Learning Word Embedding from Dictionary Definitions

Stanford CS224N Custom Project

Keertana Chidambaram
Department of Management Science & Engineering
Stanford University
vck@stanford.edu

Madhurima Mahajan
Department of Materials Science & Engineering
Stanford University
madhu12@stanford.edu

Handi Zhao
Institute for Computational & Mathematical Engineering
Stanford University
hdzhao@stanford.edu

Abstract

LLMs only learn words and meanings that arise in their training corpus. Our study is motivated by the fact that new words and definitions emerge over time. Retraining the entire language model to incorporate these changes would be too expensive. Specifically, we evaluate the ability of LLMs to acquire new words and their meanings by fine-tuning them with the word definition.

We test this strategy on GPT2 using two data sets: the LAMBADA dataset (Paperno et al., 2016) and the Urban Dictionary dataset. We are able to achieve an average perplexity of 308.9 compared to the best achievable perplexity of 79.37 for the LAMBADA dataset. For the Urban Dictionary dataset, we are able to achieve finite perplexity that is orders of magnitude smaller ($O(10^5)$ vs. $O(10^{25})$)than that for learning using definitions in context. Our experiments affirm that fine-tuning using a dictionary is a promising strategy to help LLMs learn new words.

1 Key Information to include

- Mentor: Steven Cao, Stanford NLP Group
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

New words and meanings emerge over time in any language. , But Large Language Models (LLMs) only learn words and meanings present in their training corpus. Hence the vocabulary and associated meanings of LLMs are frozen in time, and because of this, LLMs are susceptible to diachronic degradation. If LLMs don't keep up with the evolving language usage, it can affect their performance on a variety of downstream tasks like sentiment analysis, classification, machine translation, and question-answering (Huang and Paul, 2018; Lukes and Søgaard, 2018; Florio et al., 2020).

Completely retraining LLMs to help them acquire new words or information can be very slow and expensive. This motivates the need to find alternate strategies to update LLMs efficiently. Several studies have looked at various strategies to adapt LLMs to new data without complete retraining (Brown et al., 2020; Li et al., 2022; Dhingra et al., 2022; Lazaridou et al., 2021; Zaken et al., 2021). But most of this work addresses changes in factual knowledge or task distribution shift. Our work contributes to this burgeoning research area by addressing the need to learn new words. More specifically, Eisenschlos et al. (2022) looks at how LLMs can learn new words when presented with their definition in context. While this is a simple and effective strategy to help LLMs keep up with the latest language, it can become tedious to perform when more and more new words emerge and when their usage frequency explodes. In such situations, it makes more sense for the LLMs to permanently add new words and meanings to their vocabulary.

To address this gap, we propose and evaluate the simple strategy of fine-tuning the LLM with the new word and its dictionary definition. We implement this strategy for two datasets: the urban dictionary dataset and the LAMBADA dataset. The following are the main results of our paper:

- Our strategy is able to produce perplexities orders of magnitude smaller ($O(10^2)$ and $O(10^{25})$ for LAMBADA and Urban Dictionary respectively vs $O(10^{25})$) as compared to the strategy when word definitions are learned in-context.
- The average perplexity for LAMBADA dataset for our strategy is 308.9 compared to the best achievable perplexity of 79.37 even though unlike Urban Dictionary, LAMBADA has only one definition per word and no example sentences.
- Fine-tuning from word definitions is a promising strategy to update LLMs to learn new words.

3 Related Work

Updating LLMs

There have been recent work where LLMs were adapted to learn new words without training the model from scratch. Zaken et al. (2021) introduces a new fine-tuning method called BitFit that fine-tunes only a small portion of the model's parameters. The authors evaluate the performance of BitFit on several natural language processing (NLP) tasks, including text classification and named entity recognition. They compare the results with other fine-tuning methods and show that BitFit achieves comparable or better results while using significantly fewer parameters. In (Dhingra et al., 2022) authors propose time-aware language model that incorporates temporal information into language models. They introduce a simple technique for jointly modeling text with its timestamp. These models can accurately predict temporal patterns in language usage and extract temporal information from text, improving performance on tasks requiring temporal reasoning. Schick and Schütze (2020) adapted attentive mimicking to explicitly learn rare word embeddings to language models

Learning from dictionary definitions

Several studies have also utilized the definitions of new words to learn their word embeddings. The WINODICT method, proposed by Eisenschlos et al. (2022), addresses the Winograd task [1] by utilizing in-context learning with synthetic words. The method involves generating synthetic words to substitute the key concept tokens in the task, and then using the definitions of these key concept lemmas to facilitate in-context learning and solve the Winograd task. Through this approach, LLM can effectively learn words using the task's provided definitions. Yu et al. (2021)introduces Dict-BERT, a new pre-training method where word definitions of rare word are appended to pre-training corpus. The authors propose two novel self-supervised training tasks to help language model learn better representations for rare words. Wu et al. (2021) proposed to maintain a note dictionary and saves a rare word's contextual information as notes. When the same rare word occurs again during language model pre-training, the note information saved beforehand can be employed to enhance the semantics of the current sentence.

4 Approach

In the WINODICT paper Eisenschlos et al. (2022), the authors evaluate the effectiveness of in-context learning of new words by providing the word and its definition in the prompt. Our goal is similar, but

[1] https://wordnet.princeton.edu/

our strategy is different. We also want to evaluate the ability of LLMs to learn new words, but instead of learning in context, we fine-tune the LLM on the word's definitions and then evaluate how well the LLM has learned the new word.

We are using GPT2 (Radford et al., 2019) for our experiment. The version we are using is the GPT2LMHeadModel available through Huggingface [2]. The GPT2LMHeadModel inherits from the pre-trained GPT2 model, but the GPT2 Model transformer has a language modeling head on top. This makes this model suitable for language generation. We fine-tune the GPT2LMHeadModel using the new words and corresponding definitions and then evaluate the fine-tuned model.

An additional trick we have implemented is initializing the new word's token embedding following John Hewitt's blog [3]. We initialize the new word's tokens beforehand so that the words are not split into redundant tokens (for example, the word 'Frodo' will be tokenized as ['F', 'ro', 'do'] if it is not added as a token beforehand).

5 Experiments

5.1 Data

We use data from two sources: (1) Urban Dictionary dataset and (2) the LAMBADA dataset.

Urban Dictionary The raw Urban Dictionary dataset contains 809,660 crowdsourced data points. Because of limited compute, we stream about 20,000 of those data points. Further, we also enforce several filters and sanity checks to ensure the quality of these data points. The relevant features from the dataset along with their filters are shown in Table 1.

	Feature	Datatype	Description	Filters
1	word	string	The new word we are interested in learning	Word doesn't already exist in the vocabulary, word is a single word and not a phrase
2	definition	string	Definition of the word	String length is at least 0 and at most 1000
3	example_sentence	string	Example sentence usage with the word	String length is at least 0 and at most 1000, and the word appears at least once in the example_sentence
4	thumbs_up	integer	The number of votes the definition and example sentence received from other users	Value is at least 100 to ensure quality

Table 1: Urban Dictionary Dataset Features

Unlike a standard dictionary, the Urban Dictionary definitions are crowdsourced and hence also reflect biases and opinions of the author. A single word could have multiple conflicting definitions. For example, here are two conflicting definitions for the word 'republican':

(1) "Those who defend the constitution with honor and bravery. Those who think the punishment of criminals needs to be harsher and children need to be saved. Those who hate war but see that the only way through peace is through superior firepower."

(2) "Someone who is ready to kill innocent women and children because of something their leader did. An idiot whose brains would fit in an gnat's skull. A fool who is going to cause the world to hate us forever, and get the world to finally band together to kill us."

Further, some definitions may be 'bad', i.e., completely irrelevant, poorly written, or complicated (for example: using a roundabout metaphor). Therefore, to ensure that we capture rich and accurate word meanings, we only use the words with at least three available definitions, i.e., the word appears at least three times in the dataset. Finally, after all the filtering, we have 6,821 data points for 1,718

[2] `https://huggingface.co/docs/transformers/v4.27.1/en/model_doc/gpt2`

[3] `https://nlp.stanford.edu/~johnhew/vocab-expansion.html`

unique words. So every word has an average of about four definitions and four example sentences each.

LAMBADA The LAMBADA dataset (Paperno et al., 2016) has three main parts: the context, the target sentence, and the target word. The target word is the correct final word in the target sentence, and the target word's value is fully dependent on the context sentence. All three sentences are obtained from unpublished novels, and we are using a total of 2493 such sentence pairs. The final task is to predict the target word given the context and the target sentence. We modify this dataset so that the task can evaluate the ability of LLMs to learn from definitions. For this, we first generate a synthetic word using the script from the WINODICT dataset (Eisenschlos et al., 2022). We then randomly replace a common word from the context sentence with the synthetically generated word. Then we look up the definition of the replaced common word and link that definition to the generated synthetic word. Here's an example data point:

Original Context: "Yes, I thought I was going to lose the baby." "I was scared too," he stated, sincerity flooding his eyes. "You were ?" "Yes, of course. Why do you even ask?" "This baby wasn't exactly planned for."

Replaced Word: sincerity

Synthetic Word: gelomrity

Synthetic Word Meaning: the quality of being free from pretense, deceit, or hypocrisy

Updated Context: "Yes, I thought I was going to lose the baby." "I was scared too," he stated, gelomrity flooding his eyes. "You were ?" "Yes, of course. Why do you even ask?" "This baby wasn't exactly planned for."

Target Sentence: "Do you honestly think that I would want you to have a ______?"

Target Word: miscarriage

The following table summarizes the test/train/validation splits for our data:

Train	**Validation**	**Test**	**Total**
1,343 (53.9%)	150 (6%)	1000 (40.1%)	2,493

Table 2: LAMBADA dataset split

	Train	**Validation**	**Test**	**Total**
Word definitions	6139	682		6821
Example sentences	1228	136	5457	6821

Table 3: Urban Dictionary dataset split

5.2 Evaluation Method

Since we are using a language generation model, the main metric we are using to compare various methods is the perplexity (PPL). A smaller perplexity indicates that the sentence in question has higher probability of being generated. If the new word is in the sentence, this indicates that the model has learned the new word's meaning well enough to understand that its usage in the test sentences is justified. We define several tasks to evaluate our method based on the perplexity scores:

Urban Dictionary Dataset

Task 1A: For this task, we give the *test dataset* as the input to the *untrained model* and compute the perplexity.

Task 1B: For this task, we give the *test dataset and the word definition (for context)* as the input to the *untrained model* and compute the perplexity.

Task 1C: For this task, we give the *test dataset* as the input to the *model fine-tuned using word definitions and some example sentences* and compute the perplexity.

Tasks 1A and 1B act as benchmarks. 1A tests if pre-training makes any difference at all in learning new words and 1B emulates the in-context learning sceme of the Eisenschlos et al. (2022) paper. Task 1C implements our startegy. Comparing the perplexity values from the three tasks will allow us to evaluate our strategy.

LAMBADA Dataset

Task 2A: For this task, we give the *original context*, the target sentence, and the target word (filled into the target sentence) as the input to the *untrained model* and compute the perplexity.

Task 2B: For this task, we give the *updated context*, the target sentence, and the target word (filled into the target sentence) as the input to the *untrained model* and compute the perplexity.

Task 2C: For this task, we give the *updated context, word definition*, the target sentence, and the target word (filled into the target sentence) as the input to the *untrained model* and compute the perplexity.

Task 2D: For this task, we give the *updated context*, the target sentence, and the target word (filled into the target sentence) as the input to the *model trained on the synthetic word and its definition* and compute the perplexity.

Here, tasks 2A, 2B, and 2C act as benchmarks. Task 2A has nothing to do with the synthetic word but just evaluates the ability of GPT2 to even solve the basic LAMBADA task with no synthetic word. Obviously, this will be the best performance that can be expected from our strategy because all the words in the input already have rich representations in the untrained model. Task 2B and 2C serve the same purpose as 1A and 1B respectively. Finally, task 2D evaluates our strategy. As before, comparing the four tasks will further help evaluate our method.

5.3 Experimental Details

Parameter	Value
Epochs	5
Learning rate	5e-4
Warmup steps	1e2
Epsilon	1e-8
Sample every	100
Batch size	2

Table 4: Experimental details

Same parameters were used to train on Urban Dictionary dataset and LAMBADA dataset. It took 1.5 hours and 40 minutes to train on Urban Dictionary dataset and LAMBADA dataset, respectively.

5.4 Results

The average perplexity values for each of the tasks is presented in Table [5].

The perplexity values for tasks 1A and 2B were either infinity or some really high value (at least 10^{25}). This is expected because the model has encountered words it hasn't seen before and the word's token doesn't have a representative embedding. What was more surprising was that tasks 1B and 2C

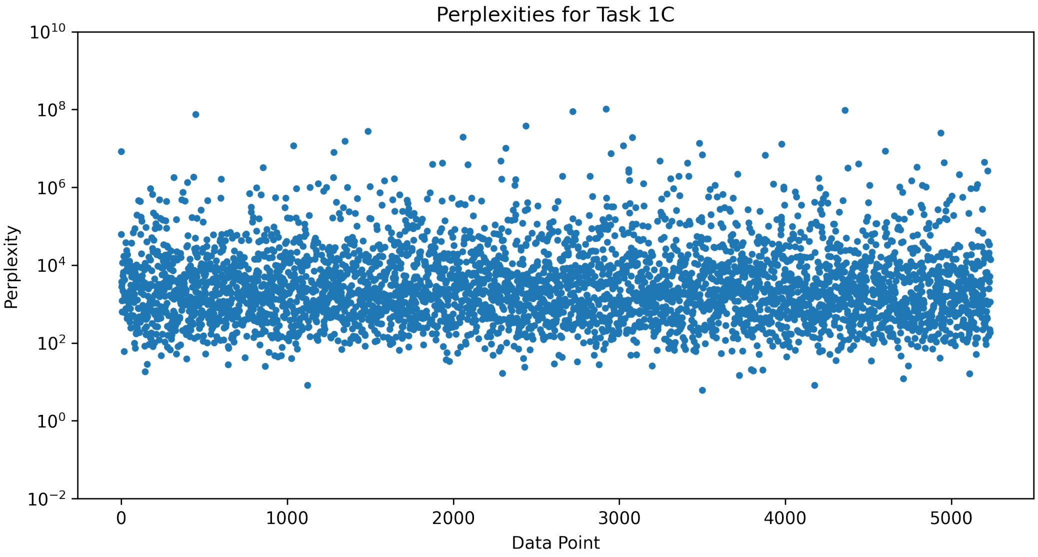

Figure 1: Task 1C results

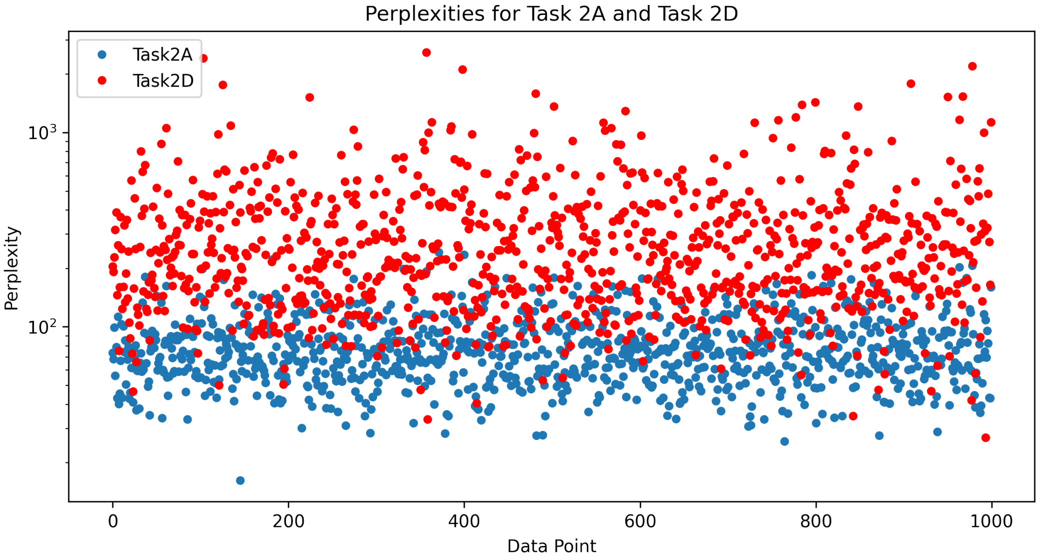

Figure 2: Task 2A and Task 2D results

mostly reported infinity or or some really high value (at least 10^{25}) of perplexity - the values were very similar to that from tasks 1A and 2B respectively. This goes on to show that in-context learning of word definitions doesn't work for GPT-2. Tasks 1C and 2D confirm that our method indeed works to some degree. The high (but lower than that of task 1A/1B) perplexity value for task 1C can be attributed to the fact that the urban dictionary dataset is crowdsourced - hence the quality of data is poor. Upon manual inspection we found lots of grammatical errors, spelling mistakes, heavy usage of

Urban Dictionary		**LAMBADA**	
Task	Avg Perplexity	Task	Avg Perplexity
Task 1A	∞	Task 2A	79.37
Task 1B	∞	Task 2B	∞
Task 1C	193492	Task 2C	∞
		Task 2D	308.91

Table 5: Perplexity results

slang words which the model may not have a representation for, etc. in the urban dictionary dataset. As expected task 2A provides somewhat of an upper bound for our tasks - for task 2D, we obtain a larger but comparable perplexity compared to task 2A

6 Analysis

Remarks on comparison Although we were able to show that our method outperforms in-context learning of definitions, the WINODICT paper(Eisenschlos et al., 2022) was performed on GPT-3 and PALM. Perhaps those models are better at in-context learning than GPT-2. So our results may not generalize to other language models and testing our strategy out on other LLMs would make for an interesting extension.

Remarks on datasets In hindsight, the Urban Dictionary and the LAMBADA dataset complement each other well. Comparing both data, on one hand, the Urban Dictionary dataset has multiple definitions and example sentences, but the definition quality is lacking (e.g., poor grammar, not concise, not spell-checked, etc.) because it is crowdsourced. On the other hand, in the LAMBADA dataset, each new word only has one definition, but the data for both the definition and the task sentences are high quality. Comparing the results in the Urban Dictionary dataset, the perplexity values for our method is very high; because of that, we weren't able to conclusively say whether or not learning from the definition works. But the LAMBADA dataset solves this issue. Since we have a measure of perplexity upper bound for the LAMBADA dataset, we can see that our model perplexity is comparable to the best achievable perplexity, which validates our strategy.

7 Conclusion

Our main findings is that fine-tuning using dictionary definitions is indeed a promising strategy to update LLMs to learn new words. For GPT-2, our method also outperforms learning through in-context prompting using the definition. We are able to decently learn word meanings even with small data per word as shown by the LAMBADA experiment results.

The Urban Dictionary dataset contains offensive, racist, sexist, and uncensored language. An extension to our study would be to rerun our experiments with a fairer and cleaner dataset; it may produce better results and is more ethical. We are mapping a synthetic word to an existing word's definition for the LAMBADA dataset. If the language model can figure out this mapping, our method may not scale well to words without a good mapping in the current vocabulary. An interesting extension is to run experiments with foreign words that do not have an equivalent word in the English language (this strategy was implemented in the Eisenschlos et al. (2022) paper) to see if our strategy is still able to do well.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Julian Martin Eisenschlos, Jeremy R Cole, Fangyu Liu, and William W Cohen. 2022. Winodict: Probing language models for in-context word acquisition. *arXiv preprint arXiv:2209.12153*.

Komal Florio, Valerio Basile, Marco Polignano, Pierpaolo Basile, and Viviana Patti. 2020. Time of your hate: The challenge of time in hate speech detection on social media. *Applied Sciences*, 10(12):4180.

Xiaolei Huang and Michael Paul. 2018. Examining temporality in document classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 694–699.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11838–11855.

Jan Lukes and Anders Søgaard. 2018. Sentiment analysis under temporal shift. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 65–71.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774.

Qiyu Wu, Chen Xing, Yatao Li, Guolin Ke, Di He, and Tie-Yan Liu. 2021. Taking notes on the fly helps language pre-training. In *International Conference on Learning Representations*.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2021. Dict-bert: Enhancing language model pre-training with dictionary. *arXiv preprint arXiv:2110.06490*.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.