“Education is the most powerful weapon we can use to change the world.“

-- Nelson Mandela

“Education is the most powerful weapon we can use to change the BERT.“

-- Petr Sojka

Towards Domain Robustness of Neural Language Models

MUNI
FI

The Fifteenth Workshop on RASLAN
Michal Štefánik & Petr Sojka, MIR MU
stefanik.m@mail.muni.cz

Outline

1. Motivation
2. Related work
3. Proposals
    a. Impact of Objectives Curricula
    b. Softer Objectives
    c. Utilization of Generalization Measures
4. Domain and Task adaptation framework

Motivation

- Neural language models (LMs) perform outstandingly well on its own data domain
- Divergence from the iid (independent+identically distributed) samples end with unknown loss in quality

Related work

- HANS Dataset (T. McCoy et al., 2019) exposes the heuristics that SOTA NLI systems follow, reaching below-random performance
- PAWS Dataset (Y. Zhang et al., 2019) performs similar demonstration on paraphrase classification
- (Berard et al., 2019) shows that SOTA machine translation models trained o cannonical domains can be close-to useless on informal text (FOURSQUARE)
- (Belinkov et al., 2018) show that neural machine translation is vulnerable to minor typos, e.g. causing 50% drop of BLEU with typos in 20% of tokens
- (Nehyba & Stefanik, 2021) show that deep LMs might not be able to extrapolate over a set of even partially inconsistent annotation models
- ...

Proposals

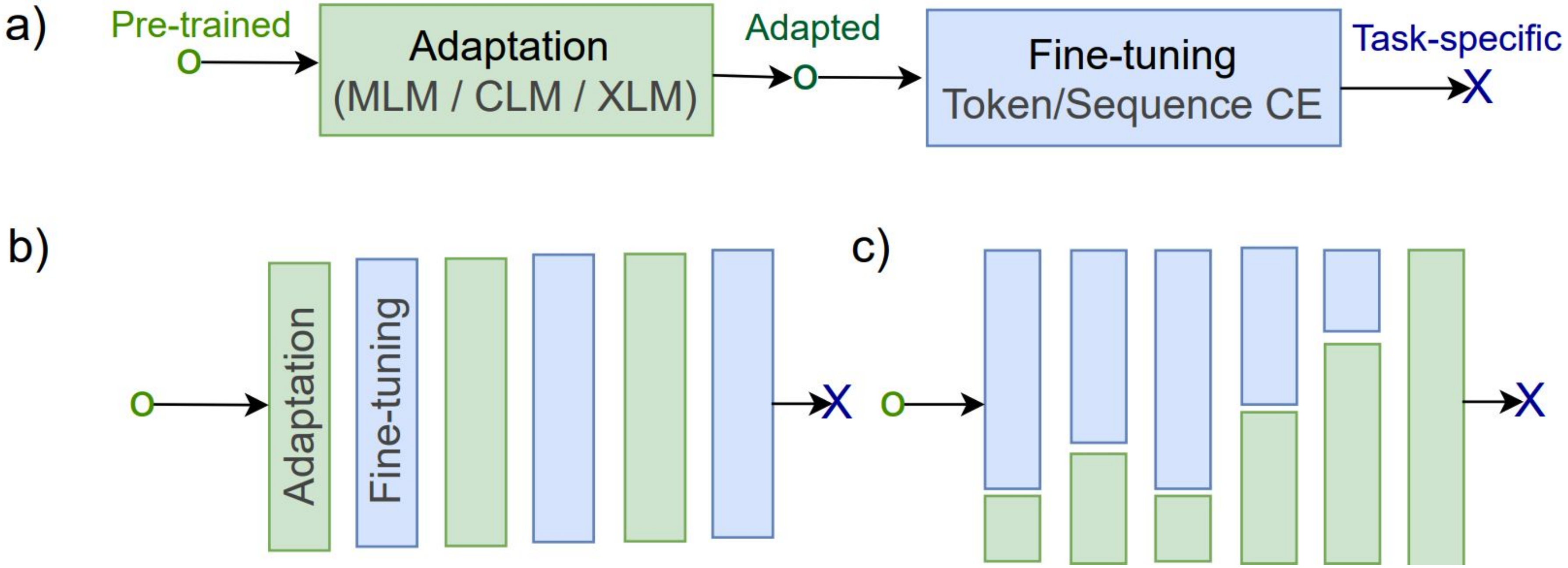
- Based also on the mentioned, our work proposes three **directions** of the future research
- Each of these is supplemented with a specific technical proposition

P1: Impact of Objectives Curricula

- Fine-tuning (adaptation) in low-resource settings in Machine Translation is prone to catastrophic forgetting or exposition bias (Ch. Wang & R. Sennrich 2020)
  - Importantly, these aspects are often not exposed on in-domain data set (D. Saunders, 2021)
- Can this be avoided by more elaborate sampling strategy?
- Previous work on "curriculum learning" did not bring significant gains (Y. Tsvetkov, 2016)

> “If we examine ourselves, we see that our faculties grow in such a manner that what goes before paves the way for what comes after.” J. A. Comenius

P1: Impact of Objectives Curricula

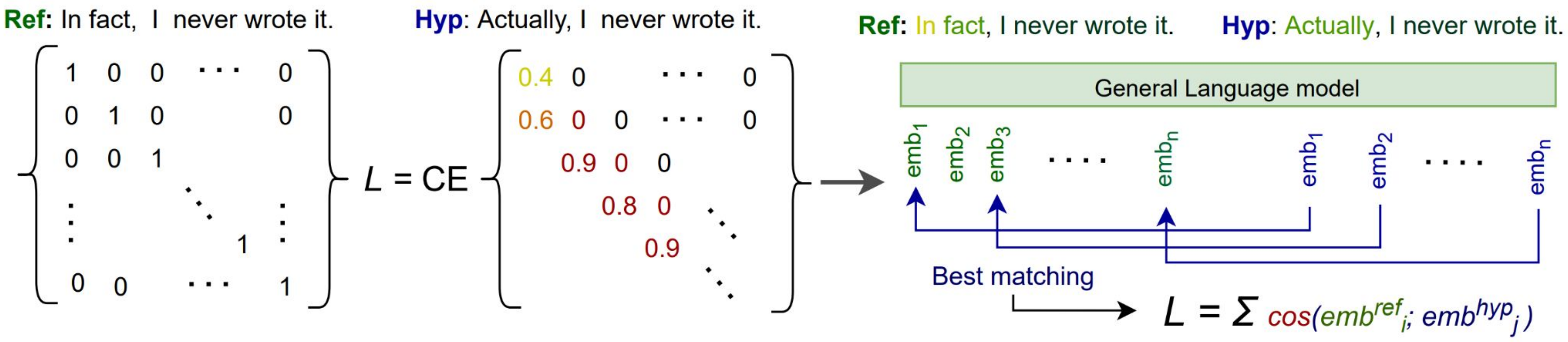


P2: Softer Objectives

- Training strategies that we use for high-level tasks expose the linguistic and logical phenomena in uncontrolled, sparse fashion
- We argue, that this sparsity and latency of the training objectives might be a cause of a vast data demands of some tasks
- Hence, we aim to expose the *semantics* of the task(s) more *explicitly*

> "The proper education of the young does not consist in stuffing their heads with a mass of words, sentences, and ideas dragged together out of various authors, but in opening up their understanding to the outer world, so that a living stream may flow from their own minds, just as leaves, flowers, and fruit spring from the bud on a tree." J. A. Comenius

P2: Softer Objectives



P3: Objectives Utilizing Generalization Measures

- We do not know, how to regularize our training routine so that it sufficiently generalize, but still reaches comparable performance
- But there is a good branch of research (e.g. Y. Jiang et al., 2019, GK. Dziugaite et al., 2020, Stefanik et al., 2021) relating some *descriptive* and *behavioural* properties of the models with *generalization*
    - *PAC-Bayesian, norm-based, gradient-based, spectral, behavioural (e.g. sharpness)*
- There are clues from image applications, that using these properties (e.g. Spectral Complexity (PL. Bartlett et al., 2019) or Sharpness (P. Forett et al., 2021) as *regularizers* can enhance out-of-distribution performance
- We think NLP calls for it as well!

What we demand is vigilance and attention
on the part of the master and the pupils." J. A. Comenius

P3: Objectives Utilizing Generalization Measures

$$\mathcal{L}(M) = (1-\alpha)\mathcal{L}_{Obj}(M) + \alpha\mathcal{L}_{Meas}(M) \tag{1}$$

> *To enhance model's distributional robustness, a task-specific training objective $\mathcal{L}_{Obj}$ can be additively complemented with a differentiable instance of the generalization measure $\mathcal{L}_{Meas}$.*

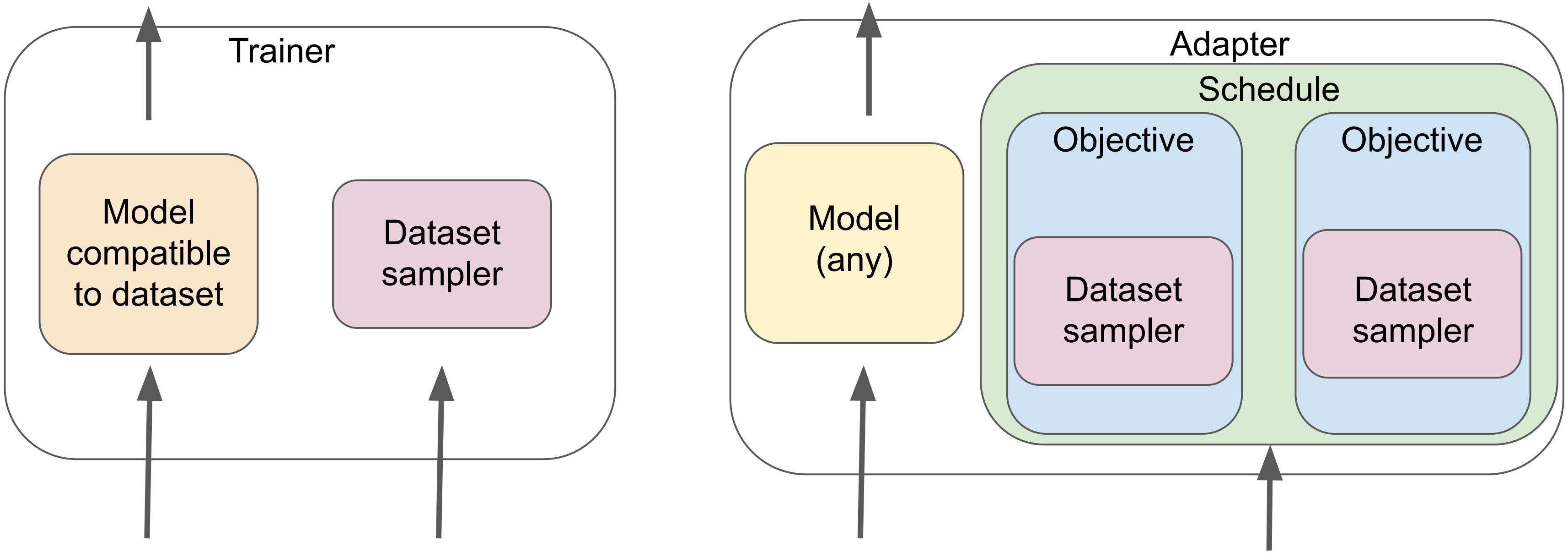
{P1, P2, P3} ⊆ {Task, Domain} Adaptation

- Separates a concept of *Objective* away from the neural network architecture
  - Association of Objective with LM head is clear, multi-headed models are fine
- Introduces a support for a *Schedule* of Objectives
- Does not care about the correspondence of Objective with the model, for which it was introduced
  - they're all transformers, anyway.
- Supports *any* PyTorch NN (why not RNN/LSTM) and *any* "language" (e.g. genome sequences)
  - Initialization of tokenizer model is deterministic, and can be replaced with *any* crafted tokenizer

{Task, Domain} Adaptation Framework

- Separates a concept of *Objective* away from the neural network architecture
    - Association of Objective with LM head is clear, multi-headed models are fine
- Introduces a support for a *Schedule* of Objectives
- Does not care about the correspondence of Objective with the model, for which it was introduced
    - they're all transformers, anyway.
- Supports *any* PyTorch NN (why not RNN/LSTM) and *any* "language" (e.g. genome sequences)
    - Initialization of tokenizer model is deterministic, and can be replaced with *any* crafted tokenizer

{Task, Domain} Adaptation Framework



Classic pipeline (Hugging Face Trainer)

Domain Adaptation pipeline

DA framework: adapted Named Entity Recognition

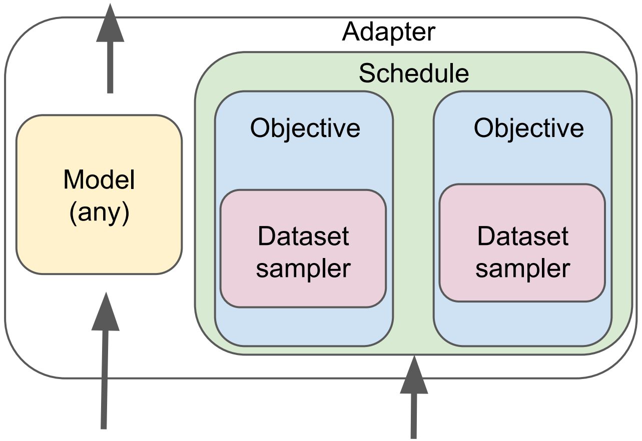
```
lang_module = LangModule("bert-base-multilingual-cased")

objectives = [
      MaskedLanguageModeling(lang_module,
                             texts_or_path="tests/mock_data/domain_unsup.txt",
                             batch_size=128),
      TokenClassification(lang_module,
                          texts_or_path="tests/mock_data/ner_texts_sup.txt",
                          labels_or_path="tests/mock_data/ner_texts_sup_labels.txt",
                          batch_size=8)
]

schedule = SequentialSchedule(objectives, training_arguments)

adapter = Adapter(lang_module, schedule, args=training_arguments)

adapter.train()
adapter.save_model("multihead_model")
```



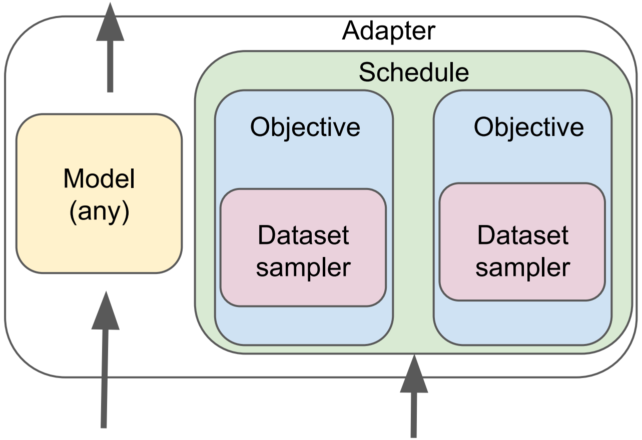
DA framework: adapted Machine Translation

```
lang_module = LangModule("Helsinki-NLP/opus-mt-en-cs")

objectives = [
      DenoisingObjective(lang_module,
                         texts_or_path="mock_data/domain_unsup.txt",
                         batch_size=16)
      CausalDecoderLanguageModelingSup(lang_module,
                                       texts_or_path="mock_data/seq2seq_sources.txt",
                                       labels_or_path=sup_translation_texts_tgt,
                                       source_lang_id="en",
                                       target_lang_id="cs",
                                       batch_size=8)
]
schedule = StridedSchedule(objectives, training_arguments)

adapter = Adapter(lang_module, schedule, args=training_arguments)

adapter.train()
adapter.save_model("model_with_LM_head")
```



{Task, Domain} Adaptation Framework

More examples:
https://github.com/authoranonymous321/DA

Thanks!

Questions / feedback / opinions welcome!

The Fifteenth Workshop on RASLAN
Michal Štefánik & Petr Sojka, MIR MU
stefanik.m@mail.muni.cz