Rethinking Our Assumptions about Language Model Evaluation

by Nancy Fulda

Brigham Young University, Provo, UT 84602, USA,
nfulda@cs.byu.edu

Abstract. Many applications of pre-trained language models use their learned internal representations, also known as word- or sentence embeddings, as input features for other language-based tasks. Over recent years, this has led to the implicit assumption that the quality of such embeddings is determined solely by their ability to facilitate transfer learning. In this position paper we argue that pre-trained linguistic embeddings have value above and beyond their utility as input features for downstream tasks. We adopt a paradigm in which they are instead treated as implicit knowledge repositories that can be used to solve common-sense reasoning problems via linear operations on embedded text. To validate this paradigm, we apply our methodology to tasks such as threat detection, emotional classification, and sentiment analysis, and demonstrate that linguistic embeddings show strong potential at solving such tasks directly, without the need for additional training. Motivated by these results, we advocate for empirical evaluations of language models that include vector-based reasoning tasks in addition to more traditional benchmarks, with the ultimate goal of facilitating language-based reasoning, or 'reasoning in the linguistic domain'. We conclude by analyzing the structure of currently available embedding models and identifying several shortcomings which must be overcome in order to realize the full potential of this approach.

Keywords: language models, language model evaluation, word embeddings, sentence embeddings, common-sense reasoning

1 Introduction

When evaluating language models and particularly their learned sentence representations, researchers often focus on cross-task generalization. The objective is to determine how well the learned sentence embeddings function as input features for other language-based tasks. The quality of the embedding space is thus, by default, defined in terms its facilitation of transfer learning.

This is a valid paradigm, but not the only possible one, and this paper encourages a community wide re-examination of our assumptions about language model evaluation. We begin by taking note of the way these models are being used in the wild – by hobbyists and industry professionals. When one reads blog posts and online articles about word embeddings, or when one browses through

discussion forums, the most common application of these learned representations is not as uninterpretable input features for downstream tasks. Instead, one observes an inherent fascination with the embeddings themselves. For example, the AI infrastructure website Skymind features an article that explores meaningful word2vec associations such as *Iraq - violence = Jordan* and *library - books = hall* [25]; computer science blogger Adrian Colyer writers about "the amazing power of word vectors" at representing semantic meaning [6]; and Chris Moody at StitchFix explores the idea of using vector addition to augment semantic search [24]. A common theme in these and other online articles is the idea that cosine distance between embedded texts can be used as an analogue for semantic similarity [14] [30] [31]. Many web sites also allow users to 'play' with various embedding spaces by calculating cosine similarity or projecting embeddings into interesting subspaces for observation [12] [21] [1] [2] [28].

These online artifacts leave us facing a strange dichotomy. Multi-word embedding spaces like skip-thoughts, BERT, and Google's universal sentence encoder gained prestige by exceeding previous transfer learning benchmarks [19] [7] [5], and yet the average user seems to want to use the linguistic embeddings *directly*, independent of any transfer learning they facilitate. Undeniably, there is a certain intuitive appeal to this desire. After all, if words and sentences can be represented as numbers, ought one not to be able to manipulate them mathematically?

1.1 Doing Math with Language

The answer, of course, is that one *can*, at least at the level of single words.

In 2013, Mikolov et al. observed what has since become a hallmark feature of word-level embedding models: their ability to represent linguistic regularities in the form of analogical relations [23]. Although trained for different purposes entirely, most word-level embedding models can be used to solve analogical queries of the form a:b::c:d (a is to b as c is to d). Leveraging this principle, it is possible to use simple linear relations to discover the unknown value d, e.g.

$$\text{Madrid - Spain + France = Paris}$$

The possibilities are tantalizing. Researchers have demonstrated that mathematical operations on word embeddings can be used to detect affordances [9], infer the locations of everyday objects [11], and condition agent behaviors on natural language instructions [8].

A natural extension of this phenomenon would be to apply these same principles at the sentence level. However, there is a notable dearth of papers demonstrating such applications, perhaps because coherent results are more difficult to achieve. For example, using a pre-trained skip-thought encoder [18] and corresponding decoder trained on reddit data, the following equivalences hold:

'I am angry' - 'angry' + 'happy' = 'I happy'

'thank you' + 'you're welcome' = 'enjoy you'

At first glance, it appears that this sentence-level embedding space is functioning with the same precision as its word-level predecessors. Most people find it intuitively obvious that if you remove anger and add happiness the sentence *'I am angry'* would transmute into *'I am happy'*, and the skip-thought embedding space has produced an acceptible approximation to that result. Similarly, the pleasantries *'thank you'* and *'you're welcome'* are generally used when we wish to create an atmosphere of congeniality and enjoyment of one another's company. The decoded phrase *'enjoy you'* is suggestive of this idea.

So far so good, but alas, the illusion of analogical coherence breaks down as soon as more equations are attempted:

'the sky is blue' - 'blue' + 'orange' = 'the orange is the orange'

We would have expected the output phrase to be *'the sky is orange'* or perhaps *'the sunset is orange'*, but instead we end up with a nonsense statement. Taking these examples together, it seems clear that the *potential* for direct mathematical manipulation of the embedding space is present, but the space is insufficiently structured (or the decoder insufficiently trained) to allow this information to be reliably extracted.

The goal of this paper is twofold: First, to demonstrate that this same potential is present in many of the currently available sentence embedding models, albeit in primordial form. Second, to outline steps that may lead to the full realization of this potential in future models.

We will proceed as follows: In Section 2 we discuss related work as pertaining to linguistic embedding models. Section 3 introduces a series of quantitative experiments that measure, to some extent, the amount of semantic knowledge encoded within a given embedding space, and presents our experimental results. Section 4 interprets and lends context to these results, issues a call for researchers to re-evaluate their default assumptions about language model evaluation, and lays out a roadmap for future work on this topic.

2 Related Work: Linguistic Embeddings as Knowledge Repositories

Common-sense knowledge is intrinsic to our perception of the world. We see a closed door and instantly understand that a new environment lies beyond it. We see gyrating reflections and immediately know we are looking at a body of water. Common-sense knowledge also helps us to make predictions: A dropped ball will bounce. A tipped glass will spill its contents. These and similar experiences are so ubiquitous that we seldom notice the assumptions we make or the way our expectations shape our experience.

We assert that one reason people find linguistic embeddings so fascinating is because they represent common-sense knowledge about the world in interpretable and manipulable ways. For example, you can add the words *Vietnam* and *Capitol* and get *Hanoi* [6], or you can calculate that *human - animal* =

ethics [25]. Somehow, although they have not been explicitly trained for it, the embedding models are allowing us to play with language and produce meaningful results. This is simultaneously intriguing and puzzling, particularly when one considers the way such embeddings are produced.

2.1 Overview of Embedding Models

Linguistic embeddings, also known as vector space models or distributed representations of words, rose to unprecedented prominence within the research community with the introduction of the GLoVE [26] and word2vec [22] algorithms, both of which use unsupervised methods to produce vector representations of words. Additional word-level embedding models followed, including the FastText algorithm [3], which utilizes subword information to enrich the resulting word vectors. More recently, the ELMo architecture [27] uses a deep bidirectional language model which was pre-trained on a large text corpus. The resulting word vectors are learned functions of the language model's internal states.

In 2016, Kiros et al. presented skip-thoughts [19], an extension to multi-word text of the context-prediction task used to train word2vec. Other neural embedding models quickly followed. Google's Universal Sentence Encoder [5] features two variants: A lightweight implementation that disregards syntax in favor of a quickly trainable bag-of-words representation [15], and a large model based on a Transformer architecture structured around attention mechanisms [32]. Most recently, the BERT architecture utilizes a multi-layer bidirectional Transformer encoder to create general purpose embeddings that generalize to a variety of downstream tasks [7].

2.2 Knowledge Extraction via Vector Offsets

A number of researchers have explored vector-based methods for extracting common-sense knowledge from learned embedding spaces. Georgios et al. used centroids of word embeddings in combination with Word Mover's distance in a biomedical document retrieval task [4]. Fulda et al. used a similar approach to determine object affordances in text-based video games [9]. Linguistic embeddings have also been used to link entities through an ontology [16], identify correlations between images and captions [17], and augment the behavior of regex matching [34]. A particularly interesting application is diachronic word embeddings [13], which were trained on a series of temporally discrete corpora and then used to analyze the evolution of cultural attitudes over time.

While these applications utilize the semantic and extractive properties of linguistic embeddings for common-sense reasoning, they also combine the embeddings with other computational techniques in order to achieve the desired result. Our work is distinct in that we explore the behavior of the embedding space directly via a form of n-shot learning in which a small number of example cases are used to generalize to a broader reasoning task.

3 Quantitative Experiments

To demonstrate the potential of linguistic embedding spaces to act as commonsense knowledge repositories, we apply a simple distance metric within the embedding space in order to solve three classification tasks.

1. **Task 1: Threat detection**
   Utilizing the Skyrim dataset presented in [10], we classify each human-generated caption as representing one of four possible interaction modes: Threat, Barter, Explore, or Puzzle. An example from the dataset is shown in Figure 1.
2. **Task 2: Emotional classification** Using a subset of the Daily Dialog dataset [20], we classify each sentence according to the emotion it expresses: anger, disgust, fear, happiness, sadness, surprise. (Sentences in which no emotion was expressed were removed from the dataset prior to classification.)
3. **Task3: Sentiment analysis** Using data from SemEval 2013 [33], we classify each tweet as being positive, negative or neutral/objective.

Generated Text		Label
Human text:	'An archer ready to fight against the enemy'	**Threat**

Fig. 1. Example image and associated caption from the SkyRim dataset. The goal of the algorithm was to determine which of four possible interaction modes was indicated based on the input text and a set of example sentences like those shown in Figure 2.

3.1 Classification Algorithm

Classification was accomplished *strictly* on the basis of cosine distance metrics within the embedding space. A set of ten exemplars per category is extracted from each dataset[1] prior to evaluation. During evaluation, each new sentence or tweet is assigned the same category as the nearest exemplar sentence. (Thus, we are using a KNN classification algorithm with K=1.)

The purpose of this highly simplified algorithm was to explore the native properties of the embedding space. We wanted to know how much common-sense knowledge was implicitly encoded within the geomoetry of the embeddings, and whether it was sufficient to solve sophisticated common-sense reasoning tasks. We specifically wanted tasks that did not rely on semantic similarity alone, but instead required the agent to distinguish between various categories of emotion, sentiment, or situation regardless of specific semantic content.

We compared results using several popular linguistic embedding models currently available for download, as well as a random baseline for comparison. It is worth noting that we also tried taking the centroid of the exemplars rather than using a nearest neighbor approach. This method performed worse overall.

Interaction Mode: Threat
'You see a soldier holding a sword'
'You are badly wounded'
'A massive troll bars the path'
'The bull paws the ground, then charges toward you'
'The poisonous spider advances, ready with its deadly bite'
'You are in danger'
'If you fall from this height, you will die'
'The battle rages around you'
'The angry man begins to attack you'
'You are plummeting to your death, with only a few seconds before you hit the ground'

Fig. 2. Example texts used to define the 'Threat' mode, meaning that an immediate physical danger is present. Similar example texts were available for the interaction modes 'Explore', 'Barter', and ''Puzzle'.

3.2 Results

Results are shown in Figure 3. Note that the interesting aspect of these experiments is not the classification accuracy per se, but what the results reveal about the underlying nature of the various embedding spaces. Our objective was to create a quantifiable measurement of the amount of semantic knowledge that

[1] In the case of the SkyRim dataset, we used the same exemplar sentences provided by the original authors

could be extracted from each embedding model on the basis of cosine similarity alone[2]. If such knowledge is demonstrably present and extricable, this provides a foundation for researchers to reconsider whether these properties should be explicitly encouraged via our evaluation metrics, rather than allowing them to develop haphazardly as a byproduct of current training methods.

	Skyrim	**Emotions**	**Sentiment**	**average**
Skip-thought	45.45%	39.48%	39.94%	41.62%
Google USE lite	54.54%	37.83%	38.96%	43.78%
Google USE large	**63.63**%	**43.45**%	**41.50**%	**49.53**%
Spacy	27.27%	20.06%	37.13%	28.15%
FastText	51.52%	28.80%	39.80%	40.04%
BERT	54.54%	35.79%	34.78%	41.70%
random	24.24%	16.62%	32.31%	24.39%

Fig. 3. Categorization accuracy on three n-shot tasks that require common-sense reasoning. The spacy vectors were generated using spacy version 2.0.11, which is based on a (possibly weighted) average of GLoVE vectors [29]. The FastText embedding was generated by averaging the FastText vectors for each word in the sentence. Other embeddings were generated using the models cited in their respective papers [19] [5] [7]. The highest accuracy in each column is bolded.

In all cases, the sole use of vector offsets within the embedding space is able to outperform a random baseline, thus demonstrating that some amount of semantic information and common-sense knowledge is present. At the same time, the generally poor performance of the algorithms reveals that the embedding spaces are not sufficiently structured to fully realize this potential. Of the algorithms explored, Google's large encoding model appears to be the most effective, with BERT, USE lite, and Skip-thought more or less tied for second place. A simple averaging of FastText word vectors performs remarkably well given that it retains no information about word order or grammatical structure.

It is interesting that performance on the sentiment analysis task is lower than on other tasks despite the relatively small number of categories. With only three options to choose from, one would expect the algorithms to perform better. We speculate that the abbreviations, urls and webisms of twitter may be functioning as distractors for embedding models that were trained on the more traditional text found in Google News, Wikipedia or the Toronto Book Corpus.

As mentioned earlier, cosine distance is not the only possible method for extracting semantic information from learned sentence representations. Other distance metrics such as correlation, Manhattan Distance, or Mahalanobis distance could be explored. But since the common practice among developers is

[2] Other extraction methods could also be explored, of course. But since the general usage of linguistic embeddings by hobbyists and developers relies on cosine distance as an estimate of semantic similarity, we chose to support that paradigm.

to take the cosine distance of word vectors when estimating their similarity, it seems logical to design an embedding space that matches these expectations.

3.3 Semantic Analysis

Linguistic embedding spaces are attractive for reasoning tasks because they contain implicit knowledge about language, causation, the physical behavior of objects, and the social behavior of humans. Unfortunately, the structure of currently-available embedding spaces does not fully utilize this potential. In particular, the inability to distinguish between polar opposites such as hot/cold, beautiful/ugly, or yes/no can become a hindrance to many analogical reasoning tasks, as can the inability to detect the difference between a sentence and its negation. Various applications ranging from embedding grammars [35] to language-based information transfer [8] would benefit from linguistic representations that made these distinctions easy to detect.

To determine the extent that these distinctions are represented in current state-of-the-art embedding spaces, we conducted a small case study based on cosine distance. Figure 4 shows the calculated distances between pairs of related sentences under six commonly used embedding models. Examination of the data reveals that across all six embedding models, semantically similar sentence pairs (e.g. "In Tahiti, the cat chased the dog" and "The cat chased the dog in Tahit") are consistently assigned a higher cosine distance than a semantically distinct pairing (e.g. "the cat chased the dog" and "the dog chased the cat"). Only one of the semantic distance challenges was successfully solved by any of the models.

	skipthought	**USE lite**	**USE large**	**spacy**	**Fasttext**	**Elmo**	**BERT**
the cat chased the dog the dog chased the cat	0.1269	0.0200	0.0045	0.0230	0.0000	0.0134	0.0060
In Tahiti, the cat chased the dog The cat chased the dog in Tahiti	0.4391	0.0491	0.0274	0.0710	0.0070	0.1590	0.1140
I am a cat I am not a cat	0.0686	0.0692	0.0776	0.1152	0.0250	**0.0891**	0.0670
I am a cat I am a domesticated cat	0.1393	0.0980	0.1405	0.1501	0.0717	**0.0745**	0.1851

Fig. 4. Case study exploring cosine distance between sentence pairs under various embedding models. Distance tuples that represent semantically appropriate relative distances are shown in bold-face text. Of the embedding models surveyed, only Elmo was able to rank semantically disparate sentences as having a higher cosine distance than a related synonymous pair, and it succeeded on only one of the two examples depicted.

This (small) case study suggests that serious semantic flaws are a common occurrence in current state-of-the-art embedding spaces. One might consider this a major setback, but from our point of view it represents a critical opportunity for linguistic embedding spaces to chart new territory. If one were able to design and train an embedding model that correctly reflects the semantic meaning of sentences via pairwise cosine distance, then a form of language-based commonsense reasoning, or 'reasoning in the linguistic domain', becomes possible. Such

an embedding space would facilitate reasoning tasks via vector offset methods, such as determining that *a jilted lover + a dangerous weapon + an argument late at night → murder*. At present, only word-level embedding spaces are able to function with such precision, but we envision a future in which things might be different.

4 Conclusions and Future Work

In this position paper we have shown that latent potential for language-based reasoning exists in current state-of-the-art embedding spaces. Our ability to leverage this potential, however, is limited by semantic flaws within the structure of the embedding space itself. We believe that this drawback can be overcome by re-examining our chosen evaluation metrics for neural language models.

As researchers continue to develop new architectures and training curricula for large language models, it becomes important to carefully consider what kind of performance we are measuring and whether it is leading us to the outcomes we desire. There is (obviously) nothing wrong with linguistic embeddings that are trained for, and evaluated based upon, the model's performance with respect to established task-transfer benchmarks. However, a myopic focus on benchmark-based evaluations might lead us to an increasingly large selection of embedding spaces that are increasingly unsuited for language-based reasoning tasks.

We strongly urge researchers to consider common-sense reasoning tasks based on cosine distance and vector offsets as potential evaluation metrics for future language models. By doing so, we will open the door to creating a new kind of embedding space, one that is able to facilitate effective task transfer while still enabling language-based reasoning. Such models, if we are able to develop them, could support research in fields such as as planning and prediction, explainable AI, question answering, and language-based interfaces.

Future work in this area should focus on neural architectures and training methods that are designed with the explicit goal of capturing semantic knowledge within the structure of the learned embedding space. Various network architectures including recurrent networks, convolutional networks, and transformers should be evaluated based on the quality of their learned embedding spaces instead of, or in addition to, their facilitation of downstream learning tasks. Finally, novel extraction methods should be customized to the nature of each kind of embedding space, and researchers should develop improved analytical methods for determining the amount of semantic knowledge contained within a linguistic embedding space.

References

1. Embedding projector. https://projector.tensorflow.org/.
2. Text similarity: Estimate the degree of similarity between publisher = Dandelion API, howpublished = https://dandelion.eu/semantic-text/text-similarity-demo/.
3. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
4. Georgios-Ioannis Brokos, Prodromos Malakasiotis, and Ion Androutsopoulos. Using centroids of word embeddings and word mover's distance for biomedical document retrieval in question answering. *CoRR*, abs/1608.03905, 2016.
5. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.
6. Adrian Colyer. The amazing power of word vectors. https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/, 2016.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
8. Nancy Fulda, Ben Murdoch Daniel Ricks, and David Wingate. Informing action primitives through free-form text. In *NIPS Workshop on Visually Grounded Interaction and Language*, 2017.
9. Nancy Fulda, Daniel Ricks, Ben Murdoch, and David Wingate. What can you do with a rock? affordance extraction via word embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1039–1045, 2017.
10. Nancy Fulda, Daniel Ricks, Ben Murdoch, and David Wingate. Threat, explore, barter, puzzle: A semantically-informed algorithm for extracting interaction modes. In *AAAI Workshop on Knowledge Extraction from Games*, 2018.
11. Nancy Fulda, Nathan Tibbetts, Zachary Brown, and David Wingate. Harvesting common-sense navigational knowledge for robotics from uncurated text corpora. In *Proceedings of the First Conference on Robot Learning (CoRL)*, 2017.
12. The Turku NLP Group. Word embeddings demo. http://bionlp-www.utu.fi/wv_demo/.
13. William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. *CoRR*, abs/1605.09096, 2016.
14. Cathal Horan. Using sentence embeddings to automate customer support, part one. https://blog.floydhub.com/automate-customer-support-part-two/, December 2018.
15. Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daume III. Deep unordered composition rivals syntactic methods for text classification. In *Association for Computational Linguistics*, 2015.
16. Ilknur Karadeniz and Arzucan Özgür. Linking entities through an ontology using word embeddings and syntactic re-ranking. *BMC bioinformatics*, 20(1):156, 2019.
17. Andrej Karpathy, Armand Joulin, and Li Fei-fei. Deep fragment embeddings for bidirectional image sentence mapping. In *In arXiv:1406.5679*, 2014.
18. Ryan Kiros. Sent2vec encoder and training code from the paper "skip-thought vectors". https://github.com/ryankiros/skip-thoughts, 2017.

19. Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. pages 3294–3302, 2015.
20. Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*, 2017.
21. Anthony Liu. Word to vec js demo. http://turbomaze.github.io/word2vecjson/, 2016.
22. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
23. Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. Association for Computational Linguistics, May 2013.
24. Chris Moody. A word is worth a thousand vectors. https://multithreaded.stitchfix.com/blog/2015/03/11/word-is-worth-a-thousand-vectors/, 2015.
25. Chris Nicholson. A beginner's guide to word2vec and neural word embeddings. https://skymind.ai/wiki/word2vec.
26. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.
27. Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
28. Hideki Shima. Ws4j demo. http://ws4jdemo.appspot.com/.
29. spaCy. Word vectors and semantic similarity. https://spacy.io/usage/vectors-similarity, 2016-2019.
30. username: DaveTheAl. Best practical algorithm for sentence similarity. https://datascience.stackexchange.com/questions/25053/best-practical-algorithm-for-sentence-similarity, 2017.
31. username: whs2k. How is the .similarity method in spacy computed? https://stats.stackexchange.com/questions/304217/how-is-the-similarity-method-in-spacy-computed, 2017.
32. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.
33. Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal, and Veselin Stoyanov. Sentiment analysis in twitter. http://www.cs.york.ac.uk/semeval-2013/task2/, 2013.
34. David Wingate, William Myers, Nancy Fulda, and Tyler Etchart. Embedding grammars. *arXiv preprint arXiv:1808.04891*, 2018.
35. David Wingate, William Myers, Nancy Fulda, and Tyler Etchart. Embedding grammars, 2018.