practice (Rajpurkar et al., 2016), we used the F1 score on the unstemmed word overlap between the normalised prediction and the normalised label as our prediction scoring metric for non-multichoice datasets. For multi-choice datasets we considered the F1 score between the normalised prediction and each normalised option and selected the option with the highest score as the choice. We refer to this method in experiments as MC.

We selected ten evaluation datasets as noted in Table 1. In all cases we started with the publicly available development split, except for MMLU-M which aggregates several test splits.

It was discovered that the DROP development set contained over 800 duplicates with other DROP development set samples. DROP-CS, MMLU-M and SIQA also contained small numbers of duplicates. All our experiments are reported on deduplicated versions of these datasets, hence counts given may not match prior work.

After evaluating the similarity of each evaluation sample to the training set, we created separate versions of each that only contain samples for which there is no answer word overlap with the most similar training sample. We refer to these versions as *filtered* and note counts for these in Table 1.

DROP and DROP-CS samples may have numeric or textual answers. In our experiments, samples with numeric answers have hugely lower prediction performance than samples with textual answers (Table 3). In creating our filtered datasets, we note that our method tends to eliminate proportionally more common numeric answers than textual ones which increases the overall performance of the filtered versions.

3.3 Similarity Evaluation Method

In order to establish the extent to which general-purpose Transformers are capable of compositionally deriving answers to evaluation questions, it is necessary to eliminate alternative explanations. Such mechanisms include a computation of the answer entirely from reasoning over the input text (abbreviated below as ROIT) of a given evaluation sample. In theory (i.e. ignoring background/commonsense knowledge requirements), this is possible by design in the case of some of our evaluation datasets (e.g. ROPES, NEWSQA and sometimes QASC-IR) but not in others such as PIQA, SIQA, CQA or QASC. This mechanism is itself compositional but differs from the phenomenon of composition over training samples. Another possibility is the derivation of a memorised answer from similar text encountered in masked language pretraining. We controlled for both of these situations by focusing on the difference in performance before and after the addition of the TD and ND tasks.

Noting that any mechanism utilising information from more than one training sample to derive a correct answer to an evaluation question requires some form of composition, we focused on removing the remaining possibility; that an evaluation answer is memorisable from a *single* training example. As discussed earlier, it is challenging to automatically distinguish memorisable training samples, from those that are similar, but upon examination carry a different meaning. Therefore, we instead focused on identifying evaluation samples that have a very low probability of having an answer derivable from a singular training sample. We performed this in three steps:

(1) We ranked training samples in order of similarity to each evaluation sample and assigned each evaluation sample into one of three similarity categories based on the similarity score to its most similar training sample.

To evaluate similarity, we used sentence embeddings produced by the 'sentence-transformers/stsb-roberta-large' model (Reimers and Gurevych, 2019), from the Huggingface library (Wolf et al., 2020). We initially conducted tests to determine whether considering both the question and the answer or just the question is necessary and concluded that considering both is most effective in the diverse question-answering domain we study.

Hence, we adopted a similarity score between each evaluation sample and each training sample as:

$$Sim(e_i, t_j) = \frac{csim\left(e_i^q, t_j^q\right)}{2} + \frac{csim\left(e_i^a, t_j^a\right)}{2}$$

Where $e_i$and t_j are evaluation and training samples, q and a refer to the question and answer components of each and *csim* is the cosine similarity function.

We then categorised evaluation sample similarity to training samples into *Sim* (*100) ranges of 0:60 (least similar), 60:90 (typically not very similar), and 90-100 (usually similar on superficial inspection but not necessarily semantically the same). Overall, we identified very few evaluation-train pairs where the questions have the same meaning and have overlapping answers. However, considering those that we did find, we set the upper bound of the least similar category (60) well below the lowest similarity score of any such example observed (81).

(2) Noting that neither of our prediction scoring methods involve stemming or lemmatisation (i.e. “cousin” will not match “cousins” and “4” will not match “four”), as already discussed we further refined our evaluation sets by eliminating all evaluation samples that have answers with any word overlap with the most similar training sample answer to create filtered datasets.

(3) We then focused on analysing the performance of the remaining samples that are both filtered and that fall into the *least similar* category.

This last step was necessary because there are two remaining possibilities for memorisation; a training example that is not the “most similar” to an evaluation example could nonetheless be memorisable, or alternatively a training sample could have a dissimilar answer, but the evaluation sample question and answer could be buried in the training sample’s input. The chances of this occurring were much reduced in both cases by considering only the items in the least similar category. We completed our analysis by a visual inspection of the remaining items in the least similar category and were unable to identify any memorisable examples.

4 Results and Discussion

All figures reported for the UQA and UQA+TDND models are the mean of three

Evaluation Dataset	Metric	UQA	+ID	+ID +TD	+ID +ND	+ID +TD +ND (UQA+TDND)	UQA → UQA+TDND Change %
DROP	F1	19.66 ±0.39	18.73	22.24	19.73	24.92 ±0.44	**26.74**
DROP-CS	F1	21.05 ±2.13	17.96	23.40	16.63	24.75 ±1.02	**17.60**
MMLU-M	MC	27.59 ±0.38	25.03	28.56	28.04	27.24 ±0.62	-1.25
PIQA	MC	63.49 ±0.82	63.87	64.64	61.81	62.26 ±0.52	-1.94
SIQA	MC	53.47 ±0.80	51.99	51.11	53.49	54.14 ±0.24	1.26
CQA	MC	55.64 ±1.31	54.79	55.77	56.67	55.42 ±0.14	-0.39
QASC	MC	37.69 ±0.97	36.50	37.37	37.58	36.25 ±0.66	-3.82
QASC-IR	MC	57.67 ±0.64	53.56	58.32	59.29	55.72 ±1.42	-3.37
ROPES	F1	41.16 ±1.74	41.19	50.40	42.56	51.88 ±3.06	**26.05**
NEWSQA	F1	57.35 ±1.34	56.49	56.05	58.12	56.57 ±0.90	-1.35

Table 2 Effect on unfiltered Evaluation Dataset Performance of changing the training regime from baseline training datasets (UQA) through adding individual digit tokenisation (+ID), textual numerical literacy (+TD), numeric literacy (+ND), and both (UQA+TDND). ± figures are one standard deviation. Bold items indicate a material change discussed in the text.

training runs. Other figures are single runs. Evaluation datasets are the full (de-duplicated) versions unless denoted with an asterisk* (filtered versions), or with a double asterisk** (filtered and in the least similar category). All figures in tables are the mean prediction performance with bracketed items denoting the corresponding number of samples.

Table 2 indicates the progressive performance difference from the UQA-trained model. Initially we added individual digit tokenisation (+ID) (Geva et al., 2020), adapted to work with the BART tokenizer to mitigate the unwanted effect of sub-word tokenisation on common number patterns. We also tried a 10E-based number representation (Nogueira et al., 2021) but found it lowered performance in our multitask environment. For brevity we omit those results. As expected, adding +ID resulted in a slight diminishment of performance, particularly for evaluation datasets that contain a lot of numbers, as we are changing the distribution of numeric tokens from the initial masked language pretraining. Therefore, we designated the original UQA model trained without +ID as our baseline.

Adding the TD dataset caused a material improvement to DROP, DROP-CS and ROPES. Other datasets were not significantly affected. This included NEWSQA which is similarity to DROP, DROP-CS and ROPES, but in contrast to them usually contains answers derivable from a single span in the input.

Adding ND by itself did not materially affect any dataset excepting a diminishment in DROP-CS performance.

Adding TD and ND in combination results in a large improvement to DROP, DROP-CS and ROPES. In all cases this improvement was slightly larger than when adding TD alone. The overall impact was far higher than on any of the other datasets, which had minimal change from baseline. Considering the nature of DROP and DROP-CS already noted, this suggests that the model had better learned to encode simple numerical strategies. It is less clear that ROPES can benefit from understanding numerical reasoning although it is tempting to ascribe some benefit from this to an ability to perform reasoning over qualitative relations such as "increase" or "less" which occur often in this dataset (Lin et al., 2019). Noting the multi-hop nature of ROPES samples it is just as plausible that improvement related to an improved ROIT strategy learned from TD. For our purposes we are less concerned with the specific strategy learned and more with evaluating a capability to compose such skills whatever they may be, so we leave further exploration of this idea to future work.

Taken across the full datasets, the observed improvement alone did not entail that the model is composing new skills with what it has already learned about natural language. Without further analysis, it could equally be the case that the model had simply seen the necessary answers during training on the additional numerical literacy tasks and the strategy learned was simply to memorise the answer. Therefore, we turned our attention to the filtered versions of our evaluation datasets.

Table 3 illustrates the previously discussed large performance gap between DROP (and DROP-CS) samples with numeric answers and those with textual answers. The superior performance of the 0:60 category compared to 60:90 in Table 4 is because very few samples with numeric answers fall into 0:60. Hence, we do not claim that being

Sim. Cat.	Answer. Type	UQA	UQA +TDND
0:60	Numeric	0.40 (84)	0.00 (5)
	Textual	41.69 (1045)	45.49 (652)
60:90	Numeric	4.11 (1154)	6.60 (1229)
	Textual	37.03 (819)	45.55 (1211)
90:100	Numeric	-	66.67 (4)
	Textual	-	0.00 (1)

Table 3 DROP*: Prediction performance for Numeric versus Textual Answer Types.

Evaluation Dataset	Sim. Cat.	UQA	UQA +TDND
DROP*	0:60	**38.62** (1129)	**45.14** (657)
25.36 → 30.04	60:90	17.77 (1973)	25.93 (2440)
	90:100	-	53.33 (5)
DROP-CS*	0:60	**39.53** (158)	**42.18** (110)
28.13 → 31.18	60:90	17.41 (168)	25.7 (215)
	90:100	-	0.0 (1)
MMLU-M*	0:60	25.3 (307)	24.26 (136)
28.32 → 27.35	60:90	33.52 (178)	28.56 (349)
	90:100	-	-
PIQA*	0:60	60.81 (598)	60.37 (588)
62.74 → 61.63	60:90	72.04 (124)	67.16 (134)
	90:100	-	-
SIQA*	0:60	57.18 (383)	55.05 (373)
58.08 → 56.31	60:90	59.01 (370)	57.54 (380)
	90:100	-	-
CQA*	0:60	56.56 (155)	60.98 (129)
58.74 → 58.33	60:90	60.08 (253)	57.11 (279)
	90:100	-	-
QASC*	0:60	34.04 (142)	33.67 (99)
38.55 → 35.36	60:90	41.71 (203)	36.04 (246)
	90:100	-	-
QASC-IR*	0:60	48.15 (81)	49.07 (72)
56.21 → 52.47	60:90	58.75 (257)	53.38 (266)
	90:100	-	-
ROPES*	0:60	**41.87** (197)	**52.62** (197)
44.86 → 61.49	60:90	47.09 (264)	68.11 (264)
	90:100	-	-
NEWSQA*	0:60	53.15 (770)	51.36 (759)
53.7 → 52.86	60:90	54.07 (1174)	53.82 (1185)
	90:100	-	-

Table 4 Prediction performance on filtered Evaluation Datasets grouped by similarity to most similar training example. Figures under dataset names are the overall mean prediction performance for UQA and UQA+TDND. Bold figures indicate discussion in the main text.

highly dissimilar to any training sample is actually necessary for improved performance, simply that when it is the case, the chances of any improvement relating to memorisation are reduced. We instead focused on the prediction improvement between the UQA and UQA+TDND models.

The number of samples in the 0:60 category often reduces between UQA and UQA+TDND due to cases where exposure to a more similar TD or ND item pushed an evaluation sample into a higher similarity category. Therefore in Table 5 and discussion below we explore whether individual evaluation samples that “move” categories are those that tend to have better prediction performance. We conclude that those that “stay” tend to do better. This eliminates the possibility that items that “moved” were low scoring to begin with and then improved through direct exposure to TD or ND samples.

ROPES* also improves materially between UQA and UQA+TDND similarly to DROP* and DROP-CS*, in both 0:60 and 60:90 categories. A difference is that in contrast to the latter, ROPES* samples in the 60:90 category tend to outperform those in the 0:60 category.

Turning to the other datasets it is variable whether the items in the 0:60 or the 60:90 categories have better prediction performance, but in comparing the same categories between UQA and UQA+TDND, the differences are generally much smaller than the corresponding differences for DROP*, DROP-CS*, or ROPES*. The difference in behaviour between these three and other datasets relates to TD and ND imparting some combination of the numerical reasoning and ROIT strategies that are directly applicable to these datasets, whereas success on the other datasets relates more to a need for alternative strategies.

After adding TD and ND to the training regime, an evaluation sample may or may not then be exposed to a more similar training sample from the newly added datasets. It can be seen in Table 5 that there is often more improvement for evaluation samples that did *not* encounter a more similar training sample.

In the case of DROP** and DROP-CS** it is thus possible to be sure that there are many evaluation examples that have significantly better prediction performance than the overall mean and did not derive this improvement from memorizing a training sample. Without any alternative explanation, we take this as strong evidence that the compositional conjecture we started with is evidenced in the actual model behaviour. ROPES** is slightly less clear-cut in this regard as the small number of samples that “moved” improved by more than those that “stayed”. However, we note that “stayers” also improved by a large amount and did not do so by memorisation.

For the other datasets that would not be expected to benefit from the addition of numerical literacy tasks, we can see that improvement is variable between “stay” and “move” samples, but this is less interesting given that these datasets were not

Evaluation Dataset	Adding +TDND Did Not Add Closer Training Sample	Adding +TDND Added Closer Training Sample
DROP**	**5.53 (351)**	1.80 (778)
DROP-CS**	**8.55 (68)**	-1.52 (90)
MMLU-M**	**1.47 (68)**	0.70 (239)
PIQA**	**0.57 (522)**	-4.39 (76)
SIQA**	**-0.79 (339)**	-10.61 (44)
CQA**	**1.87 (89)**	0.00 (66)
QASC**	-5.56 (54)	**-0.38 (88)**
QASC-IR**	-1.64 (61)	**3.33 (20)**
ROPES**	9.77 (178)	**19.88 (19)**
NEWSQA**	-2.04 (737)	**1.99 (33)**

Table 5 Effect of adding/not adding more similar training example. Samples are from the lowest similarity category (0:60) from filtered datasets.

expected to benefit from the addition of the numerical literacy tasks, whether by memorisation or by learning a strategy to begin with.

5 Conclusion

There has been limited detailed empirical confirmation of the ability of Transformers to compositionally generalise in the natural language question-answering domain. We have built upon much informative prior work to develop a platform for analysing whether performance improvement on unseen datasets from adding disparate new training tasks to an existing multitask training regime can be attributed to memorisation or to a compositional mechanism. In our experiments, we created filtered evaluation datasets containing only samples that are unlikely to have memorisable answers and demonstrated that performance on these samples can be improved in a manner attributable to a compositional mechanism and not to memorisation.
We also began by observing that the simple compositional mechanism that general-purpose Transformers explicitly instantiate could hypothetically provide a basis for an ability to compositionally generalise and we conclude that our experiments provide evidence that it actually does.

References

Jacob Andreas. 2020. Good-Enough Compositional Data Augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online, July. Association for Computational Linguistics.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2019. Systematic Generalization: What Is Required and Can It Be Learned? In *International Conference on Learning Representations (ICLR)*.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 375(1791):20190307.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(05), pages 7432–7439. Association for the Advancement of Artificial Intelligence.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, et al. 2020. Language Models are Few-Shot Learners. *arXiv:2005.14165v3 [cs.CL]*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457v1 [cs.AI]*.

Peter Clark, Oren Etzioni, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, and Michael Schmitz. 2019b. From "F" to "A" on the N.y. regents science exams: An overview of the Aristo project. *arXiv:1909.01958v3 [cs.CL]*.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2021. The paradox of the compositionality of natural language: a neural machine translation case study. *arXiv: 2108.05885v1 [cs.CL]*.

Ishita Dasgupta, Demi Guo, Samuel J. Gershman, and Noah D. Goodman. 2020. Analyzing machine‐learned representations: A natural language case study. *Cognitive science*, 44(12):e12925.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. 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 2368–2378, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. 2021. Memorization vs. Generalization: Quantifying Data Leakage in NLP Performance Evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1325–1335, Online. Association for Computational Linguistics.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: a critical analysis. *Cognition*, 28(1–2):3–71.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. Specialized architectures. *arXiv:2007.08970v2 [cs.CL]*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, et al. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting Numerical Reasoning Skills into Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Christopher Pal. 2020. Measuring Systematic Generalization in Neural Proof Generation with Transformers. In *Advances in Neural Information Processing Systems 33*, Vancouver, Canada.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations (ICLR)*.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality Decomposed: How do Neural Networks Generalise? *The journal of artificial intelligence research*, 67:757–795.

Prabhu Prakash Kagitha. 2020. Systematic Generalization Emerges In Seq2Seq Models With Variability In Data. In *Bridging AI and Cognitive Science (BAICS) Workshop, International Conference on Learning Representations (ICLR)*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, and Others. 2020. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *International Conference on Learning Representations (ICLR)*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A Dataset for Question Answering via Sentence Composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(05), pages 8082–8090. Association for the Advancement of Artificial Intelligence.

Najoung Kim and Tal Linzen. 2020. COGS: A Compositional Generalization Challenge Based on Semantic Interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 9087–9105, Online. Association for Computational Linguistics.

Segwang Kim, Joonyoung Kim, and Kyomin Jung. 2021. Compositional generalization via parsing tree annotation. *IEEE access*, 9:24326–24333.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2873–2882, Stockholm Sweden. PMLR.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning Over Paragraph Effects in Situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences of the United States of America*, 117(48):30046–30054.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2021. Investigating the Limitations of Transformers with Simple Arithmetic Tasks. *arXiv: 2102.13019v1 [cs.CL]*.

Santiago Ontañón, Joshua Ainslie, Vaclav Cvicek, and Zachary Fisher. 2021. Making transformers solve compositional tasks. *arXiv: 2108.04378v1 [cs.AI]*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of machine learning research: JMLR*, 21:1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Lingustics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203, Seattle, Washington. Association for Computational Linguistics.

Jacob Russin, Roland Fernandez, Hamid Palangi, Eric Rosen, Nebojsa Jojic, Paul Smolensky, and Jianfeng Gao. 2021. Compositional processing emerges in neural networks solving math problems. *arXiv:2105.08961v1 [cs.LG]*.

Jacob Russin, Randall C. O'Reilly, and Yoshua Bengio. 2020. Deep Learning Needs a Prefrontal Cortex. In *Bridging AI and Cognitive Science (BAICS) Workshop, International Conference on Learning Representations (ICLR)*.

Jake Russin, Jason Jo, Randall C. O'Reilly, and Yoshua Bengio. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv:1904.09708v3 [cs.LG]*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense Reasoning about Social Interactions. 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 4463–4473, Hong Kong, China, November. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. SyGNS: A Systematic Generalization testbed based on natural language semantics. *arXiv:2106.01077v1 [cs.CL]*.

Pengcheng Yin, Hao Fang, Graham Neubig, Adam Pauls, Emmanouil Antonios Platanios, Yu Su, Sam Thomson, and Jacob Andreas. 2021. Compositional generalization for neural semantic parsing via span-level supervised attention. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823, Online. Association for Computational Linguistics.

6 Appendices

Appendix A. Hyperparameters and other Details

Models: After experimenting with T5-Base (220 million parameters) and T5-Large (770 million parameters) we determine that BART with 440 million parameters is a good trade-off between training speed and performance.

Batch Size: For all experiments reported we use a batch size of 32 with two gradient accumulation steps.

Steps: For all reported experiments we take the best model after training for 150,000 steps (batches) irrespective of the number of tasks in the particular training mixture.

Learning Rate: All experiments have an initial learning rate of 2e-5 with a linear decay to zero over 250,000 steps.

Sequence Length: We use a maximum input sequence length of 512 and a maximum output sequence length of 100.

Hardware: We train each model on a single machine running Ubuntu 20.04 LTS with 768 GB of RAM. We utilise two RTX8000 GPU cards for all training runs.

Training Time: Each model in the above configuration takes approximately 80 hours to reach 150,000 steps.

Appendix B. ND and TD Dataset formatting examples

ND Example:

What is 13441 + 3068? \n **Answer: 16509**

TD Example:

How many more urban families were in the country than Spanish families ? \n There were 522 urban families in the country . The commander executed

644 Japanese families . The commander appointed 411 Spanish families in the city . The commander appointed 942 urban families and the military appointed 1592 urban families . The military borrowed 1179 English families from the commander . **Answer: 111**

Appendix C. Challenges in using sentence embedding similarity to determine memorisability.

Considering the following memorisable QASC example which has similarity score of 95.14 against the most similar training example:

What is a tool for indicating air pressure? \n (A) rain guage (B) vibration (C) seismograph (D) lamphreys (E) barometer (F) Otoacoustic (G) thermometer (H) weater **Answer: barometer**

Most similar training example (from the REGENTS easy dataset):

Which weather instrument measures air pressure? \n (A) thermometer (B) anemometer (C) rain gauge (D) barometer **Answer: barometer**

However after additional retrieved text is added to the same example in QASC-IR, the additional paragraphs obscure the original meaning of the example such that the similarity score is now only 81.04 (noting though the most similar training example is still correctly identified as the same above REGENTS example):

What is a tool for indicating air pressure? \n (A) rain guage (B) vibration (C) seismograph (D) lamphreys (E) barometer (F) Otoacoustic (G) thermometer (H) weater\nThermometer barometer and hygrometer give the complete weather picture. … Otoacoustic emissions are sounds the ear generates. **Answer: barometer**

Appendix D. Evaluation and Similar Training Samples

All samples in this section are from the filtered versions of evaluation datasets.

Evaluation Sample	**Most Similar Training Sample**
DROP*: Which quarter were the only touchdowns scored during? \n Hoping to rebound from their tough road loss to the Ravens the Chiefs played their Week 2 home opener against their AFC West foe the Oakland Raiders. Kansas City would score in the first quarter as rookie kicker Ryan Succop got a 23-yard field goal. In the second quarter the Raiders tied the game as kicker Sebastian Janikowski made a 48-yard field goal. Oakland would take the lead in the third quarter as Janikowski nailed a 54-yard field goal. In the fourth quarter the Chiefs would retake the lead as quarterback Matt Cassel completed a 29-yard touchdown pass to wide receiver Dwayne Bowe. However the Raiders sealed the win as running back Darren McFadden got a 5-yard touchdown run. **Answer: fourth**	TD: How many rushing touchdowns did Jaguars' quarterback completed ? \n Jaguars' quarterback completed 23 passing yards and 3 impressive wins . Eagles' receiver had 30 points Manning had 33 points and Jaguars' quarterback had 26 points . Manning completed 13 field goal yards and 4 tight wins . Jaguars' quarterback completed 4 rushing touchdowns and 34 field goal yards . Jaguars' quarterback completed 5 impressive wins . **Answer: 4**
DROP-CS*:How many yards was Jason Elam's second shortest field goal? \n Coming off their divisional road win over the Texans the Colts went home for an intraconference duel with the Denver Broncos. In the first quarter Indianapolis trailed early with Broncos kicker Jason Elam getting a 35-yard field goal while QB Jay Cutler 7-yard TD pass to WR Brandon Marshall. In the second quarter the Colts would respond with RB Joseph Addai getting a 14-yard field goal. Denver tried to increase its lead with Elam kicking a 22-yard field goal. Indianapolis would take the lead with QB Peyton Manning completing a 9-yard TD pass to TE Dallas Clark. In the third quarter the Colts began to dominate with Manning getting a 1-yard TD run. He would also hook up with Clark again on a 3-yard TD pass. The Broncos' only response was Cutler's 2-yard TD run. In the fourth quarter Indianapolis managed to put the game away with Manning's 5-yard TD pass to WR Reggie Wayne along with kicker Adam Vinatieri nailing a 22-yard field goal. **Answer: 35**	TD: How many passing yards did Dolphins nailed ? \n Dolphins nailed 36 passing yards and 5 tight wins . Vikings nailed 33 rushing yards in Pittsburgh . Dolphins drove 4 tight wins in Pittsburgh . Vikings drove 7 field goals and Dolphins drove 5 field goals . Lions nailed 47 rushing yards and Vikings nailed 21 rushing yards . **Answer: 36**

Table 6 Most similar evaluation-training pairs in the highest similarity category (90:100).

Evaluation Sample	**Most Similar Training Sample**
DROP*: Which kicker made more field goals? \n Coming off their home win over the Texans the Titans stayed at home for a Week 4 interconference duel with the Minnesota Vikings. In the first quarter Tennessee drew first blood as kicker Rob Bironas got a 20-yard field goal along with rookie RB Chris Johnson getting a 1-yard TD run. In the second quarter the Vikings responded with RB Adrian Peterson getting a 28-yard TD run. Afterwards the Titans answered with Bironas kicking a 32-yard field goal along with RB LenDale White getting a 1-yard TD run. Minnesota closed out the half with kicker Ryan Longwell getting a 42-yard field goal. In the third quarter Tennessee increased its lead with Bironas nailing a 49-yard field goal. In the fourth quarter the Vikings tried to rally as Peterson got a 3-yard TD run yet the Titans pulled away with	TD: How many running yards did Lions completed ? \n 5 impressive wins 38 field goal yards and 25 points were fired in Chicago . Lions completed 28 running yards . Houston threw 20 field goal yards and 2 tight wins . **Answer: 28**

Johnson getting a 6-yard TD run. With the win Tennessee acquired its first 4-0 start in franchise history. **Answer: Rob Bironas**	
DROP-CS*: Which receiver got the Giants first and second TD? \n The Giants opened their new home in search of revenge against the Panthers who had soundly defeated them in the last game at Giants Stadium. In the first quarter Carolina scored the stadium's first points as kicker John Kasay got a 21-yard field goal. New York would answer with the stadium's first touchdown as quarterback Eli Manning found wide receiver Hakeem Nicks from 26 yards out. The Panthers would retake the lead in the second quarter as Kasay made field goals from 52 and 43 yards. Manning found Nicks again on a 19-yard touchdown pass with less than a minute left in the first half but Carolina quarterback Matt Moore completed a 19-yard touchdown pass to wide receiver Steve Smith with six seconds remaining. The Giants would get back on top in the third quarter as kicker Lawrence Tynes nailed a 32-yard field goal followed by Nicks' third touchdown of the game (a 6-yard catch). In the fourth quarter the Giants added one more touchdown as running back Ahmad Bradshaw ran for a 4-yard score. Carolina's Greg Hardy blocked a Matt Dodge punt out of the end zone to round out the scoring with a safety. The Giants' historic win had come with a price however; tight end Kevin Boss left the game in the first quarter with a concussion and Will Beatty who filled in for Boss afterward was benched with a broken foot. The Giants signed tight end Bear Pascoe from their practice squad to play against the Colts. **Answer Hakeem Nicks**	TD: Who had less field goals Eagles' receiver or Brady ? \n 19 running yards 4 tight wins and 3 running touchdowns were got in Pittsburgh . Eagles' receiver fired 9 field goals and 51 passing yards . Patriots fired 2 impressive wins . Patriots threw 12 points . Brady fired 8 field goals and Eagles' receiver fired 4 field goals . **Answer: Brady**
ROPES*: Which spot should Allan take his family to have a better chance to view limestone formations? \n About 10% of sedimentary rocks are limestones. The solubility of limestone in water and weak acid solutions leads to karst landscapes in which water erodes the limestone over thousands to millions of years. Most cave systems are through limestone bedrock. Allan has to plan a couple of adventures this year. One adventure involves taking his family on vacation and his son has been interested in seeing different formations of limestone. The other adventure Allan must plan for is a trip with his coworkers one of which has mentioned that they have seen all the limestone formations that they want to see and want to see other rock formations. He has narrowed down his adventure spots to Wilson Caves and Mt. Everest. **Answer: Wilson Caves**	RACE: What is the best title for the story? \n (A) Father and Son (B) A Father's Wish (C) Catching Crabs (D) Tips for Job Hunting \n "So?"he said."Er...so what?""So what do you really want to do?"he asked. My father was a lawyerand I had always assumed he wanted me to go to law schooland follow his path through life."I want to traveland I want to be a writer."I replied. This was not the answer he would expect."Interesting idea"he said."I kind of wish I'd done that when I was your age."I wailed. "You have plenty of time.You need to find out what you really enjoy now.Lookit's late. Let's take the boat out tomorrow morningjust you and me. Maybe we can catch some crabs for dinnerand we can talk more." Early next morning we set off along the coast. We didn't talk muchbut enjoyed the sound of the seagulls and the sight of the coastline and the sea beyond. There was no surf on the coastal waters at that time."Let's see if we get lucky"he saidpicked up a mesh basket with a rope attached and threw it into the sea. We waited a whilethen my father stood up and said"Give me a hand with this"and we pulled up the crab cage onto the deck. The cage was filled with dozens of soft shell crabs."Why don't they try to escape?" "just watch them for a moment. Look at that

	onethere!He's trying to climb outbut every time the other crabs pull him back in"said my father. After several timesnot only did the crab give up its struggle to escapebut it actually began to help stop other crabs trying to escape.He'd finally chosen an easy way of life. Suddenly I understood why my father had suggested catching crabs that morning. He looked at me. "Don't get pulled back by the others"he said."Spend some time figuring out who you are and what you want in life.Think about what's really important to youwhat really interests youwhat skills you have.If you can't answer these questions nowthen take some time to find out. Because if you don'tyou'll never be happy." My father started the motor and we set off back home. **Answer Catching Crabs**
QASC*: What is a bolus? \n (A) moistened food (B) SI units (C) a producer (D) unicellular organisms (E) precipitation (F) Fractions (G) holding nutrients (H) measuring device **Answer: moistened food**	ND: What is argmax(reflectional 10928.9 audiology 6019 moist 17187.0)? \n **Answer: moist**
PIQA*: Turn any cup into a travel cup \n (A) use press and seal to make a super tight seal at the top of your cup (B) use press and seal to make a super tight opening at the top of your cup **Answer: use press and seal to make a super tight seal at the top of your cup**	SQUAD1.1: How is a vacuum created inside of a manual water pump? \n (Vacuum) To continue evacuating a chamber indefinitely without requiring infinite growth a compartment of the vacuum can be repeatedly closed off exhausted and expanded again. This is the principle behind positive displacement pumps like the manual water pump for example. Inside the pump a mechanism expands a small sealed cavity to create a vacuum. Because of the pressure differential some fluid from the chamber (or the well in our example) is pushed into the pump's small cavity. The pump's cavity is then sealed from the chamber opened to the atmosphere and squeezed back to a minute size. **Answer: a mechanism expands a small sealed cavity**

Table 7 Randomly selected evaluation-train pairs after filtering that are in the least similar category (0:60).