Open Source for DG embeddings and DG-BERT

The versions of our PyTorch and Huggingface transformers are "1.9.0+cu111" and "4.10.2" respectively for pre-training DG-BERT.

1) DG embeddings

To use ParetoPrinciple.py, filtering (lemmatize; remove stop words, conjunctions, punctuation and duplicates) dictionary resources is required beforehand. A filtered example is given in "**entry_2_filtered_definition_example.csv**" (only 5,000 examples for demo, since there might exist the restriction for dataset redistribution. But the resources that we use are open-source, and you can obtain them from the links provided in those papers). Also, preparation for feature-frequencies is required; the full example in our case is given in "**feature_frequency.csv**".

After ParetoPrinciple.py, the file "**entry_2_extracted_features_example.csv**" can be obtained (same as above, only 5,000 examples are given for demo). Truncated SVD should be used to factorize the definitions in this file. In our case, we save the factorized matrix into a "**svd_latent_representation_768d.csv**" file (available under this link).

To use DG_embeddings.py, "**entry_2_extracted_features_example.csv**", "**svd_latent_representation_768d.csv**", and **vocab.txt (BERT vocabulary)** are used as input, and the output is "DGembeddings.pth" saved under the path named "DG" (available under this link).

2) DG-BERT

The pre-trained model is available under this link (Note that the current model is pre-trained with only max-seq-length 448, and we will update the new model pre-trained with 512 once done.).

"DGmodeling.py" is the source code for DG-BERT, which is based on Huggingface bert_modeling. There are only two differences:

1) in lines 149-166, the code for DG, scalar and bias embeddings are added, which is further used in the lines 175, 203 and 207 for the addition with Bert Embeddings;

2) in lines 982-995, the code for initializing these embeddings is added (when BertForPreTraining is invoked, the weight initialization should be done correspondingly, the weight for DG embeddings is under the path named “DG”).

For fine-tuning tasks, “import DGmodeling” should be in your .py scripts, and then as usual, different classes are used for different tasks such as

“model = DGmodeling.BertForSequenceClassification.from_pretrained()”, and the pre-trained model is under the path named “pretrained”.

Also, if you want to use DG-BERT without the whole package of DG, scalar and bias embeddings, e.g., just “from transformers import BertForSequenceClassification”. Then “model = BertForSequenceClassification.from_pretrained()”, and the pre-trained model is under the path named “pretrained”.

Please use the BibTeX below for citation.

@article{LIU2024111883,
title = {DG Embeddings: The unsupervised definition embeddings learned from dictionary and glossary to gloss context words of Cloze task},
journal = {Knowledge-Based Systems},
volume = {296},
pages = {111883},
year = {2024},
issn = {0950-7051},
doi = {https://doi.org/10.1016/j.knosys.2024.111883},
url = {https://www.sciencedirect.com/science/article/pii/S0950705124005173},
author = {Xiaodong Liu and Rafal Rzepka and Kenji Araki},
keywords = {Unsupervised definition embeddings, Semantic features of glosses, Context words, Auto-encoding models, Natural language processing},
abstract = {For both humans and machines to acquire vocabulary, it is effective to learn words from context while using dictionaries as an auxiliary tool. It has been shown in previous linguistic studies that for humans, glossing either target words to be learned or words comprising context is an effective approach. For machines, however, previous NLP studies are mainly focused on the former. In this paper, we investigate the potentiality of context words-glossed setting. During pre-training BERT, to infuse context words with semantic features of glosses, we propose DG embeddings — the unsupervised definition embeddings learned from dictionaries and glossaries. To employ unsupervised learning is inspired by a real-world scenario of dictionary use called headword search. This can also prevent a technical duplicate from happening, as learning words from context is already based on auto-encoding models with self-supervised learning. BERT-base is used for evaluation, and we refer to BERT-base with DG embeddings as DG-BERT. According to our experimental results, compared to the vanilla BERT, DG-BERT shows the following strengths: faster pre-training convergence, noticeable improvements on various downstream tasks, a better grasp of figurative semantics, more accurate self-attention for collocation of phrases, and higher sensitivity to context words for target-word predictions in psycholinguistic diagnostics.}
}