PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks

Yufei Wang\textsuperscript{1}, Can Xu\textsuperscript{2}, Qingfeng Sun\textsuperscript{2}, Huang Hu\textsuperscript{2}, Chongyang Tao\textsuperscript{2}, Xiubo Geng\textsuperscript{2}  Daxin Jiang\textsuperscript{2}

Macquarie University, Sydney, Australia\textsuperscript{1}
Microsoft Corporation, Beijing, China\textsuperscript{2}

Paper https://arxiv.org/abs/2202.12499
Code https://github.com/GaryYufei/PromDA

April 4, 2022
Table of Contents

Data Augmentation for NLP

Motivations

Method

Experiment

Conclusion
Table of Contents

Data Augmentation for NLP

Motivations

Method

Experiment

Conclusion
Data Augmentation (DA) aims to create more training data from the existing training data. There are two types of DA technologies:

- **Continuous DA**: Dropout, SeqMix. These methods often manipulate the continuous embeddings of the inputs.
- **Discrete DA**: EDA, LAMBADA. These methods directly produce discrete training instances (e.g., sentences and labels).

This paper focuses on the **Discrete DA**.
The settings in this paper

The low-resource NLU task only has a small set of labeled training data $\mathcal{T} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$. The Data Augmentation Algorithm generates synthetic labeled training data $\mathcal{T}_{LM} = \{({\hat{x}}_1, {\hat{y}}_1), \cdots, ({\hat{x}}_n, {\hat{y}}_n)\}$ from $\mathcal{T}$ using (Pre-trained) language models. The goal is that the NLU models trained using $\mathcal{T} \cup \mathcal{T}_{LM}$ outperform the NLU models only trained using $\mathcal{T}$. 
Table of Contents

Data Augmentation for NLP

Motivations

Method

Experiment

Conclusion
Existing Solution for Low-Resource NLU

Previous works often produce extra “labeled data” to boost the performance of the Low-Resource NLU models.

- [4] deploys the *self-training* framework to produce *pseudo labelled training data* from *unlabeled in-domain data* which could be expensive to obtain.

- [5, 1] expand the original small training data using automatic heuristic rules, which could distort the text.
To solve the above dilemma, Language Models (LMs) or Pre-trained Language Models (PLMs) are used for data augmentation.

PLMs could be trained to generate synthetic training data.

However, in the low-resource NLU tasks, directly fine-tuning PLMs could result in over-fitting. PLMs memorize the small training data.
Recent work [2, 3] propose prompt tuning, which only back-propagates the loss to the *Soft Prompts*, instead of the entire model. They show that prompt tuning is sufficient to be competitive with full model tuning while significantly reducing the amount of parameters to be tuned.
Soft Prompt

Fine-tuning

Transformer

name  Clowns  type  Britain  [SEP] Clowns  serves  British  food

Prefix-tuning

Transformer

[Prefix] [Prefix] name  Clowns  type  Britain  [SEP] Clowns  serves  British  food
Table of Contents

Data Augmentation for NLP

Motivations

Method

Experiment

Conclusion
Overview of PromDA

Seq2Seq PLMs with Soft Prompt

| Encoder | Decoder |
|---------|---------|
| ![Diagram](Image1) | ![Diagram](Image2) |

Dual-View Data Augmentation

| Output View | Input View |
|-------------|------------|
| [Positive]  | [enjoy, watching]  |
|Positive: I really enjoy watching this movie | Positive: I really enjoy watching this movie |

Consistency Filtering

| Synthetic | Data |
|-----------|------|
| ![Diagram](Image3) | ![Diagram](Image4) |

Few-shot Data $\mathcal{T}$
Seq2Seq PLMs with Soft Prompt

We prepend a sequence of trainable vectors at each transformer layer. We denote $P^j = \{p^j_1, \cdots, p^j_k\}$ as the Soft Prompt at the $j^{th}$ layer. The $i^{th}$ hidden states at the $j^{th}$ layer $h^j_i$ in the Transformer model is defined as follows:

$$h^j_i = \begin{cases} 
p^j_i & i \leq k \\
w_i & i > k \land j = 0 \\Trans(h^{j-1}_i) & \text{Otherwise} \end{cases}$$ (1)

where $Trans(\cdot)$ is the forward function the Transformer layer and $w_i$ is the fixed word embedding vector at the input layer.
Pre-training for Prompt Initialization

- The parameter initialization of the *Soft Prompt* $P$ could be important.
- Given that data augmentation produces full syntactic data from partial information (e.g., output tags and keywords), we propose *Synonym Keywords to Sentence* pre-training task.
- We only use the task-agnostic general-purpose pre-training corpus.
Dual-View DA

**Sequence Labelling**

GT: \[\text{Org All Fishermen ’s Association} \text{ secretary [Per N.J. Bose]} \text{ said the strike would continue indefinitely.}\]
IV: \text{All Fishermen ’s Association and N.J. Bose and strike and indefinitely}\nOV: \text{Organization and Person}

**Sentence Classification**

GT: \text{The story has its redundancies, and the young actors, not very experienced, are sometimes inexpressive. Negative}\nIV: \text{redundancies and young actors and experienced and inexpressive}\nOV: \text{Negative}
Dual-View with Different Prompt

After prompt pre-training, we treat Input View and Output View as two independent models and use the Soft Prompt parameters $P$ to initialize the parameters of $P_{input}$ and $P_{output}$ to ensure the diversity.
Consistency Filtering

Given synthetic data with generated labels produced by PromDA, we use the NLU models to label these data again and only keep the instances with consistent outputs from PromDA and the NLU models. We iterate this process $N$ times to obtain stronger NLU models.
Final Algorithm of PromDA

Algorithm 1 Dual-View Data Augmentation:
Given few-shot labeled dataset $\mathcal{T}$, the number of iteration $N$; return a trained NLU model $M_{NLU}$.

1: procedure DUALVIEWDA($\mathcal{D}, N$)
2: \hspace{1em} $M_{LM} \leftarrow \text{TRAIN}(LM, \mathcal{T})$
3: \hspace{1em} $\mathcal{T}_I^1 \leftarrow \text{GEN}(M_{LM}, \mathcal{T}, I)$ \hspace{1em} $\triangleright$ Input
4: \hspace{1em} $\mathcal{T}_O^1 \leftarrow \text{GEN}(M_{LM}, \mathcal{T}, O)$ \hspace{1em} $\triangleright$ Output
5: \hspace{1em} $\mathcal{T}_I^2 \leftarrow \text{GEN}(M_{LM}, \mathcal{T}_I^1, I)$
6: \hspace{1em} $\mathcal{T}_O^2 \leftarrow \text{GEN}(M_{LM}, \mathcal{T}_I^1, O)$
7: \hspace{1em} $\hat{T}_{LM} \leftarrow \mathcal{T}_I^1 \cup \mathcal{T}_I^2 \cup \mathcal{T}_O^1 \cup \mathcal{T}_O^2$
8: \hspace{1em} $M_{NLU}^0 \leftarrow \text{TRAIN}(NLU, \mathcal{T})$
9: \hspace{1em} for $r \in 1, \ldots, N$ do
10: \hspace{2em} $\mathcal{T}_{LM}^r \leftarrow \text{CONSIST}(M_{NLU}^{r-1}, \hat{T}_{LM})$
11: \hspace{2em} $\mathcal{T}^r \leftarrow \mathcal{T}_{LM}^r \cup \mathcal{T}$
12: \hspace{2em} $M_{NLU}^r \leftarrow \text{TRAIN}(NLU, \mathcal{T}^r)$
13: \hspace{1em} $M_{NLU} \leftarrow M_{NLU}^N$
14: return $M_{NLU}$
Table of Contents

Data Augmentation for NLP

Motivations

Method

Experiment

Conclusion
Experiments on Sequence Labelling

| Dataset  | C03  | Wiki |
|----------|------|------|
| Shot     | 10   | 50   |
| Baseline | 72.7 | 82.9 | 50.8 | 65.4 |
| SDANER   | 72.9 | 82.8 | 51.7 | 65.8 |
| LAMBADA  | 75.0 | 83.7 | 52.9 | 66.4 |
| MetaST   | 76.7 | 83.6 | 56.6 | 69.2 |
| PromDA   | 77.5 | 84.1 | 58.3 | 70.1 |
# Experiments on Sentence Classification

| DataSet   | SST2  | RT   |
|-----------|-------|------|
| Shot      | 10    | 50   |
| Baseline  | 66.1  | 81.5 |
| EDA†      | 66.7  | 80.4 |
| Back T.   | 70.0  | 81.4 |
| CBERT‡    | 67.8  | 83.4 |
| LAMBADA   | 70.6  | 82.0 |
| PromDA    | 81.4  | 86.3 |
## Ablation Study

| DataSet                        | C03  | SST2 | Ave.  |
|-------------------------------|------|------|-------|
| Few-shot NLU Baseline         | 72.7 | 66.1 | 69.4  |
| **PromDA**                    | 77.5 | 81.4 | 79.5  |
| **Ablation for PT Pre-Training** |     |      |       |
| No PT                         | 75.2 | 74.5 | 74.9  |
| No PT Pre-Training            | 74.0 | 78.2 | 76.1  |
| Full Pre-Training             | 75.0 | 72.0 | 73.5  |
| LM Adaptation                 | 75.4 | 73.3 | 74.4  |
| **Ablation for Dual-View DA**  |     |      |       |
| Output Only                   | 75.6 | 81.0 | 78.0  |
| Input Only                    | 74.4 | 70.6 | 72.5  |
| Single Prompt                 | 76.7 | 79.5 | 78.1  |
Ablation Study on Iteration

| Setup | w/o Filtering | Iter-1 | Iter-2 | Iter-3 |
|-------|---------------|--------|--------|--------|
| C03   | 72.0          | 76.7   | 77.6   | 77.5   |
| SST2  | 69.2          | 77.5   | 79.7   | 81.4   |
## Diversity Analysis

| Model       | NM↑  | Self-B↓ | F1↑  |
|-------------|------|---------|------|
| **CoNLL03** |      |         |      |
| SDANER      | 141.4| 0.770   | 72.9 |
| LAMBADA     | 107.6| 0.761   | 75.0 |
| **PromDA**  | 351  | 0.259   | 77.5 |
| **SST2**    |      |         |      |
| EDA         | 59.6 | 0.889   | 66.7 |
| BackT.      | 101.8| 0.826   | 70.0 |
| CBERT       | 127  | 0.900   | 67.8 |
| LAMBADA     | 51.8 | 0.926   | 70.6 |
| **PromDA**  | 276  | 0.578   | 81.4 |
Unlabelled Data

We design three settings: *Unlabeled In-domain Data* (UID), *Unlabeled Near-domain Data* (UND) and *Unlabeled General-domain Data* (UGD) where the unlabeled data come from exactly same, similar and general-purpose domains.

| Dataset | C03 | Wiki | SST2 | RT  | Δ   |
|---------|-----|------|------|-----|-----|
| Baseline| 72.7| 50.8 | 66.1 | 57.8| -   |
| w/ UID  | 76.2| 55.2 | 70.2 | 59.7| +3.5|
| w/ UND  | 71.5| 51.3 | 69.3 | 59.4| +1.0|
| w/ UGD  | 64.6| 44.8 | 66.4 | 58.7| -3.2|
| **PromDA** | | | | | |
| w/ UID  | 80.0| 61.7 | 83.0 | 73.9| +12.8|


Table of Contents

Data Augmentation for NLP

Motivations

Method

Experiment

Conclusion
Conclusion

In this paper, we present the first prompt-based pre-trained language model PromDA for low-resource NLU data augmentation. Experiments on four benchmarks show the effectiveness of our proposed PromDA method. In the future, we plan to expand PromDA to other NLP tasks, including question answering, machine reading comprehension and text generation tasks.
References

[1] Xiang Dai and Heike Adel. “An Analysis of Simple Data Augmentation for Named Entity Recognition”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 3861–3867. DOI: 10.18653/v1/2020.coling-main.343. URL: https://aclanthology.org/2020.coling-main.343.

[2] Brian Lester, Rami Al-Rfou, and Noah Constant. “The Power of Scale for Parameter-Efficient Prompt Tuning”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. DOI: 10.18653/v1/2021.emnlp-main.243. URL: https://aclanthology.org/2021.emnlp-main.243.

[3] Xiang Lisa Li and Percy Liang. “Prefix-Tuning: Optimizing Continuous Prompts for Generation”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353.

[4] Yaqing Wang et al. “Meta Self-training for Few-shot Neural Sequence Labeling”. In: *SIGKDD 2021 (Research Track)*. Aug. 2021.

[5] Jason Wei and Kai Zou. “EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks”. 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6382–6388. DOI: 10.18653/v1/D19-1670. URL: https://aclanthology.org/D19-1670.