Overcoming Poor Word Embeddings with Word Definitions

Christopher Malon
NEC Laboratories America
Princeton, NJ 08540
malon@nec-labs.com

Abstract

Modern natural language understanding models depend on pretrained subword embeddings, but applications may need to reason about words that were never or rarely seen during pretraining. We show that examples that depend critically on a rarer word are more challenging for natural language inference models. Then we explore how a model could learn to use definitions, provided in natural text, to overcome this handicap. Our model's understanding of a definition is usually weaker than a well-modeled word embedding, but it recovers most of the performance gap from using a completely untrained word.

1 Introduction

The reliance of natural language understanding models on the information in pre-trained word embeddings limits these models from being applied reliably to rare words or technical vocabulary. To overcome this vulnerability, a model must be able to compensate for a poorly modeled word embedding with background knowledge to complete the required task.

For example, a natural language inference (NLI) model based on pre-2020 word embeddings may not be able to deduce from "Jack has COVID" that "Jack is sick." By providing the definition, "COVID is a respiratory disease," we want to assist this classification.

We describe a general procedure for enhancing a classification model such as natural language inference (NLI) or sentiment classification, to perform the same task on sequences including poorly modeled words using definitions of those words. From the training set $\mathcal{T}$ of the original model, we construct an augmented training set $\mathcal{T}'$ for a model that may accept the same token sequence optionally concatenated with a word definition. In the case of NLI, where there are two token sequences, the definition is concatenated to the premise sequence. Because $\mathcal{T}'$ has the same form as $\mathcal{T}$, a model accepting the augmented information may be trained in the same way as the original model.

Because there are not enough truly untrained words like "COVID" in natural examples, we probe performance by scrambling real words so that their word embedding becomes useless, and supplying definitions. Our method recovers most of the performance lost by scrambling. Moreover, the proposed technique removes biases in more *ad hoc* solutions like adding definitions to examples without special training.

2 Related Work

We focus on NLI because it depends more deeply on word meaning than sentiment or topic classification tasks. Chen et al. (2018) pioneered the addition of background information to an NLI model's classification on a per-example basis, augmenting a sequence of token embeddings with features encoding WordNet relations between pairs of words, to achieve a 0.6% improvement on the SNLI (Bowman et al., 2015) task. Besides this explicit reasoning approach, implicit reasoning over background knowledge can be achieved if one updates the base model itself with background information. Lauscher et al. (2020) follows this approach to add information from ConceptNet (Speer et al., 2018) and the Open Mind Common Sense corpus (Singh et al., 2002) through a fine-tuned adapter added to a pretrained language model, achieving better performance on subsets of NLI examples that are known to require world knowledge. Talmor et al. (2020) explore the interplay between explicitly added knowledge and implicitly stored knowledge on artificially constructed NLI problems that require counting or relations from a taxonomy.

In the above works, explicit background infor-

Proceedings of the 10th Conference on Lexical and Computational Semantics, pages 288–293
August 5–6, 2021, Bangkok, Thailand (online) ©2021 Association for Computational Linguistics

mation comes from a taxonomy or knowledge base. Only a few studies have worked with definition text directly, and not in the context of NLI. Tissier et al. (2017) used definitions to create embeddings for better performance on word similarity tasks, compared to word2vec (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017) while maintaining performance on text classification. Their work pushes together embeddings of words that co-occur in each other's definitions. Recently, Kaneko and Bollegala (2021) used definitions to remove biases from pretrained word embeddings while maintaining coreference resolution accuracy. In contrast, our work reasons with natural language definitions without forming a new embedding, allowing attention between a definition and the rest of an example.

Alternatively, Schick and Schütze (2020) improved classification using rare words by collecting and attending to all of the contexts in which they occur in BookCorpus (Zhu et al., 2015) combined with Westbury Wikipedia Corpus.[1] Like the methods above that use definitions, this method constructs a substitute or supplementary embedding for a rare word.

3 Methods

3.1 Critical words

The enhanced training set $\mathcal{T}'$ will be built by providing definitions for words in existing examples, while obfuscating the existing embeddings of those words. If a random word of the original text is obfuscated, the classification still may be determined or strongly biased by the remaining words. To ensure the definitions matter, we select carefully.

To explain which words of a text are important for classification, Kim et al. (2020) introduced the idea of input marginalization. Given a sequence of tokens $\mathbf{x}$, let $\mathbf{x}_{-i}$ represent the sequence without the ith token x_i. They marginalize the probability of predicting a class y_c over possible replacement words $\tilde{x_i}$ in the vocabulary $\mathcal{V}$ as

$$p(y_c|\mathbf{x}_{-i}) = \sum_{\tilde{x_i} \in \mathcal{V}} p(y_c|\tilde{x_i}, \mathbf{x}_{-i}) p(\tilde{x_i}|\mathbf{x}_{-i}) \quad (1)$$

and then compare $p(y_c|\mathbf{x}_{-i})$ to $p(y_c|\mathbf{x})$ to quantify the importance of x_i. The probabilities $p(\tilde{x_i}|\mathbf{x}_{-i})$ are computed by a language model.

We simplify by looking only at the classification and not the probability. Like Kim et al. (2020), we truncate the computation of $p(y_c|\tilde{x_i}, \mathbf{x}_{-i})$ to words such that $p(\tilde{x_i}|\mathbf{x}_{-i})$ exceeds a threshold, here .05. Ultimately we mark a word x_i as a *critical word* if there exists a replacement $\tilde{x_i}$ such that

$$\text{argmax}_y p(y|\tilde{x_i}, \mathbf{x}_{-i}) \neq \text{argmax}_y p(y|\mathbf{x}) \quad (2)$$

and

$$p(\tilde{x_i}|x_{-i}) > .05. \quad (3)$$

Additionally we require that the word not appear more than once in the example, because the meaning of repeated words usually impacts the classification less than the fact that they all match. Table 1 shows an example.

Premise	A young man sits, looking out of a *train* [side → Neutral, small → Neutral] window.
Hypothesis	The man is in his room.
Label	Contradiction

Table 1: An SNLI example, with critical words shown in italics and replacements shown in brackets.

A technicality remains because our classification models use subwords as tokens, whereas we consider replacements of whole words returned by `pattern.en`. We remove all subwords of x_i when forming $\mathbf{x}_{-i}$, but we consider only replacements $\tilde{x_i}$ that are a single subword long.

3.2 Definitions

We use definitions from Simple English Wiktionary when available, or English Wiktionary otherwise.[2] Tissier et al. (2017) downloaded definitions from four commercial online dictionaries, but these are no longer freely available online as of January 2021.

To define a word, first we find its part of speech in the original context and lemmatize the word using the `pattern.en` library (Smedt and Daelemans, 2012). Then we look for a section labeled "English" in the retrieved Wiktionary article, and for a subsection for the part of speech we identified. We extract the first numbered definition in this subsection. In practice, we find that this method usually gives us short, simple definitions that match the usage in the original text.

[1] http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html

[2] We use the 2018-02-01 dumps.

When defining a word, we always write its definition as "*word* means: *definition*." This common format ensures that the definitions and the word being defined can be recognized easily by the classifier.

3.3 Enhancing a model

Consider an example $(\mathbf{x}, y_c) \in \mathcal{T}$. If the example has a critical word $x_i \in \mathbf{x}$ that appears only once in the example, and $\tilde{x_i}$ is the most likely replacement word that changes the classification, we let $\mathbf{x}'$ denote the sequence where x_i is replaced by $\tilde{x_i}$, and let $y'_c = \text{argmax}_y p(y|\mathbf{x}')$. If definitions $\mathbf{h}_i$ and $\mathbf{h}'_i$ for x_i and $\tilde{x_i}$ are found by the method described above, we add $(\mathbf{x}, \mathbf{h}_i, y_c)$ and $(\mathbf{x}', \mathbf{h}'_i, y'_c)$ to the enhanced training set $\mathcal{T}'$.

In some training protocols, we scramble x_i and $\tilde{x_i}$ in the examples and definitions added to $\mathcal{T}'$, replacing them with random strings of between four and twelve letters. This prevents the model from relying on the original word embeddings. Table 2 shows an NLI example and the corresponding examples generated for the enhanced training set.

Original	A blond man is drinking from a public fountain. / The man is drinking water. / Entailment
Scrambled word	a blond man is drinking from a public yfcqudqqg. yfcqudqqg means: a natural source of water; a spring. / the man is drinking water. / Entailment
Scrambled alternate	a blond man is drinking from a public lxuehdeig. lxuehdeig means: lxuehdeig is a transparent solid and is usually clear. windows and eyeglasses are made from it, as well as drinking glasses. / the man is drinking water. / Neutral

Table 2: Adding background information to examples from SNLI

4 Experiments

4.1 Setup

We consider the SNLI task (Bowman et al., 2015). We fine-tune an XLNet (base, cased) model (Yang et al., 2019), because it achieves near state-of-the-art performance on SNLI and outperforms Roberta (Liu et al., 2019) and BERT (Devlin et al., 2019) on later rounds of adversarial annotation for ANLI (Nie et al., 2020). For the language model probabilities $p(\tilde{x_i}|\mathbf{x}_{-i})$, pretrained BERT (base, uncased) is used rather than XLNet because the XLNet probabilities have been observed to be very noisy on short sequences.[3]

One test set $SNLI_{crit}^{full}$ is constructed in the same way as the augmented training set, but our main test set $SNLI_{crit}^{true}$ is additionally constrained to use only examples of the form $(\mathbf{x}, \mathbf{h}_i, y_c)$ where y_c is the original label, because labels for the examples $(\mathbf{x}', \mathbf{h}'_i, y'_c)$ might be incorrect. All of our derived datasets are available for download.[4]

In each experiment, training is run for three epochs distributed across 4 GPU's, with a batch size of 10 on each, a learning rate of 5×10^{-5}, 120 warmup steps, a single gradient accumulation step, and a maximum sequence length of 384.

4.2 Results

Table 3 compares the accuracy of various training protocols.

Protocol	$SNLI_{crit}^{true}$
Original	85.1%
No scrambling, no defs	84.6%
No scrambling, defs	85.2%
Scrambling, no defs	36.9%
Scrambling, defs	81.2%
Scrambling, subs	84.7%
Train on normal SNLI, test on scrambled no defs	54.1%
Train on normal SNLI, test on scrambled defs	63.8%
Train on unscrambled defs, test on scrambled defs	51.4%

Table 3: Accuracy of enhancement protocols

Our task cannot be solved well without reading definitions. When words are scrambled but no definitions are provided, an SNLI model without special training achieves 54.1% on $SNLI_{crit}^{true}$. If trained on $\mathcal{T}'$ with scrambled words but no definitions, performance drops to 36.9%, reflecting that $\mathcal{T}'$ is constructed to prevent a model from utilizing the contextual bias.

With definitions and scrambled words, performance is slightly below that of using the original words. Our method using definitions applied

[3] https://github.com/huggingface/transformers/issues/4343

[4] https://figshare.com/s/edd5dc26b78817098b72

to the scrambled words yields 81.2%, compared to 84.6% if words are left unscrambled but no definitions are provided. Most of the accuracy lost by obfuscating the words is recovered, but evidently there is slightly more information accessible in the original word embeddings.

If alternatives to the critical words are not included, the classifier learns biases that do not depend on the definition. We explore restricting the training set to verified examples $\mathcal{T}'_{true} \subset \mathcal{T}'$ in the same way as the $SNLI^{true}_{crit}$, still scrambling the critical or replaced words in the training and testing sets. Using this subset, a model that is not given the definitions can be trained to achieve 69.9% performance on $SNLI^{true}_{crit}$, showing a heavy contextual bias. A model trained on this subset that uses the definitions achieves marginally higher performance (82.3%) than the one trained on all of $\mathcal{T}'$. On the other hand, testing on $SNLI^{full}_{crit}$ yields only 72.3% compared to 80.3% using the full $\mathcal{T}'$, showing that the classifier is less sensitive to the definition.

Noisy labels from replacements do not hurt accuracy much. The only difference between the "original" training protocol and "no scrambling, no defs" is that the original trains on $\mathcal{T}$ and does not include examples with replaced words and unverified labels. Training including the replacements reduces accuracy by 0.5% on $SNLI^{true}_{crit}$, which includes only verified labels. For comparison, training and testing on all of SNLI with the original protocol achieves 90.4%, so a much larger effect on accuracy must be due to harder examples in $SNLI^{true}_{crit}$.

Definitions are not well utilized without special training. The original SNLI model, if provided definitions of scrambled words at test time as part of the premise, achieves only 63.8%, compared to 81.2% for our specially trained model.

If the defined words are not scrambled, the classifier uses the original embedding and ignores the definitions. Training with definitions but no scrambling, 85.2% accuracy is achieved, but this trained model is unable to use the definitions when words are scrambled: it achieves 51.4%.

We have not discovered a way to combine the benefit of the definitions with the knowledge in the original word embedding. To force the model to use both techniques, we prepare a version of the training set which is half scrambled and half unscrambled. This model achieves 83.5% on the unscrambled test set, worse than no definitions.

Definitions are not simply being memorized. We selected the subset $SNLI^{new}_{crit}$ of $SNLI^{true}_{crit}$ consisting of the 44 examples in which the defined word was not defined in a training example. The definition scrambled model achieves 68.2% on this set, well above 45.5% for the original SNLI model reading the scrambled words and definitions but without special training. Remembering a definition from training is thus an advantage ($SNLI^{true}_{crit}$ accuracy was 81.2%), but not the whole capability.

Definition reasoning is harder than simple substitutions. When definitions are given as one-word substitutions, in the form "*scrambled* means: *original*" instead of "*scrambled* means: *definition*", the model achieves 84.7% on $SNLI^{true}_{crit}$ compared to 81.2% using the definition text. Of course this is not a possibility for rare words that are not synonyms of a word that has been well trained, but it suggests that the kind of multi-hop reasoning in which words just have to be matched in sequence is easier than understanding a text definition.

4.3 A hard subset of SNLI

By construction of the SentencePiece dictionary (Kudo and Richardson, 2018), only the most frequent words in the training data of the XLNet language model are represented as single tokens. Other words are tokenized by multiple subwords. Sometimes the subwords reflect a morphological change to a well-modeled word, such as a change in tense or plurality. The language model probably understands these changes well and the subwords give important hints. The lemma form of a word strips many morphological features, so when the lemma form of a word has multiple subwords, the basic concept may be less frequently encountered in training. We hypothesize that such words are less well understood by the language model.

To test this hypothesis, we construct a subset $SNLI^{true}_{multi}$ of the test set, consisting of examples where a critical word exists whose lemma form spans multiple subwords. This set consists of 332 test examples. The critical word used may be different from the one chosen for $SNLI^{true}_{crit}$. This subset is indeed harder: the XLNet model trained on all of SNLI attains only 77.7% on this subset using no definitions, compared to 90.4% on the original test set.

In Table 4 we apply various models constructed in the previous subsection to this hard test set. Ideally, a model leveraging definitions could compen-

Protocol	$SNLI^{true}_{multi}$
Normal SNLI on unscrambled	77.7%
Defs & unscrambled on defs & unscrambled	77.1%
Defs & some scrambling on defs & unscrambled	73.8%
Defs & scrambled on defs & scrambled	69.9%
Defs & scrambled on defs & unscrambled	62.7%

Table 4: Accuracy on the hard SNLI subset

sate for these weaker word embeddings, but the method here does not do so.

5 Conclusion

This work shows how a model's training may be enhanced to support reasoning with definitions in natural text, to handle cases where word embeddings are not useful. Our method forces the definitions to be considered and avoids the application of biases independent of the definition. Using the approach, entailment examples like "Jack has COVID / Jack is sick" that are misclassified by an XLNet trained on normal SNLI are correctly recognized as entailment when a definition "COVID is a respiratory disease" is added. Methods that can leverage definitions without losing the advantage of partially useful word embeddings are still needed. In an application, it also will be necessary to select the words that would benefit from definitions, and to make a model that can accept multiple definitions.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *arXiv preprint*, 1607.04606.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*, 1810.04805.

Masahiro Kaneko and Danushka Bollegala. 2021. Dictionary-based debiasing of pre-trained word embeddings. *arXiv preprint*, 2101.09525.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020. Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online. Association for Computational Linguistics.

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. *arXiv preprint*, 1907.11692.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint*, 1301.3781.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online. Association for Computational Linguistics.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and*

ODBASE, pages 1223–1237, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *Journal of Machine Learning Research*, 13(66):2063–2067.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge. *arXiv preprint*, 1612.03975.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2017. Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Copenhagen, Denmark. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.