

Paper Review #6

[EMNLP 2020] Self-Supervised Meta-Learning
for Few-Shot Natural Language Classification Tasks

Dept. of Computer Science & Engineering
202122029 Meeyun Kim

2021. 08. 23.

Contents

1. Introduction

2. Method

3. Experiments

1. Introduction

Self-supervised learning

- **Automatically** generate labels.
- ex) **Masked Language Modeling**
- Can **avoid meta-overfitting**, and can **solve memorization problem**.

2. Method

SMLMT(Subset Masked Language Modeling Tasks)

Subset: {Democratic, Capital}



Support set

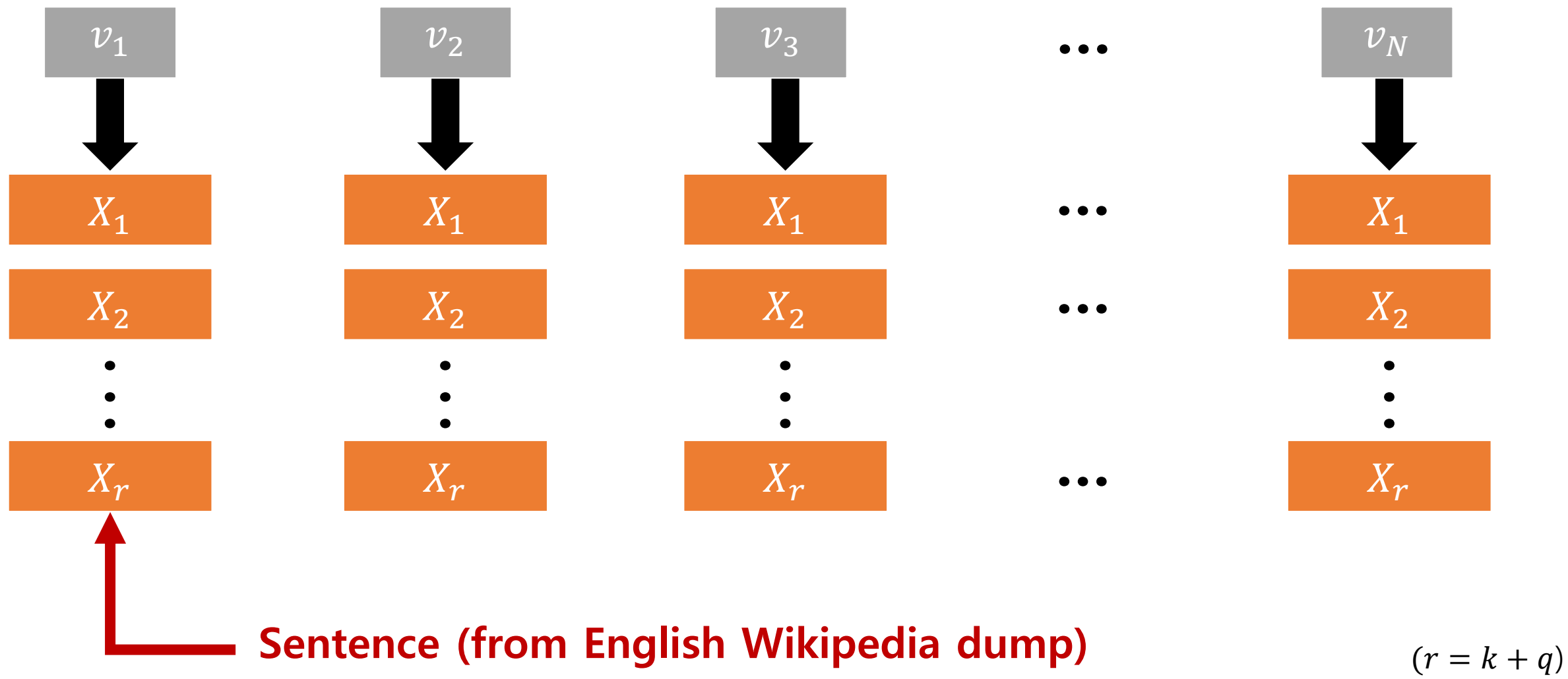
Sentence	Class
A member of the [m] Party, he was the first African American to be elected to the presidency.	1
The [m] Party is one of the two major contemporary political parties in the United States, along with its rival, the Republican Party.	1
Honolulu is the [m] and largest city of the U.S. state of Hawaii.	2
Washington, D.C., formally the District of Columbia and commonly referred to as Washington or D.C., is the [m] of the United States.	2

Query: New Delhi is an urban district of Delhi which serves as the [m] of India

Correct Prediction: 2

SMLMT (Subset Masked Language Modeling Tasks)

Creating an N-way classification task with SMLMT



SMLMT(Subset Masked Language Modeling Tasks)

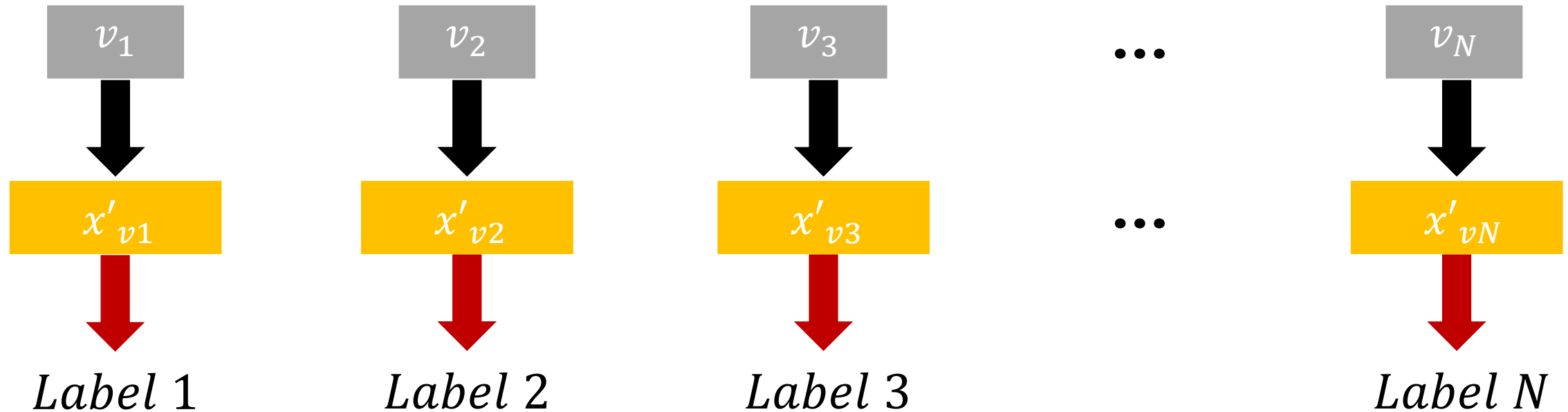
Creating an N-way classification task with SMLMT (Cont'd)

$$x_{vi} \rightarrow x'_{vi} = \{\text{Mask}(X_1, v_i), \dots, \text{Mask}(X_r, v_i)\}$$



SMLMT(Subset Masked Language Modeling Tasks)

Creating an N-way classification task with SMLMT (Cont'd)

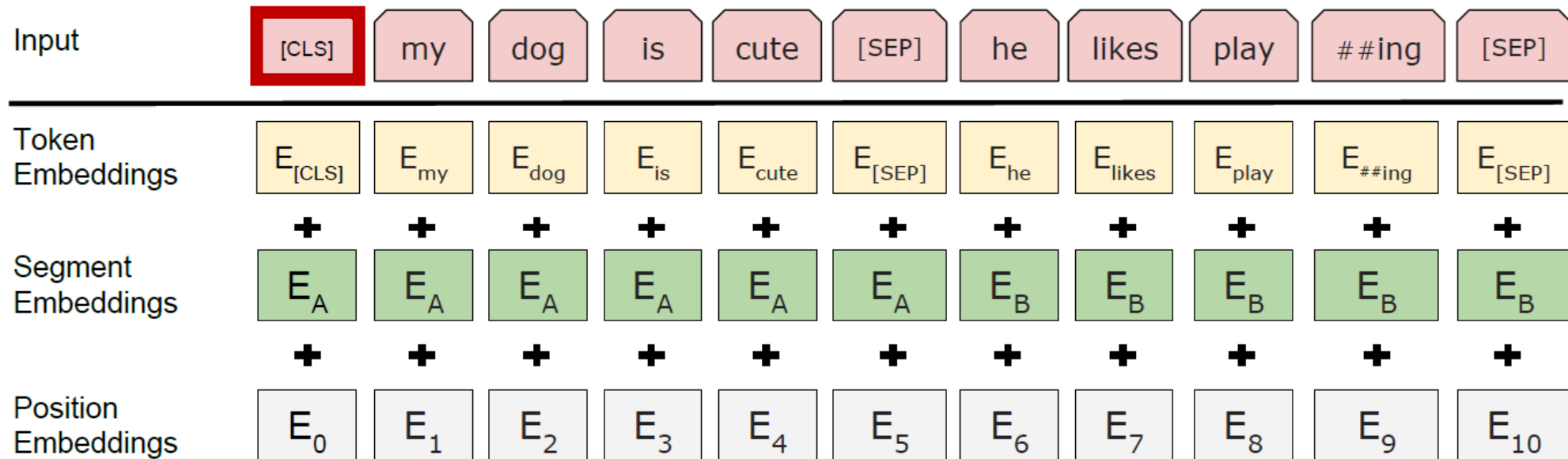


Model cannot memorize the input-label!

Meta-learning Model

Text encoder

- Follow the **BERT** model.
- Add a special **CLS token** to the start of the input.



Meta-learning Model

MAML(Model-Agnostic Meta-Learning)

Learn

$$\theta'_i \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}_i(\mathcal{D}^{tr}, \theta)$$

Meta learn

$$\Theta \leftarrow \Theta - \beta \nabla_{\Theta} \mathbb{E}_{T_i \sim \mathcal{P}(\mathcal{T})} \left[L_i(\mathcal{D}^{val}, \theta'_i) \right]$$

$(\Theta := \{\theta, \alpha\})$

3. Implementation

Implementation of SMLMT

Models (same transformer architecture)

- **BERT** : used the **cased base model**.
- **MT-BERT** : **multi-task learning model**. (proposed in Bansal et al. (2019))
- **MT-BERT_{softmax}** : only the softmax layer is fine-tuned.
- **LEOPARD** : **meta-learning model** (proposed in Bansal et al. (2019))
- **SMLMT** : **Self-supervised meta-learning model**
- **Hybrid-SMLMT** : combination of **SMLMT** and **supervised tasks**.

Implementation of SMLMT

Hyper-parameters

Hyper-parameter	Value
Tasks per batch	4
Support samples per task	80
Query samples per task	10
Number of classes in SSLMT	[2,3,4]
d	256
Attention dropout	0.1
Hidden Layer Dropout	0.1
Outer Loop Learning Rate	1e-05
Adaptation Steps (G)	7
λ	0.5
Meta-training Epochs	1
Lowercase text	False
Sequence Length	128
Learning-rate Warmup	10% of steps

- SMLMT & Hybrid-SMLMT
use the **same set of hyper-parameter**.

← **Not valid for SMLMT!**