MEDIATEK research

Cross-Lingual Transfer with MAML on Trees

Adapt-NLP 2021

Jezabel R. Garcia, Federica Freddi, Feng-Ting Liao, Jamie McGowan, Tim Nieradzik, Da-shan Shiu, Ye Tian, Alberto Bernacchia

TreeMAML

Many NLP models rely on training in one high-resource language, and they cannot be directly used to make predictions for other languages at inference time. Most of the languages of the world are under-resourced and rely on Machine Translation (MT) to English to make use of Language Models. However, having an MT system in every direction is costly and not the best solution for every NLP task. We propose the use of meta-learning to solve this issue. Our algorithm, TreeMAML, extends a meta-learning model, MAML, by exploiting hierarchical languages relationships.

Methodology

MAML [1] fig.(a), adapts the model to each task with a few gradient steps. In our method, TreeMAML, this adaptation follows the hierarchical tree structure:

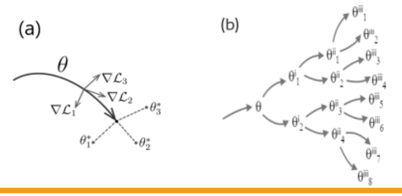
- In each step down the tree, gradients are pooled across language clusters, Algorithm 1 & fig.(b).
- Algorithm 2 is a non-binary modification of the OTD clustering [2], that generates the language tree without previous knowledge of the structure, allowing us to use implicit relationships between the languages.

Algorithm 1

Algorithm 1 TreeMAML

Require: distribution over tasks $p(\tau)$; distribution over data for each task $p(\mathcal{D}|\tau)$;
Require: number of inner steps K; number of training tasks m; learning rates α, β;
Require: number of clusters C_k for each step k; loss function $\mathcal{L}_\tau(\omega, \mathcal{D})$ for each task

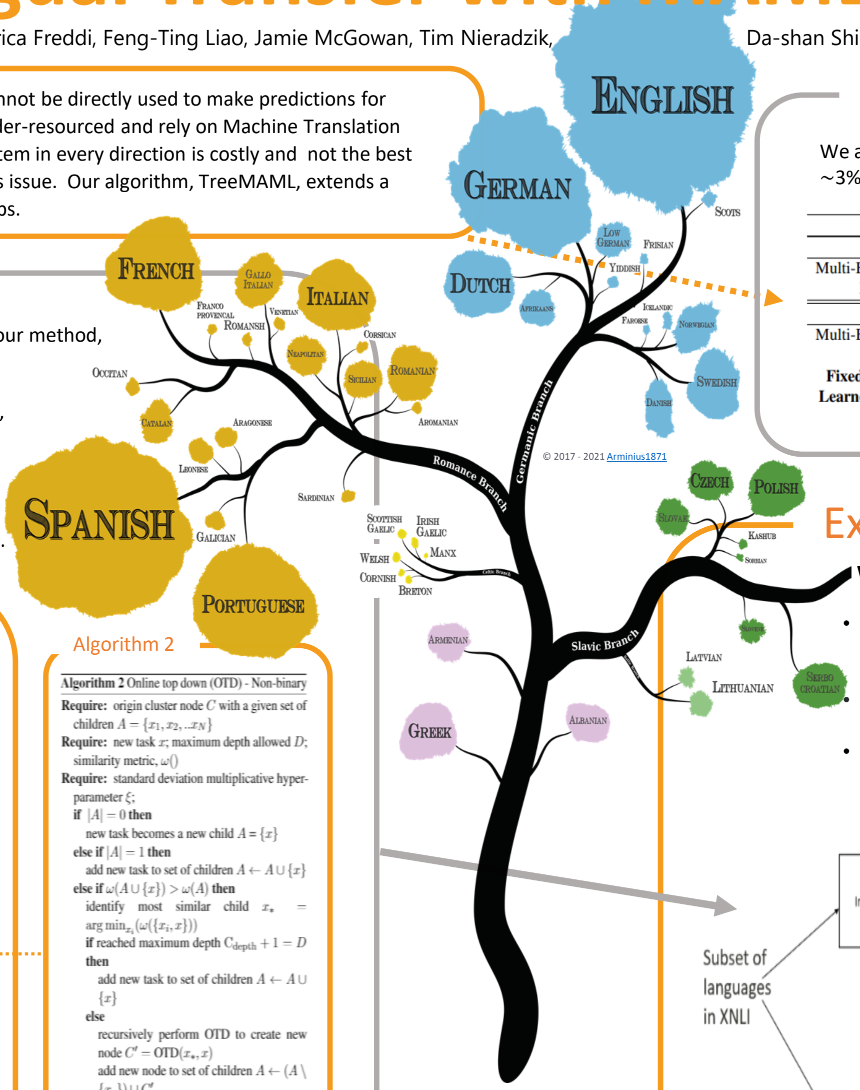
randomly initialize ω
while not done **do**
  sample batch of $i = 1 : m$ tasks $\{\tau_i\} \sim p(\tau)$
  for all tasks $i = 1 : m$ initialize a single cluster $c_i = 1$
  initialize $\theta_{1,0} = \omega$
  for steps $k = 1 : K$ **do**
    for tasks $i = 1 : m$ **do**
      sample batch of $j = 1 : n_v$ data points $\{\mathcal{D}_{ij}\} \sim p(\mathcal{D}|\tau_i)$
      evaluate gradient $g_{ik} = \frac{1}{n_t} \sum_{j=1}^{n_t} \nabla \mathcal{L}_{\tau_i}(\theta_{c_i,k-1}; \mathcal{D}_{ij})$
    end for
    regroup tasks into C_k clusters $\mathcal{T}_c = \{i : c_i = c\}$
    according to similarity of $\{g_{ik}\}$ and parent clusters $\{p_c\}$
    update $\theta_{c,k} = \theta_{p_c,k-1} - \frac{\alpha}{|\mathcal{T}_c|} \sum_{i \in \mathcal{T}_c} g_{ik}$ for all clusters $c = 1 : C_k$
  end for
  update $\omega \leftarrow \omega - \beta \frac{1}{m n_v} \sum_{i=1}^{m} \sum_{j=1}^{n_v} \nabla_\omega \mathcal{L}_{\tau_i} (\theta_{c_i,K}(\omega); \mathcal{D}_{ij})$
end while



Algorithm 2

Algorithm 2 Online top down (OTD) - Non-binary

Require: origin cluster node C with a given set of children $A = \{x_1, x_2, ..x_N\}$
Require: new task x; maximum depth allowed D; similarity metric, $\omega()$
Require: standard deviation multiplicative hyper-parameter ξ;
if $|A| = 0$ **then**
  new task becomes a new child $A = \{x\}$
else if $|A| = 1$ **then**
  add new task to set of children $A \leftarrow A \cup \{x\}$
else if $\omega(A \cup \{x\}) > \omega(A)$ **then**
  identify most similar child $x_* = \arg\min_{x_i}(\omega(\{x_i, x\}))$
  if reached maximum depth $C_{depth} + 1 = D$ **then**
    add new task to set of children $A \leftarrow A \cup \{x\}$
  else
    recursively perform OTD to create new node $C' = OTD(x_*, x)$
    add new node to set of children $A \leftarrow (A \setminus \{x_*\}) \cup C'$
  end if
else if $\omega(A \cup \{x\}) < \omega(A) - \xi\sigma_T$ **then**
  current node and new task become children to new cluster $A \leftarrow \{C, x\}$
else
  add new task to set of children $A \leftarrow A \cup \{x\}$
end if



References

[1] C. Finn, P. Abbeel, and S. Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv:1703.03400.
[2] A. Menon, A. Rajagopalan, B. Sumengen, G. Citovsky, Q. Cao, and S. Kumar. 2019. Online Hierarchical Clustering Approximations. arXiv:1909.09667
[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.
[4] A. Conneau, G. Lample, R. Rinott, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. arXiv:1809.05053
[5] F. Nooralahzadeh, G. Bekoulis, J. Bjerva, and I. Augenstein. 2020. ZeroShot Cross-Lingual Transfer with Meta Learning. arXiv:2003.02739.

Cross-lingual NLI Results

We applied TreeMAML to the cross-lingual XNLI problem [4] and show an improvement in accuracy ~3% with respect to the state of the art obtained by XMAML [5].

	en	fr	es	de	el	bg	ru	vi	th	zh	hi	ur	avg
two languages (Nooralahzadeh et al., 2020)													
Multi-BERT (Baseline)	81.94	75.39	75.79	73.25	69.54	71.60	70.84	73.23	61.18	73.93	64.37	63.71	71.23
XMAML	82.71	75.97	76.51	74.07	70.66	72.77	72.12	73.87	**62.5**	74.85	65.75	64.59	72.20
all languages (ours)													
Multi-BERT (Baseline)	83.56	76.22	76.89	73.11	**72.89**	72.89	71.33	74.67	57.56	74.89	63.11	63.33	71.70
MAML	83.11	78.22	77.11	73.56	69.33	71.78	71.33	74.22	57.33	75.11	63.33	63.78	71.52
Fixed TreeMAML	**84.67**	**79.78**	78.22	76.89	72.00	**74.22**	73.33	74.44	59.56	**79.11**	**66.00**	**66.89**	**73.76**
Learned TreeMAML	84.22	77.33	**79.78**	**78.00**	71.56	73.78	**74.00**	**74.89**	59.78	76.44	65.11	65.56	73.37

Experiments

We adapt a high-resource language model, Multi-BERT [3], to a Few-Shot NLI task with these steps:

- We use the XNLI data set [4]. XNLI corpus is a crowd-sourced collection of *pairs for the MultiNLI corpus* with 10 different genres in 15 languages. The pairs are annotated with textual entailment.
- Each combination of a language and a genre is consider a task.
- We perform Few-Shot meta-learning using three shots for each task during meta-training.

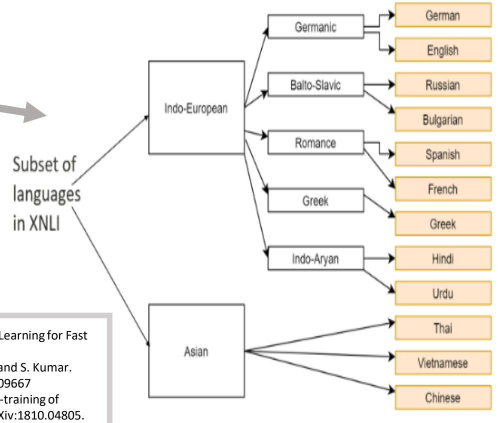


Fig 1. Simplified version of the phylogenetic Language tree used in Fixed TreeMAML.

- **We applied TreeMAML to fine tune the 12 layer of Multi-Bert.** We perform two experiments:

  Experiment 1 - Fixed TreeMAML: Assume that the language tree structure is known, and correspond to the one in Fig 1., and applying Algorithm 1.

  Experiment 2 - Learned TreeMAML: More general case where the relation among languages is not known. Algorithm 2 is used in each inner step of Algorithm 1 to cluster the gradients and learn the hierarchy between languages.

- We compare our methods with the baseline (Multi-Bert) and with the last state of the art results (XMAML, [5])