Do Language Models Understand Frequent Words Better?

Eliav Mor
moreliav@mail.tau.ac.il

Abstract

Pre-trained language models (LMs) have recently demonstrated outstanding results across a variety of tasks. However, it remains unclear precisely what knowledge the LM manages to capture during pre-training and how word frequency in the training corpus affects the acquisition of knowledge about these words. In this work we explore the correlation between word occurrence in the language and general world knowledge acquired by a Language Model during pre-training. We propose a framework for testing this subject, using a downstream "Yes/No" QA task. The model (in our case T5) is trained on a large set of questions and then tested on its knowledge of relations between different animals and their properties, e.g. "Does a fish have wings?". Our findings show positive correlations between: (a) word occurrence and the accuracy of answers for this word. (b) co-occurrence of two words and a tendency of the model to answer "Yes" for questions about their relation. We also created a classifier for predicting the answer of the model for animal/property relation questions based on the co-occurrence rate of these two words.

1 Introduction

In the field of Natural Language Processing, neural models have been shown to succeed in a variety of tasks such as question answering, translation, summarizing and more. Research exploring what is captured by the contextualized representations that these LMs compute reveals that they encode substantial amounts of syntax and semantics (Linzen et al., 2016; Shwartz and Dagan, 2019; Coenen et al., 2019).

However, the understanding and formulation of their limitations remains an open issue. For example, Talmor et al. (2020a) show that when RoBERTa-L (Liu et al., 2019) is asked to predict a completion for the phrase "Cats usually drink [MASK].", the top completion is "coffee", a frequent drink in the literature it was trained on, rather than water. Talmor et al. (2020a) shed light on the field of understanding language models' capabilities and limitations using systematic probes designed to answer the question: What symbolic reasoning capabilities are difficult to learn from an LM objective only?

One of the probes used by Talmor et al. (2020a) is a test of sentence completion of a finite set of adverbial modifiers ("always", "often", "sometimes", "usually" and "never") where the tested sentences convey general **world knowledge** such as "rhinoceros [MASK] have fur". Results showed that the tested LM performs poorly. A possible explanation for this could be one of the following:

1. **Lacking linguistic abilities** - The model does not capture the semantic meaning of the adverbial modifiers. Hence, it randomly chooses an adverbial to complete the sentence.

2. **Lack of world-knowledge** - The model lacks seemingly trivial common knowledge. Unlike humans, LMs gain their world knowledge only through text processing (which happens during training). They learn from statistical occurrences of words across their source corpora (Bengio et al., 2003). Humans, on the other hand, gain their world-knowledge through use of multiple senses such as vision, hearing and touch. Hence, it is possible that some of the trivial world-knowledge (such as the fact that a rhinoceros does not have fur) is not documented in textual form. This fact makes it hard for language models to gain an understanding of the world equivalent to that of humans. This phenomenon is known as reporting bias (Gordon and Van Durme, 2013).

In this work, we investigate the effect of word occurrence and co-occurrence on the model's world knowledge, focusing on animals and their properties. Our basic hypothesis is that due to pre-training objectives (especially, masked sentence completion a.k.a MLM (Devlin et al., 2019)), the LM acquires "world knowledge" that is highly dependent upon the distribution of words in the corpus. For example, the word "drinks" is usually followed by "coffee". This co-occurrence likely affects the model's knowledge about drinking. Our research was done by utilizing the power of transfer learning on T5 (Raffel et al., 2019) for a downstream task of Yes/No QA that allowed us to extract pre-training "world knowledge" from the fine-tuned LM. Our code and dataset is available on GitHub [1].

2 Background

These days, Transformers (Vaswani et al., 2017) are one of the most powerful architectures for training state-of-the-art LMs. This is due to the fact that their architecture enables sequential data processing in parallel and thanks to their self-attention mechanism. Thanks to these abilities, transfer-learning has become a common methodology for training language models in a variety of NLP tasks, including question answering. Even though nowadays the use of Transformers and transfer-learning is very common in the NLP community, the limitations of LMs - especially their world-knowledge acquired during pre-training and transferred in fine-tuning - remains a mystery. In this work, we developed a systematic method for testing the world-knowledge acquired by a language model during pre-training. Ultimately we present our method's results on T5, a state-of-the-art text-to-text transfer Transformer (Raffel et al., 2019), and present a few possible explanations for our findings.

2.1 Transfer Learning

Transfer learning (Bozinovski, 2020) is a training methodology used in many fields of machine learning. This method focuses on utilizing knowledge of one model trained on a certain task for training a second model to perform a different but related task. For example, using a model that was trained on detecting cars in order to detect trucks. The training of the first model is called pre-training, and that of the second model is called fine-tuning.

[1] https://github.com/eliavmor-tau/UnderstandLMWordKnowledge

Pre-training is the phase when a model is trained on a huge amount of data to perform a specific task. During training, the model is considered to acquire fundamental knowledge which is not necessarily embedded explicitly in the training task, but is necessary for succeeding in this task. For example, a language model that was successfully taught to perform sentence completion had to learn language capabilities such as understanding sentence syntax, word semantics, and certain facts about the world.

Fine-tuning is the concept of using a pre-trained model for training a second model on a different task, usually called a "downstream" task. This method utilizes the original model's knowledge in order to speed-up training and improve performance on small data-sets.

2.2 T5

T5, text-to-text transfer Transformer (Raffel et al., 2019), is a unified framework that converts all text-based language model problems into a text-to-text format. The architecture of T5 is relatively similar to the original Transformer's. T5 was pre-trained on a huge corpus of web-scraped data called C4 ("Colossal Clean Crawled Corpus"). The large versions of T5 (T5-3B and 11B) have been shown to achieve state-of-the-art results in 17 of 24 well-known benchmarks in the NLP community.

Thanks to the above mentioned achievements and its simplicity in adaptation for question answering tasks, T5 was the ultimate choice of language model for performing our analysis.

3 Method

We now present a systematic method for analysis of an LM's acquisition of specific world-knowledge during pre-training. For example, this method can allow us to check whether or not the model *"knows"* that a bird has wings.

Stages:

- **Generate test questions:** Create a set of Yes/No questions/statements that test knowledge of the chosen field. Questions can be made using a pre-defined template, such as: `"A [ENTITY] has [PROPERTY]"`. Additionally, in order to make sure that the model's answer to a given question is not coincidental, each question should be paraphrased.

#	Question
0	Does a [animal] fly
1	Can a [animal] fly
2	A [animal] flies

Table 1: Example of paraphrasing.

In this way, the test set will contain multiple wordings of each question (See Table 1).

- **Generate training questions:** Create a set of Yes/No questions/statements that does not contain information on the tested knowledge.
- **Fine tuning:** Fine tune the tested LM on a downstream task of Yes/No question answering using the training set.
- **Testing:** Check LM's answers on the test set. (See 3.1).

3.1 Measuring LM's World Knowledge

LM's knowledge is measured using a "full confidence" approach. The LM is considered fully confident regarding an animal/property pair iff it gives the same answer to all wordings/versions of the corresponding question, i.e all answers are Yes or No. The model's "knowledge" of this pair can then be classified as either correct or incorrect. Otherwise, its knowledge regarding the given pair is considered unknown.

4 Data Set

Training and Validation Set For training the model, we used the following two data sets merged together: ConceptNet triples from oLMpics (Talmor et al., 2020a) turned into True/False statements, and 20Q as used in Leap-Of-Thought (Talmor et al., 2020b). These data sets include questions such as: *"Is piano human?"* (No), and *"Does car use gas?"* (Yes). Training on these questions teaches the model to retrieve real world facts from its internal implicit knowledge (Talmor et al., 2020b).

From this data set, we filtered out all questions containing words stemming from the names of animals and properties we tested. e.g. for the property "flies" we also removed questions containing the words fly and flying, for nouns we removed questions with both singular and plural forms. Ultimately, the training set contained 118,878 questions, half of their answers being yes and half no. We split the data set of questions as follows: 80% training set and 20% validation set (preserving the Yes/No ratio in each subset).

Test Set We manually created a data set comprised of 878 distinct (animal, property) pairs, with multiple wordings of questions about each one (paraphrasing, see Table 1). We did this by choosing 9 properties (e.g. "has wings", "lives underwater", "has fur"), and assigned to each one 80-100 animals, such that half of them have the property and half don't. We chose animals whose names are a single word (for convenience purposes). Overall, our test set contained **3108** questions (see Table 2 for examples).

#	Question	Answer
0	A whale has feathers	No
1	Does a puma fly	No
2	Does a swan have a horn	No
3	Can a crow fly	Yes
4	A tiger has fur	Yes
5	A prawn lives underwater	Yes

Table 2: Examples of test questions and statements.

5 Experiments

We fine-tuned T5-base on a downstream task of Yes/No question answering on the training set described above. Our model reached a validation accuracy of 85%, where 55% of the answers were "No" and 45% were "Yes". We then tested the LM's world-knowledge on the test set and looked for statistical correlations with word occurrence and co-occurrence in the language. Since C4 dataset is incredibly large, we used Wikimedia dump [2] instead, as a representation of the distribution of words in the language (and specifically in C4).

5.1 Hypotheses

To elaborate and clarify our intention for this study, we present the hypotheses that guided us.

- **Effect of Word Occurrence:** The more frequent the occurrence of an animal/property, the higher the accuracy of the model in answering questions including this word.
  Rationale: Due to the statistic nature of LM pre-training, we expect that the more frequently a word appears, the more likely the model is to understand the meaning of this

[2] https://dumps.wikimedia.org/enwiki/20210401/

word. Thus, we suppose the LM will more likely answer questions including this word correctly.

- **Effect of Word Co-occurrence:** The more frequent the occurrence of the particular `(animal,property)` pair, the more likely the LM is to answer `Yes`.
  Rationale: If the `(animal,property)` pair's co-occurrence is high, it could be for one of two reasons:
  - The property is related to the animal - we expect the model to have acquired this knowledge, therefore it will probably answer `Yes`.
  - The property is *negatively* related to the animal (the animal does not have this property) - it has been shown that language models do not take into account the presence of negation Talmor et al. (2020a). Therefore, the model will probably answer `Yes`.

  All in all, in both cases, we assume the LM will understand the animal and property as being related, and therefore will answer positively.

Disclaimer: Additional factors that aren't expressed in our experiments may affect the model's answers. For instance, the model may acquire knowledge that an `animal` has `property A` because it knows this animal has a related `property B`, which is not a part of our data set. Thus, low occurrence or co-occurrence does not necessarily result in a lack of knowledge.

5.2 Effect of Word Occurrence

An estimation of word occurrence was done using a unigram model of Wikipedia. This was achieved by removing all punctuation and special symbols from the text, and then counting appearances of words in the corpus.

Next, correlation between word occurrence and LM accuracy was measured separately for animals and properties. i.e. when testing animals, we checked for each animal which portion of the tested properties the LM answered correctly about. We used the "full confidence" approach described in 3.1 for each animal/property pair, such that each gets counted as 1 or 0, and the ratio is calculated out of only fully confident answers.

$$Accuracy = \frac{\#\,Full\,Confidence\,Correct\,Answers}{\#\,Full\,Confidence\,Answers} \quad (1)$$

5.3 Effect of Word Co-occurrence

An estimation of each `(animal, property)` pair's co-occurrence was done using the Wikipedia corpus. We define the co-occurrence of an `(animal, property)` pair as the count of the appearance of these two words in the corpus at a distance of at most 512 characters. We decided on this range as a reasonable distance between related words. To calculate the co-occurrences, we first removed all punctuation and special symbols from the corpus, then counted the co-occurrences. Next, correlation between word co-occurrence and LM's tendency to answer Yes was measured as follows. We used the "full confidence" method as before, i.e. we counted only `(animal, property)` pairs for which the model answered consistently. We then calculated the distribution of the model's answers (Yes and No) given the co-occurrence.

6 Results and Analysis

After collecting the model's answers to all questions from the test set, we removed all pairs the model did *not* answer with "full confidence". As a result, 579 of the tested `(animal, property)` pairs remained, which are 66% of the total test set. The following analysis is based on this set of answers.

6.1 Effect of Word Occurrence

As can be seen in Figure 1, there seems to be a positive correlation between the occurrence of properties and the accuracy of the model's answers.

In Figure 2, the results are much more noisy, and the correlation between animal occurrence and model's accuracy is less clear. Using linear regression, we can see a slight positive correlation. In occurrence counts up to 40,000, the accuracy of the model is very scattered, and no trend can be seen. For higher occurrences, the accuracy seems to improve relatively consistently, but there are very few data points in this range. Thus, we can't reach a definite conclusion.

In addition, the coloring in Figure 2 shows that the animals for which the model answered the least questions with "full confidence" are the ones with the lowest occurrence rates. This implies that the

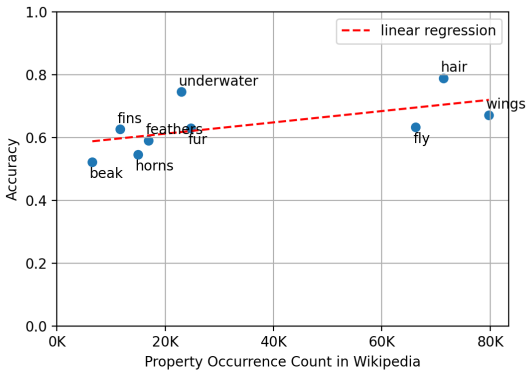

Figure 1: Effect of property occurrence on LM's accuracy. Each point represents the accuracy of the model's answers about a certain property.

model is more confident in its answers about animals that appear more frequently.

Overall, these results support our hypothesis that the more frequent a word is in the language, the better the model understands the "sense" of it.

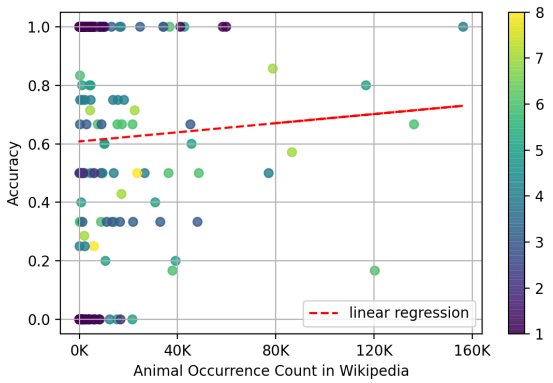

Figure 2: Effect of animal occurrence on LM's accuracy. Each point represents the accuracy of the model's answers about a certain animal. Color represents the number of properties for which the model answered with "full confidence" (regarding each animal).

6.2 Effect of Word Co-occurrence

Figures 3, 4 show a correlation between the co-occurrence of `(animal, property)` pairs and the portion of "Yes" answers (by the LM). As the co-occurrence rises, so does the portion of positive answers. In Figure 4, the aggregation of the model's answers into buckets is used to calculate the proportion of "Yes"'s across pairs of a certain range of co-occurrence counts. This shows clearly the positive correlation, which supports our hypothesis.

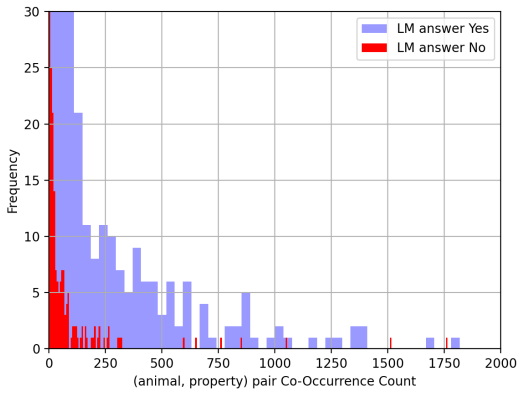

Figure 3: Effect of co-occurrence on Yes/No Answer distribution (out of "fully confident" answers). Omitted counts above 2000 and frequencies above 30 (for sake of clarity) - these appear in Table 2.

Additionally, we constructed a linear classifier that predicts the model's Yes/No answers for a given `(animal, property)` pair's co-occurrence count using a threshold. i.e. if the count is greater than the threshold, the classifier predicts "Yes", otherwise it predicts "No". Our classifier achieved a 68% accuracy on the test set when using a threshold of 17 co-occurrences.

Overall, these results support our hypothesis that for frequent co-occurrences of pairs, the model is more likely to answer "Yes".

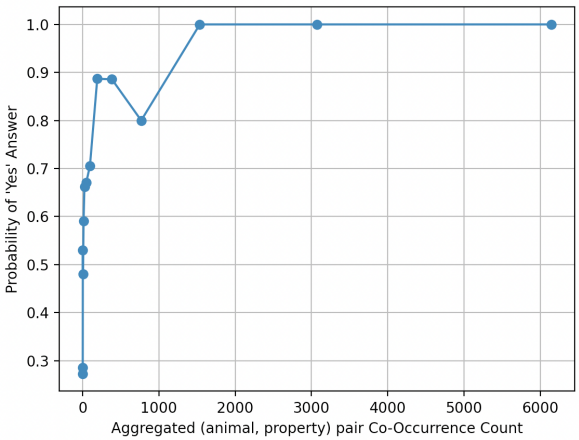

Figure 4: Effect of animal/property pair co-occurrence on probability of "Yes" answers (by the LM). Pairs are grouped into 14 buckets containing values from exponentially growing ranges (due to the distribution of the occurrence counts). Each point represents the center of a bucket. See Table 3 for detailed ranges.

Bin range	Pairs count	"Yes" rate
0-1	70	0.28
1-2	11	0.27
2-4	49	0.53
4-8	52	0.28
8-16	83	0.59
16-32	77	0.66
32-64	76	0.67
64-128	51	0.71
128-256	53	0.88
256-512	35	0.88
512-1024	15	0.8
1024-2048	3	1.0
2048-4096	3	1.0
4096-8192	1	1.0

Table 3: Effect of animal/property pair co-occurrence on probability of "Yes" answers (by the LM). Pairs are grouped into 14 buckets containing exponentially growing ranges (due to the distribution of the occurrence counts).

7 Conclusions

In this work we have shown the statistical relation of word occurrence and co-occurrence with the knowledge acquired by a language model during pre-training. Our results show two phenomena expressing the LM's knowledge which can be explained by the statistical distribution of words in the language. We believe that this work can serve as a baseline for further research on the topic of improving LM's limitations that occur due to pre-training, and are not fixed during fine tuning.

8 Discussion and Further Work

This work was executed using limited resources, and therefore can be perceived as a prototype of research on this topic. All testing data was created manually, including choice of animals and properties, writing questions and paraphrasing them, etc. Thus, all conclusions are based on a relatively small amount of data, which may not suffice for rigorous statistical proof.

To improve the accuracy of the results and the scope of the work, data creation can be crowd-sourced, e.g. using Amazon Mechanical Turk. Creating and utilizing a larger test set and additional test sets (on different topics) can help strengthen our conclusions, and at large advance research on limitations of LMs.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155.

Stevo Bozinovski. 2020. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3).

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, page 25–30, New York, NY, USA. Association for Computing Machinery.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020a. olmpics – on what language model pre-training captures.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020b. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.