Today Years Old:
Adapting Language Models to Word Shifts

Stanford CS224N Custom Project

Name: Olivia Lee
SUNet ID: oliviayl
Department of Computer Science
Stanford University
oliviayl@stanford.edu

Name: Jason Chen
SUNet ID: jasonjin
Department of Computer Science
Stanford University
jasonjin@stanford.edu

Name: Zachary Xi
SUNet ID: zlxi
Department of Computer Science
Stanford University
zlxi@stanford.edu

Abstract

Large language models (LMs) are typically pretrained on large, fixed-vocabulary text corpora. However, pretrained LMs currently face challenges adapting to novel or modified lexical items. This project is motivated by the question: how can unseen lexical items be incorporated into pretrained LMs? Leveraging new lexical items defined in Urban Dictionary, we propose two approaches to editing LMs for lexical adaptation: (1) initializing new word embeddings by averaging existing embeddings instead of small-norm random noise, and (2) using supervised learning to predict embeddings of new lexical items given their definition. We also design a multiple choice, fill-in-the-blank evaluation method to assess changes in LM adaptive ability. (findings in progress) The averaging method for embedding initialization enables better adaptation and circumvents issues with finetuned LMs only generating the new words. Our fill-in-the-blank evaluation task also demonstrates learning mappings between definitions and word embeddings enables LMs to successfully learn the definitions of new lexical items.

Our code can be found at this repository.

1 Key Information to include

- Mentors: Jesse Mu (TA), Steven Cao (External)
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

Changes in word meanings are inherent in language, especially with a global communication system popularizing terms with new meanings or facilitating the creation of new terms altogether. A majority of words have multiple meanings; certain meanings may become more or less prevalent, and a new meaning for a word can be added to the existing list or even replace a former meaning [1]. For instance, consider the term 'lit', which has gained a new sense of 'exciting' or 'awesome', derived from its established use as slang for 'intoxicated' to describe the vibrant environment in which acts of becoming intoxicated often occur [2]. Among younger populations, these updated meanings are often more prevalent than its original meanings of 'illuminated' or the past participle of 'light'.

Accounting for lexical semantic shifts (whether for changing word meanings, adapting to specific downstream tasks, or fine-tuning on new domains) would consequently be important for any system attempting to model human language. There have been rapid advancements in the development of language models (LMs) in recent years that have performed well on a wide variety of tasks, in part due pretraining on large, fixed-vocabulary text corpuses. However, given the ever-changing nature of society and human language, it is thus natural to pose the question: how well do pretrained LMs adapt to lexical semantic shifts? It appears that current pretrained LMs face challenges incorporating novel or modified lexical items. Empirically, it has been observed that attempts to add new words to the vocabulary of pretrained LMs result in the updated model only generating the new words. This is because the logits of existing words often become negative and large after pretraining, whereas the default behavior of e.g. HuggingFace transformer-based models is to initialize the embeddings of new words with the same distribution used before pretraining, i.e., small-norm random noise. [3]

The goal of this project is to develop an approach to editing LMs for lexical adaptation, motivated by the question: how can unseen lexical items be incorporated into pretrained LMs without having to constantly retrain the model? We use Urban Dictionary data to investigate this question. Urban Dictionary is a crowdsourced English-language online dictionary for slang words and phrases, often including both definitions and example sentences for new lexical items, both of which will be useful training data. Using off-the-shelf pretrained LMs GPT-2 [4] and RoBERTa [5], we aim to compare our proposed approaches to these baselines. This project also designs and implements a multiple choice, fill-in-the-blank evaluation method to assess changes in adaptive ability, inspired by fill-in-the-blank evaluation task developed in WINODICT [6].

3 Related Work

Evaluating LLMs' ability to learn novel words at inference. Previous works have explored creation of benchmarks for new or adjusted lexical items to support the continued semantic evolution of language. More specifically, [6] created a new benchmark named WINODICT, which is a dataset of co-reference resolution tasks that builds upon previous WINOGRAD[7] and WINOGRANDE[8] The work of WINODICT centers around the idea of introducing new knowledge through prompting, assisting the model learn in learning new concepts by defining them in terms of previously existing concepts. [6] then evaluates this dataset on existing LLMs, finding that smaller versions of GPT-3 and PaLM with fewer parameters, such as PaLM-8B yield performance that resembles guessing. Our approach in leveraging the Urban Dictionary dataset pursues the same goal of improving model recognition of new or adjusted lexical items, adapting models to shifting semantic meaning. The primary distinction between the WINODDICT and the Urban Dictionary dataset is that Urban Dictionary terms adjust the semantic definitions and corresponding usage of existing lexical terms, whereas the WINODICT dataset tests models for new semantic learning of synthetic lexical terms.

Detecting semantic shifts. Prior works have investigated evaluation frameworks of semantic change detections over time, specifically proposing a novel evaluation framework for semantic changes of lexical items [9]. This framework collected Tweet data between January 2012 and January 2017 and computed distinct word embeddings for the dataset between monthly time bins. They then measured the cosine distance between word embeddings over time to determine the degree of semantic change. The study then constructs a dataset consisting of a random sample of words with the greatest computed semantic change alongside "pseudowords," words that have their word-embeddings artificially adjusted to simulate unnatural semantic shifts. The evaluation method then tests if LLMs can distinguish between naturally shifted words and artificially adjusted "pseudowords." The paper found that independently trained and aligned embeddings performed better in semantic change

detection than embeddings that were continuously changed for long periods. Our work pursues a similar goal to [9] of detecting semantic shifts across time periods, but focuses on utilizing lexical items with adjusted semantic definitions rather than detecting the semantic change. In our approach, we aim to tailor our model to words with new or adjusted semantic definitions from the Urban Dictionary dataset and identify semantic changes across these examples fine-tune model learning and adaptation with semantic shifts.

Learning word embeddings via reconstruction. Prior works have investigated learning word embeddings via reconstruction, since dictionary definitions use words that are themselves dictionary entries, to facilitate parameter sharing [10]. Such approaches leverage the inherent recursivity of dictionaries by encouraging similarity between input definitions and encoded definition embeddings via a consistency penalty, as both embeddings lie in the same vector space. Specifically, [10] uses an LSTM to encode a dictionary definition into an embedding, and a conditional language model trained by maximum likelihood to decode or regenerate the original definition given the definition embedding. The resulting embeddings capture semantic similarity (as opposed to relatedness) better than regular distributional methods, and such methods can generalize one-shot when trained exclusively on dictionary data. Our approach is also motivated by leveraging dictionary definitions that use other seen-before words to learn word embeddings for novel lexical items. A key distinction is that embeddings of words used in Urban Dictionary definitions may already exist and need not be learned. This removes the need for an autoencoder model, and our approach involves simply retrieving learned word embeddings from the pretrained LM.

4 Approach

To edit LMs for lexical adaptation, we train a separate neural network to predict the embedding of the new word from Urban Dictionary given its definition by finetuning one of the two pretrained models: GPT-2 (unidirectional) or RoBERTa (bidirectional) [5].

The pretrained LM, GPT-2 or RoBERTa, is finetuned via supervised learning. We finetune the model on a dictionary of common words (words which the pretrained LM has likely encountered during training, which is separate from the Urban Dictionary dataset; see Section 5.1) to learn a mapping from dictionary definitions to word embeddings. Concretely, for a common word w, the model outputs a predicted embedding given the definition(s) of w, which are concatenated and collectively tokenized. As the ground-truth for supervised learning, we extract the embedding corresponding to word w from the embedding matrix of the pretrained LM. If w is tokenized into multiple tokens, we extract the embeddings corresponding to each of the tokens and use the average as the ground-truth. Since w has likely been already learned during the large-scale pretraining process of the LM, this ground-truth is likely to be accurate.

We then perform gradient descent on the L2 loss between the [CLS] embedding at the model's last hidden layer and the ground-truth embedding from pretrained LM of the word w. Once the pretrained LM is finetuned to learn mappings from English definitions to word embeddings, we predict new word embeddings for Urban Dictionary words given the definitions in the Urban Dictionary dataset. See Figure 1 for architectural details.

Our approach involve adding new word embeddings to the existing embedding matrix. Therefore, to evaluate our approach, we compare it against a baseline adaptation method proposed in [3] that initializes new word embeddings using an average of all existing embeddings. Initializing new word embeddings for domain-specific tasks with small-norm random noise often leads to LMs only generating the tokens which correspond to those new word embeddings, because the probability of pre-expansion words can only decrease as the new partition function of the post-expansion LM can only be larger than the old one. Hence, the post-expansion probability of an existing embedding is its old probability multiplied by a factor smaller than one. Therefore, to initialize new embeddings for Urban Dictionary lexical items, we average over all pre-expansion embeddings and use this as our standard normal vector to construct a multivariate normal distribution which we sample from for each novel word added to the tokenizer's vocabulary. This bounds the KL divergence between pre- and post-expansion distributions since the assigned probability of the new word cannot be higher than $\frac{1}{n}$, so as the vocabulary size of the LM grows, the new word's probability decreases. The sampled embeddings are appended to the embedding matrix of our respective model. This model expansion process is implemented on both GPT-2 (unidirectional) and RoBERTa (bidirectional).

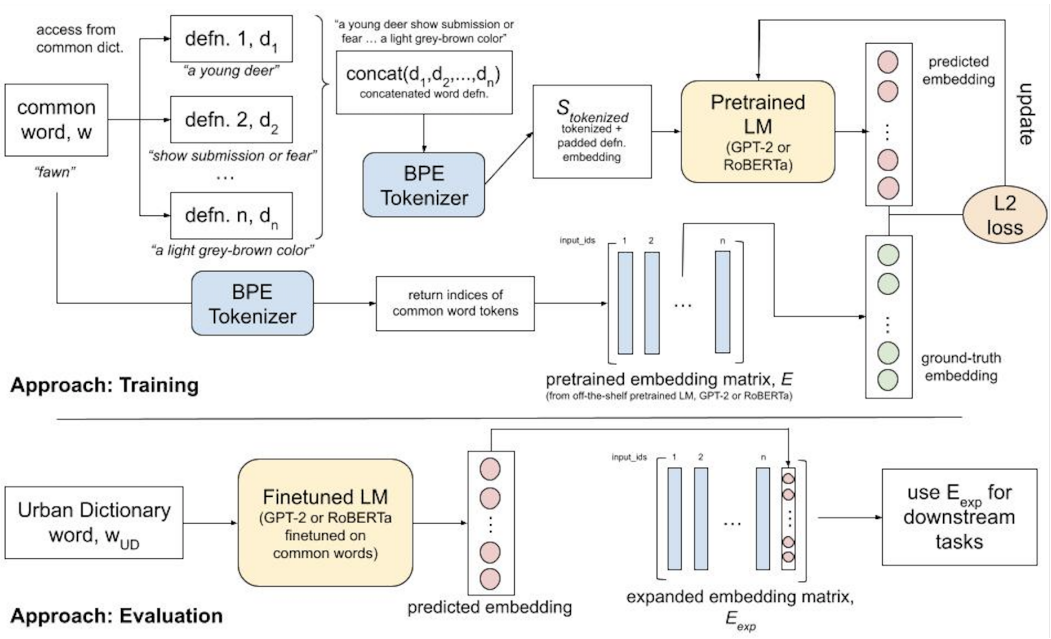

Figure 1: Training and Evaluation procedure for our approach

Both our approach and the baseline adaptation method involve adding new word embeddings to the existing embedding matrix, however, it is a open question whether unseen lexical items from Urban Dictionary require entirely new word embeddings, or if they can be approximated using subword embeddings generated via Byte-Pair Encoding (BPE) tokenizers used by GPT-2 and RoBERTa. As such, we will also be comparing our approach to the GPT-2 and RoBERTa pretrained LMs used off-the-shelf with no adaptation, which will not involve adding any new word embeddings.

5 Experiments

5.1 Data

We made use of the two main datasets. First, the Urban Dictionary dataset, a preprocessed dataset provided to us by our project mentor. The dataset contains 1,591,600 unique words, 2,606,521 total definitions, and example sentences from Urban Dictionary. Second, a dictionary of common word items, which is a preprocessed dataset provided to us by our mentor. It contains 147,306 common words and their definitions, likely incorporated in the large-scale dataset used for pretraining the LMs.

5.2 Evaluation method

To evaluate our approach, masked language modeling involves masking over a specific word within an input sentence and identifying the best candidate word from the model's vocabulary that should replace that masked word. As RoBERTa is a bidirectional model, we performed masked language modeling with RoBERTa, going over every entry in our dataset and masking the respective word within the provided example sentence and feeding the masked example sentence as input to the model. Then we calculate the ranking of the respective word within the distribution of all possible logits in the embedding space of RoBERTa, which is fetched by retrieving the index of the masked token from the outputs. We calculate the average ranking of the respective words over the entire dataset along with the total number of times that an urban dictionary word appears within the top 10, 100, 1000, and 5000 most likely embeddings for its respective evaluation.

To evaluate the GPT-2 model, we could not perform masked language modeling and used a different approach that leveraged causal language modeling because GPT-2 is a unidirectional model. Thus, we had to condition on the initial part of the example sentence existing before the target word. To

account for this, we instead sampled 50 embeddings from the embedding space of GPT-2 and masked their respect tokens in place of the target word. We evaluated the perplexity of these 50 sample sentences using their log likelihood and calculated the relative ranking of the target word within this sampled distribution, reporting the top 5, 10, and 25 most likely embeddings for evaluation. Because of the approach of sampling sentences was more computationally intensive than the masked language modeling approach, we imposed a higher minimum upvote threshold (see Section 5.3) to reduce the number of lexical items used for evaluation.

5.3 Experimental details

To finetune on common words to learn mappings from word definitions to embeddings, training starts one of the two following pretrained models: the unidirectional GPT-2 Model transformer with a language modeling head on top (linear layer with weights tied to the input embeddings) or the bidirectional RoBERTa Model with a language modeling head on top. The initial set of word token embeddings is loaded from the corresponding model. For training on the dictionary of common words, we use the L2 loss function, a batch size of 8, a learning rate of 3e-5, optimized using the Adam optimizer. We trained the model for 3 epochs, as 3 training epochs was what was feasible given the timeframe and the recommended number of epochs for finetuning LMs is 2-4 epochs [11].

We run the following three experiments, comparing our approach to the respective baseline:

Experiment 1. We expanded the embedding matrix to incorporate 961 unique single-word lexical items for GPT-2 and 5,383 unique single-word lexical items for RoBERTa. This primary experiment incorporated Urban Dictionary lexical items consisting of a single word and tokenized as more than one token (if a word was tokenized as one token, it was likely a common word disguised as a novel word, and we did not add that word to the model). Our baseline was expanding the embedding matrix with new embeddings for each new token, initialized using the averaging method.

Experiment 2. In this experiment, we incorporate terms from Urban Dictionary that have already been learned by the base pretrained LM (i.e., tokens that already exist in the pretrained embedding matrix). This is to demonstrate the benefit or learning the novel definitions of common words (e.g., "lit") that exist in regular dictionaries. For GPT-2 we evaluated on 546 single-word lexical items, and for RoBERTa we evaluated on 3,363 single-word lexical items. Since this experiment involves evaluation on tokens that the pretrained LMs already have in their vocabulary, our baseline was the off-the-shelf GPT-2 and RoBERTa models, with no new tokens added.

Experiment 3. In this experiment, on top of novel single-word lexical items, we also add multi-word lexical items from Urban Dictionary to demonstrate the benefit of learning the definitions of novel phrases (e.g., "today years old"). For GPT-2, we incorporated 976 multi-word lexical items, and for RoBERTa, we incorporated 9007 multi-word lexical items. Our baseline was expanding the embedding matrix with new embeddings, initialized using the averaging method.

For all experiments, novel lexical items were filtered such that all items added had more upvotes than downvotes, at least 5,000 upvotes for GPT-2, and at least 1,000 upvotes for RoBERTa for high-quality definitions (see Section 5.2 for why differing number of unique lexical items were added).

5.4 Results

Below are the results for Experiment 1 (novel, single-word lexical items):

	GPT-2			RoBERTa	
	Baseline	**Ours**		Baseline	**Ours**
Top 5	123	**175**	Top 10	20	**46**
Top 10	287	**371**	Top 100	241	**410**
Top 25	**758**	734	Top 1000	2451	**4660**
Avg. Rank	25.53185	**24.7931**	Avg. Rank	**2633.67000**	2679.18237

Table 1: Experiment 1

Based on these results, our approach performs comparably or outperforms the baseline on all metrics. Notably, for RoBERTa, our approach results in almost 100% increase in Urban Dictionary word

appearances in the top k for $k = 10, 100, 1000$. Assessing the extent of improvement from the baseline between the two models, RoBERTa generally did better than GPT-2, which suggests that a bidirectional model is capable of more effectively learning word embeddings for novel lexical items as it is able to utilize the context on both sides on the masked token, and not just the context preceding the masked token, for prediction.

Below are the results for Experiment 2 (common single-word lexical items):

	GPT-2			RoBERTa	
	Baseline	**Ours**		Baseline	**Ours**
Top 5	55	**78**	Top 10	31	**710**
Top 10	130	**147**	Top 100	219	**1499**
Top 25	**281**	267	Top 1000	1169	**3285**
Avg. Rank	**20.18329**	20.32947	Avg. Rank	15191.21566	**12724.65158**

Table 2: Experiment 2

Based on the results, our approach performs comparably or outperforms the baseline on all metrics. Notably, for RoBERTa, our approach results in approximately a 22x increase in word appearances in the top 10, 7x increase in top 100, and 3x increase in the top 1000. Assessing the extent of improvement from the baseline between the two models, the bidirectional RoBERTa model seems capable of more effectively learning word embeddings for novel lexical items, for similar reasons to Experiment 1.

Below are the results for Experiment 3 (novel single- and multi-word lexical items):

	GPT-2			RoBERTa	
	Baseline	**Ours**		Baseline	**Ours**
Top 5	**26**	19	Top 10	**25**	11
Top 10	103	**188**	Top 100	**223**	201
Top 25	548	**595**	Top 1000	2018	**2175**
Avg. Rank	**23.5482932182**	25.5260416667	Avg. Rank	4493.93757395	**4435.44025015**

Table 3: Experiment 3

Based on the results, our approach performs comparably or outperforms the baseline on most metrics. The improvement from the bidirectional RoBERTa model over the unidirectional GPT-2 model is not as significant here compared to previous experiments. The advantages of our appraoch may be less apparent due to the more complex, compositional meanings that multi-word phrases, especially novel ones, have. Overall, training language models to derive abstract, compositional meanings of phrases and idioms is an active area of research [12], thus, further work can investigate more complex approaches to handling multi-word, novel lexical items.

Collectively, the results suggest that the trained model from our approach has successfully learned mappings from word definitions to word embeddings, for both novel lexical items as well as previously learned words with novel meanings. That is, the model can predict word embeddings for unseen single-word lexical items, and predicted embeddings can be used for downstream language inference tasks. The results suggest that editing pretrained LMs to incorporate novel lexical items improve the inference abilities of such LMs on sentences containing novel lexical items, and demonstrates improved performance compared to using the existing token embeddings in off-the-shelf models with no adaptation (as in Experiment 2).

All code for model training and evaluation can be found at this repository.

6 Analysis

For a qualitative analysis of our model, we examined the embedding space of Urban Dictionary entries generated through our neural dependency pairing method. Although the newly created Urban

Dictionary word embeddings did not exhibit strong correlations with relevant words to the same degree as pre-trained RoBERTa embeddings, we observed that neural dependency parsing managed to establish connections between novel terms. We scrutinized specific novel words added to our tokenizer vocabulary and inspected those with high cosine similarity. We discovered that words signifying geographic origins or individuals from respective regions, such as *British*, *Manhattan*, *California*, and *Peruvian*, tended to cluster. Additionally, terms related to religiosity, like *scientology*, *creationist*, and *sacrelicious* (a word denoting holiness with positive or negative connotations), were found in proximity. Famous individuals, such as pop stars *Beyonce*, *Adele*, and *Ariana*, also exhibited high cosine similarity. Interestingly, terms associated with social liberalism, including *vegetarian*, *pro-choice*, *California*, *Canadian*, *ACLU*, and *homosexual*, formed closely clustered representations. Derogatory expressions like *dolt*, *stupid*, *lame*, and *jag* (an annoying person lacking a social filter) had smaller distances between them. Similarly, expressions of amusement, such as *Lmao*, *lol*, *lolol*, and the smiley face emoticon "*:)*", were found to be closely related within the embedding space.

Overall, this suggests that our model was generally able to learn sensible similarity relations between novel terms. The dissimilarity between novel words and relevant previously learned words could have arisen from the nature of our approach that adds new tokens. Future work could look into modifying existing tokens, or a hybrid approach of modifying existing tokens and adding new ones, to create a more unified structure within the embedding space.

7 Conclusion

We present a novel supervised learning method for editing pretrained LMs to incorporate novel lexical items from Urban Dictionary. Collectively, the results suggest that the trained model from our approach has successfully learned mappings from word definitions to word embeddings, and is able to predict word embeddings for unseen lexical items that can be used for downstream language inference tasks. This approach addresses a pitfall that pretrained LMs have in adapting to lexical shifts in language, as our results show that using the existing token embeddings in off-the-shelf models with no adaptation causes performance to suffer. Thus, editing pretrained LMs to incorporate novel lexical items improves inference on sentences containing novel lexical items.

An area that warrants additional research is handling words in Urban Dictionary, e.g. "dog", that are not actually novel words. In Urban Dictionary, these terms may have the same definition as the corresponding lexical item in a regular dictionary, or may have a novel slang definition, but either way our method adds the word as a new token to the model, which may create redundancy in our embedding matrix. Our results in Experiment 2 show that learning these novel meanings is beneficial, so future work can look into instead concatenating the word's Urban Dictionary definition to its regular dictionary definition and tokenizing the concatenated definitions, using the model in our approach to predict the new embedding, and replacing the corresponding embedding in the embedding matrix with the predicted embedding.

Similarly, our model currently adds multi-word novel lexical items as new tokens to the model's tokenizer. It may not be most sensible to add a token consisting of multiple words, e.g. "day friend". Phrases often consist of multiple common words, but when put together have a novel compositional meaning. Future work can investigate instead updating the multiple individual word embeddings in the tokenized phrase such that they implicitly represent the novel Urban Dictionary definition. For instance, a new approach could tokenize the phrase as usual, then finetune the representations at an intermediate layer once it has processed the meaning of the phrase to some degree, rather than at the embedding layer where it's still separate and non-contextual. Another approach would involve performing masked language modeling on the synthetic sentence "[word]: [definition]." Such approaches could also be applied to model editing to incorporate idioms and metaphorical meanings, or more broadly investigate how pretrained LMs combine subparts (words or morphemes) to derive more abstract or complex meanings.

References

[1] Peter Koch. Meaning change and semantic shifts. In Päivi Juvonen and Maria Koptjevskaja-Tamm, editors, *The Lexical Typology of Semantic Shifts*, page 21–66. De Gruyter Mouton, Berlin/Boston, 2016.

[2] Merriam Webster. Accessed 11 February 2023.

[3] John Hewitt. Initializing new word embeddings for pretrained language models, 2021.

[4] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[6] Julian Martin Eisenschlos, Jeremy R. Cole, Fangyu Liu, and William W. Cohen. Winodict: Probing language models for in-context word acquisition, 2022.

[7] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press, 2012.

[8] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, Apr. 2020.

[9] Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott Hale, and Barbara McGillivray. Room to Glo: A systematic comparison of semantic change detection approaches with word embeddings. 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 66–76, Hong Kong, China, November 2019. Association for Computational Linguistics.

[10] Tom Bosc and Pascal Vincent. Auto-encoding dictionary definitions into consistent word embeddings. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[12] Murathan Kurfali and Robert Östling. Disambiguation of potentially idiomatic expressions with contextual embeddings. In *Workshop on Multiword Expressions*, 2020.