BERT Multitask Learning in a Semi-Supervised Learning Setup

Danhua Yan

Department of Computer Science, Stanford University

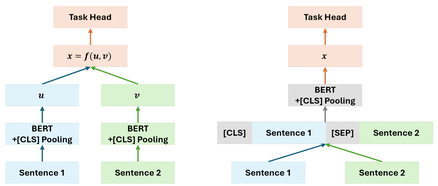
Introduction

- **Objective** Enhance $\text{BERT}_{\text{BASE}}$ embeddings across tasks like sentiment classification, paraphrase detection, and textual similarity using multitask learning and semi-supervised learning (SSL) settings.
- **Motivation** Fine-tuning BERT for multiple NLP tasks requires high-quality labeled data, which is often scarce. SSL provides a promising approach in training using limited labeled datasets.

Background

- **BERT Model** Developed by Devlin et al., utilizes transformer architecture for contextual word relationships.
- **Consistency Learning** One of the key areas in SSL field, stands out for its proven effectiveness across numerous benchmarks. This approach ensures that a model produces consistent outputs for an unlabeled example, even when subjected to minor perturbations, such as the introduction of small noise.
- **Unsupervised Data Augmentation (UDA)** Proposed by Xie et al. [1], demonstrates that effective data augmentation of unsupervised label can significantly enhance the performance of supervised learning of NLP tasks on limited labeled data.

Baseline Extensions



(a) Archetecture 1: Pooling seperate embeddings

(b) Archetecture 2: Fused sentence embedding

Figure 1. Two architectures for sentence-pair inputs: (a) Each sentence is BERT-encoded separately, then pooled into a single embedding. (b) Sentence-pairs are concatenated with a `[SEP]` token and BERT-encoded into one embedding.

Baseline Uses frozen pre-trained $\text{BERT}_{\text{BASE}}$ embeddings with a single linear projection layer as task head.

Combining Sentence Embeddings (Figure 1)

- Pooling Separate Embeddings: Generates a joint embedding from two sentences encoded separately by BERT.
- Fused Sentence Embedding: Utilizes BERT's inherent training with sentence pairs separated by the `[SEP]` token to encode sentence context similarities.

Training Strategies

- Sequential Training: Tasks are trained in succession within each epoch. Specifically, model weights are updated per batch, with three tasks forming a batch queue in the order of paraphrase detection, textual similarity, and sentiment analysis.
- Simultaneous Training: Aggregates the losses for all tasks concurrently. Each batch comprises three different tasks of the same batch size, and losses are summed without weighting.

UDA Framework for Semi-Supervised Learning

Advanced Data Augmentation Generating augmented text for unlabeled datasets, via prompting LLaMA-3 Chat (8B) Model differently.

- Back-translation: Translates English sentences to French then back to English
- Sentence Completion: Completes sentence followed by "To put it differently,"
- Random-mask Completion: Completes randomly masked sentence with similar amount of words

Training Signal Annealing (TSA) Dynamically selects a subset of the labeled dataset at each training step t, we test linear, log, and exponential release schedule. Different threshold functions dictate the rate at which training signals of labeled examples are released. As shown in Figure 2, the TSA component proves effectiveness in mitigating overfitting during early epochs.

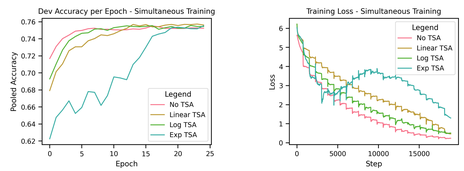


Figure 2. Comparing characteristics of different TSA threshold functions

Confidence Masking To focus the unsupervised loss on distribution discrepancies arising from suboptimal data augmentation, we calculate the loss only on a subset of the unlabeled data where the model is confident in its prediction.

UDA Loss Function Loss function is a combination of supervised mean cross-entropy loss, and unsupervised mean KL-divergence loss between unsupervised and augmented data. TSA and confidence masking is incorporated in UDA framework by dynamically calculates loss on a subset of samples based on model performance.

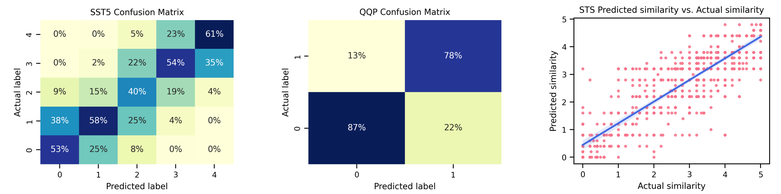
$$\mathcal{L} = -\mathbb{E}\left[\sum_{i=1}^{|L_t|} y_i \log p_\theta(x_i)\right] - \lambda \mathbb{E}\left[\sum_{j=1}^{|U_t|} p_{\tilde{\theta}}(x_j) \log \frac{p_{\tilde{\theta}}(x_j)}{p_\theta(\hat{x}_j)}\right]$$

Results

The best performing model on dev set is using fused sentence embeddings + Linear TSA, training simultaneously. **Our highest performing model has test set overall accuracy of 0.763, 0.517 on SST5, 0.839 on QQP, and 0.866 on STS separately.**

Model	SST5	QQP	STS	Acc
Baseline (last-layer only)	0.309	0.667	0.209	0.527
Arc1-Simple Concat	0.479	0.738	0.369	0.634
Arc1-Absolute difference	0.510	0.733	0.532	0.670
Arc1-Dot Product Attention	0.514	0.737	0.486	0.665
Arc2-Fused Sentence Embedding (BE)	0.501	0.829	0.852	0.752
BE + Linear TSA	0.520	0.836	0.861	0.762
BE + Linear TSA + UDA Back Translation	0.520	0.821	0.862	0.758
BE + Linear TSA + UDA Sentence Completion	0.506	0.820	0.853	0.751
BE + Linear TSA + UDA Random-mask Completion	0.520	0.808	0.827	0.747

Analysis



(a) SST5 Dev Set Confusion Matrix normalized to predicted labels

(b) QQP Dev Set Confusion Matrix normalized to predicted labels

(c) STS Correlation Plot, line showing best fit

Figure 3. Analysis of Dev set performance of the model

In essence, the model exhibits a strong grasp of sentiment analysis and sentence-pair similarity analysis. Mis-classifications primarily occur between adjacent categories. Despite not explicitly training using an ordinal categorical approach, the model effectively learns and interprets sentiment group relationship. On the hand, the model has challenges in:

- Fine-grained Sentiment Analysis SST5: Text ambiguities and human evaluator biases for neutral classes.
- Paraphrase Detection QQP: The model struggles with precision in positive class cases, especially when sentences have similar words in different orders or ambiguous human labels.
- Textual Similarity STS: The model tends to produce a narrower range of scores and has difficulty with extreme score assignments when sentences share common words.

Conclusion

- **Key Findings:** Multitask learning strategies significantly enhance BERT's performance across multiple NLP tasks. The fused sentence embedding approach combined with Linear TSA proves most effective.
- **Limitations:** Quality of data augmentation generated by LLaMA-3 has redundancy that undermines UDA's effectiveness.
- **Future Work:** Explore more efficient data augmentation techniques and advanced language features to further improve model robustness.

Ethics Statements

- **Environmental impact:** UDA escalates computational expenses due to generation of augmented datasets for large amount of unsupervised datasets via PLM prompts. Recent progress in few-shot and zero-shot learning with advanced PLMs suggests more efficient training paradigms.
- **Bias and toxicity amplification:** Existing biases or toxic language in training data and PLMs can be exacerbated in data augmentation process. Employing existing models or APIs to scrutinize PLM-generated data can aid in filtering out biased and toxic content and rephrasing biased text, thereby mitigating the amplification issue.

Reference

[1] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

dhyan@stanford.edu