What a Corpus Teaches

Tasks a language model learns without labels

A report from Wrennick Research

Summary

Question answering, translation, reading comprehension and summarisation are normally taught with labelled data for each task.
This report shows that a language model picks up a good deal of each from unlabelled text alone, once it is trained on a new corpus of several million web pages called Longfield.
Given a passage and a question, the answers it writes reach 41.8 F1 on one reading set, ahead of two of the three systems built for that set, while using none of its 96,000 labelled examples.
Capacity matters: every task improves in a log-linear way as the model grows.

Prepared for software testing. The text is fictional.

1 Introduction

A language model is trained on one job: given the tokens so far, say what comes next. That job needs no labels beyond plain text.

The question this report asks is whether such a model, given enough text and enough capacity, starts doing tasks nobody trained it for, once the task is described in the prompt.

A language model can carry out tasks it was never trained on, in a zero-shot setting – with no change to its parameters or architecture.

The model and the training loop are kept as plain as possible; the effort goes into the corpus, which is assembled with a simple rule for finding pages a person would actually read.

2 Building the corpus

The corpus is built from the open web, and the model is trained on it with a single objective: guess the next token.

To gather it, we followed outbound links from a discussion board and kept the pages that had drawn at least 5 recommendations,
as a rough sign that people found them worth reading. The corpus holds just over 5 million documents, some 23 GB of text,
once repeats and pages whose text could not be extracted were removed.

Table 1: The corpus by the year its pages were linked.

Period	Documents	Text
2015	1.2 M	5.2 GB
2016	1.3 M	6.0 GB
2017	1.4 M	6.4 GB
2018	1.1 M	5.4 GB
Total	5.0 M	23.0 GB

2.2 Reading bytes

A byte-level alphabet keeps the vocabulary small and means no text ever meets an unknown token.
The vocabulary is 41,209 tokens, and the context window grows from 640 to 1152 tokens, so that longer passages, which the comprehension and summarisation sets need, fit inside one window.
A byte-level alphabet lets the model read any language without an unknown-token entry, keeps the vocabulary small, and means new text needs no preprocessing.

3 What was measured

Each evaluation writes its input as a prompt in ordinary language, and the answer is whatever the model continues with. Nothing is trained on the task itself.

Results are reported on the usual benchmarks for each task, beside systems that were trained on that task's labels where such numbers exist.

Gains track model size, roughly linearly in the logarithm of the parameter count over the range trained here.

Table 2: The four sizes trained for these measurements.

Model	Parameters	Blocks	Width
Small	95 M	12	704
Medium	280 M	22	960
Large	610 M	34	1216
Largest	1.1 B	44	1536

3.6 Summarisation

Summaries are produced by asking for a shorter version of a news story; nothing in the model was trained on summaries.

To make the job clear, we place the cue TL;DR: after the article
and read what follows as the summary. That one cue is usually enough to get a short abstractive summary.

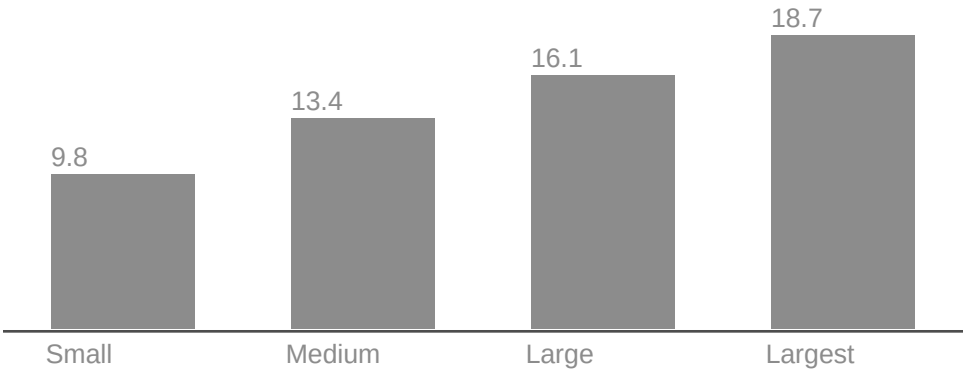
Summarisation is harder, because something has to be left out. A cue at the end of the prompt helps, and the summaries read well though they often miss the point.

3.7 Translation

Translation is weakest, and it works best between languages that commonly appear together in the corpus, which suggests the model is drawing on parallel text it has seen.

The corpus was tested by training the same model on a smaller curated set; the broader corpus gave lower loss on held-out text from many sources.

On a small sample the model's answers were compared with a person's: the topic is usually right and the detail is sometimes wrong.



Zero-shot translation score by model size, out of 100.

3.1 Language modelling

Failures are reported too: long documents, arithmetic, and anything needing knowledge from outside the training text.
A self-attention block costs the square of the context length, which caps how much text the model can read at once in our setup.

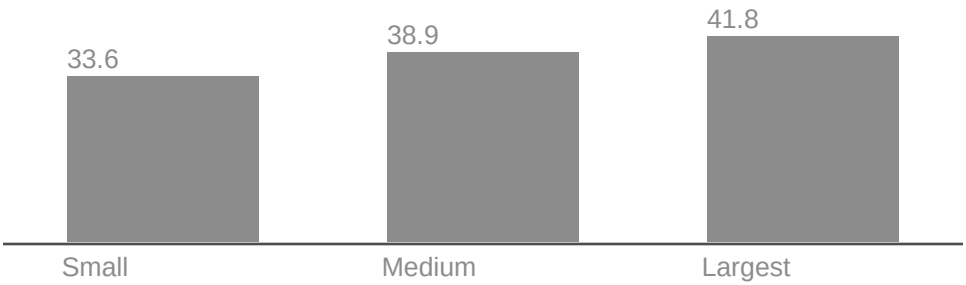
Table 3: Held-out loss on four collections, by model size.

Collection	Small	Medium	Largest
News	3.58	3.19	2.90
Books	3.26	2.98	2.72
Web	3.81	3.40	3.11
Code	3.02	2.70	2.46

3.5 Reading comprehension

No fine-tuning is done for any evaluation. Every number below comes from a hand-written prompt and a continuation read off the model.

Reading comprehension is the strongest of the three: the model answers with the span that follows the question, and its score is set beside systems trained on the set.



F1 on a reading comprehension set in the zero-shot setting, by model size.

3.2 Children's book test

A language model is trained on one job: given the tokens so far, say what comes next. That job needs no labels beyond plain text.

The question this report asks is whether such a model, given enough text and enough capacity, starts doing tasks nobody trained it for, once the task is described in the prompt.

To a degree it does. Reading comprehension, summarisation and translation all improve without any change to the parameters or the architecture for each task.

3.3 LAMBADA

The model and the training loop are kept as plain as possible; the effort goes into the corpus, which is assembled with a simple rule for finding pages a person would actually read.
Each evaluation writes its input as a prompt in ordinary language, and the answer is whatever the model continues with. Nothing is trained on the task itself.
Results are reported on the usual benchmarks for each task, beside systems that were trained on that task's labels where such numbers exist.

3.4 Winograd

Gains track model size, roughly linearly in the logarithm of the parameter count over the range trained here.

Failures are reported too: long documents, arithmetic, and anything needing knowledge from outside the training text.

The corpus was gathered by taking outbound links from a discussion board and keeping pages that its readers had recommended, a cheap stand-in for what people choose to read.

3.8 Question answering

Repeated pages, pages that were mostly markup, and pages with too little text were dropped; the rest became one large unlabelled corpus.

A byte-level alphabet lets the model read any language without an unknown-token entry, keeps the vocabulary small, and means new text needs no preprocessing.

The model is a stack of self-attention blocks in the established style, with normalisation moved to the input of each block, and it is trained with one objective throughout.

4.1 Model capacity

No fine-tuning is done for any evaluation. Every number below comes from a hand-written prompt and a continuation read off the model.

Reading comprehension is the strongest of the three: the model answers with the span that follows the question, and its score is set beside systems trained on the set.

Summarisation is harder, because something has to be left out. A cue at the end of the prompt helps, and the summaries read well though they often miss the point.

4.2 Data overlap

Translation is weakest, and it works best between languages that commonly appear together in the corpus, which suggests the model is drawing on parallel text it has seen.

The corpus was tested by training the same model on a smaller curated set; the broader corpus gave lower loss on held-out text from many sources.

A model can be fluent and wrong. That is discussed below, and fluency is never taken as evidence of accuracy.

4.3 Robustness

A smaller version of the model and a description of the data pipeline are released so that others can repeat the training and study the behaviour.
A self-attention block costs the square of the context length, which caps how much text the model can read at once in our setup.
Performance is reported against parameter count, and the trend is smooth on every task, with no sudden jump at any particular size.

4.4 Bias

On a small sample the model's answers were compared with a person's: the topic is usually right and the detail is sometimes wrong.
The results suggest that the size of the data and of the model matters more than task-specific design, at least for the tasks measured, and that the trend deserves more testing.
The report closes with the failure modes we could see and with a note on how to measure them properly, including held-out sets that were not on the public web when the corpus was collected.

5 Discussion

A language model is trained on one job: given the tokens so far, say what comes next. That job needs no labels beyond plain text.

The question this report asks is whether such a model, given enough text and enough capacity, starts doing tasks nobody trained it for, once the task is described in the prompt.

To a degree it does. Reading comprehension, summarisation and translation all improve without any change to the parameters or the architecture for each task.

5.1 Limitations

The model and the training loop are kept as plain as possible; the effort goes into the corpus, which is assembled with a simple rule for finding pages a person would actually read.
Each evaluation writes its input as a prompt in ordinary language, and the answer is whatever the model continues with. Nothing is trained on the task itself.
Results are reported on the usual benchmarks for each task, beside systems that were trained on that task's labels where such numbers exist.

5.2 Future work

Gains track model size, roughly linearly in the logarithm of the parameter count over the range trained here.

Failures are reported too: long documents, arithmetic, and anything needing knowledge from outside the training text.

The corpus was gathered by taking outbound links from a discussion board and keeping pages that its readers had recommended, a cheap stand-in for what people choose to read.

6 Evaluations of the largest model

Repeated pages, pages that were mostly markup, and pages with too little text were dropped; the rest became one large unlabelled corpus.

A byte-level alphabet lets the model read any language without an unknown-token entry, keeps the vocabulary small, and means new text needs no preprocessing.

The model is a stack of self-attention blocks in the established style, with normalisation moved to the input of each block, and it is trained with one objective throughout.

6 Closing remarks

A large model trained on a broad corpus can do several tasks without task-specific training, and its quality rises smoothly as it grows.
The failures are reported as well as the successes; measuring them properly comes next, since fluency makes a mistake easy to miss.

References

[1] A. Author, B. Author and C. Author. Weighting keys for sequence models. A technical report, 2017.

[2] A. Author and B. Author. Byte-level alphabets for large corpora. A technical report, 2018.

[3] A. Author, B. Author, C. Author and D. Author. Reading comprehension with pretrained models. A technical report, 2018.

[4] A. Author. A survey of zero-shot transfer. A technical report, 2019.

Appendix A: Prompts

The prompts used for each evaluation are listed here exactly as they were given, with one sample continuation beside each.
Every prompt closes with a short cue naming the task in everyday words, such as a question mark for questions and a separator for summaries.

Table 4: The prompts used, by evaluation.

Evaluation	Cue	Answer read
Reading comprehension	Question about the passage	The text after the question
Summarisation	TL;DR: after the article	The next two sentences
Translation	A source sentence, then =	The sentence after the equals sign
Question answering	Q: ... A:	The text after A: