Warsaw University of Technology

DISCIPLINE OF SCIENCE INFORMATION AND COMMUNICATION TECHNOLOGY

FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

Ph.D. Thesis

Mateusz Klimaszewski, M.Sc.

Multi-objective modularity in Natural Language Processing

Supervisor

Tomasz Gambin, Prof.

Additional supervisor

Piotr Andruszkiewicz, Ph.D.

WARSAW 2025

Acknowledgements

Multi-objective modularity in Natural Language Processing

The Natural Language Processing (NLP) field has seen rapid advancements in recent years through the continuous scaling of deep neural networks (DNNs). As a result, the emergence of Foundation Models significantly changed the landscape of NLP, where Pre-trained Language Models and Large Language Models (LLMs) demonstrate exceptional performance across numerous NLP tasks.

The ongoing DNNs scaling has led to increased popularity of Modular Deep Learning (MDL). The MDL methods present a cheaper alternative to full fine-tuning, offering parameter-efficient fine-tuning, which reduces the computational requirements of modifying a model. In particular, with flagship LLMs exceeding 100 billion parameters, even the inference of the state-of-the-art models requires substantial infrastructure, to say nothing of training costs. Moreover, the MDL modules can enhance Foundation Models with diverse capabilities, offering an opportunity to involve multiple modules simultaneously and combine them for specific downstream objectives.

This thesis focuses on Modular Deep Learning and studies multi-objective aspects of modular methods as a series of four publications. In the first three publications, we state research questions that examine a distinct problem in the existing aspects: multi-task, multi-domain and multilinguality. These works focus on one multi-objective at a time, making the rest of the variables constant (e.g. while exploring multi-task setup, we keep domain and language(s) unchanged). Each publication introduces a new system or method to address a shortcoming of the current methods in the corresponding aspect. Finally, in the last work, we formulate a new aspect, which we call multi-model. Here, we examine existing modular methods under new conditions and showcase the limitations of the current modular methods.

Keywords: Modular Deep Learning, Parameter-efficient fine-tuning, Natural Language Processing

Wielokryterialna modularność w przetwarzaniu języka naturalnego

Dziedzina przetwarzania języka naturalnego (NLP) odnotowała w ostatnich latach błyskawiczny postęp dzięki ciągłemu skalowaniu głębokich sieci neuronowych (DNN). W rezultacie, pojawienie się modeli podstawowych znacząco zmieniło krajobraz NLP, gdzie wstępnie wytrenowane modele językowe i duże modele językowe (LLM) osiągają wyjątkowe rezultaty w licznych zadaniach.

Trwający proces skalowania głębokich sieci neuronowych doprowadził do wzrostu popularności modularnego uczenia głębokiego (MDL). Metody MDL prezentują tańszą alternatywę do całkowitego dostrajania modeli poprzez parametrycznie efektywny odpowiednik, który redukuje wymagania obliczeniowe modyfikacji modelu. W szczególności, w przypadku sztandarowych modeli LLM, które swoim rozmiarem przekraczają 100 miliardów parametrów, nawet predykcja wymaga znacznej infrastruktury sprzętowej, pomijając całkowicie koszty treningu takiego modelu. Co więcej, moduły MDL mogą wzbogacać modele podstawowe dodając im różne umiejętności, często wykorzystując jednocześnie wiele z nich i łącząc je w zależności od potrzeb zadania docelowego.

Niniejsza praca skupia się na modularnym uczeniu głębokim i bada wielokryterialne aspekty metod modularnych w formie serii czterech publikacji. W pierwszych trzech z nich stawiane jest pytanie badawcze, które rozważa konkretny problem w istniejących aspektach: wielozadaniowości, wielodomenowości i wielojęzyczności. Owe prace skupiają się na jednym, konkretnym przypadku wielokryteriowości na raz, bez modyfikacji pozostałych (np. podczas eksploracji konfiguracji zadań wielozadaniowych zachowujemy niezmienioną domenę i język(i)). Każda z pierwszych trzech publikacji przedstawia nowy system lub metodę, która ma na celu usunięcie ograniczeń obecnych rozwiązań w danym aspekcie. W ostatniej, czwartej publikacji, został sformułowany nowy aspekt, nazwany wielomodelowym. Praca testuje istniejące metody modularne w nowych warunkach i ukazuje ograniczenia obecnych metod modularnych.

Słowa kluczowe: Modularne uczenie głębokie, Parametrycznie efektywne dostrajanie, Przetwarzania języka naturalnego

Contents

1. Introduction

The Natural Language Processing (NLP) field has seen rapid advancements in recent years. The continuous scaling of deep neural networks has led to the emergence of Foundation Models [71], significantly shaping the landscape of NLP. Recent advancements have introduced new architectures, such as the Attention mechanism [14] and Transformer [179], alongside semi-supervised training objectives like masked language modelling and next token prediction [48, 147]. These innovations, combined with unseen until this point scaling, have led to the creation of Pre-trained Language Models (PLMs) [130, 48, 108] and Large Language Models (LLMs) [148, 176, 30] that demonstrate exceptional performance across numerous tasks that include, among other, machine translation [44, 5], semantic analysis [80], and summarisation [200].

Considering the increasing model sizes, the cost of adapting such a Foundation Model to a new objective, whether for a different domain, task or language, plays a critical role. While the PLMs and LLMs offer exceptional zero-shot performance, some scenarios require the development of domain-specific models, such as those for medical, financial, or legal applications [151, 32, 192, 41]. Moreover, even regular fine-tuning to a specific task or language can become prohibitively expensive. With Large Language Models exceeding 100 billion parameters [160, 52, 1], even the inference of the state-of-the-art models requires substantial infrastructure, to say nothing of training costs.

Due to the ongoing scaling of deep neural networks, Modular Deep Learning (MDL) [134] has been gaining increasing attention. The MDL methods offer a cheaper, in terms of computational cost, alternative to full fine-tuning. The MDL's parameter-efficient fine-tuning (PEFT) techniques lower the cost by either modifying a subset of existing parameters [67, 23, 9] or training a new, but relatively small, set of parameters [83, 84, 107]. Therefore, thanks to PEFT, the Foundation Models can still serve as a base, general-purpose building block for downstream applications with relatively low modification costs.

The ongoing scaling, a prime force driving MDL's expansion, represents just one aspect of modularity's appeal. Modularity, at its core, seeks to encapsulate a specific responsibility into a dedicated module [83, 84]. Due to their clearly defined responsibilities (whether task-specific, language-specific, or domain-specific), the

modules can be independently evaluated and analysed based on their downstream application. Through their specialised compute function, modules can enhance Foundation Models with diverse capabilities, additionally offering an opportunity to involve multiple modules simultaneously and combine them (e.g. multi-domain setup) to address a specific downstream objective [141, 31, 122].

In the thesis, we investigate various challenges that modularity can address; rather than concentrating on an isolated problem such as a singular task, we examine configurations involving concurrent multiple objectives, specifically:

- multi-task
- multi-domain
- multilinguality
- multi-model.

Each listed aspect has specific use cases and features. The first aspect targets multi-task setups where an NLP system aims to solve multiple different tasks. Examples include models that extract jointly entities and relations [54, 55] or hierarchical models that handle tagging, parsing, relatedness, and entailment tasks at the same time [74]. The following multi-domain objective deals with cases where data comes from different domains (while keeping the task constant). For example, in machine translation, a source sentence might belong to medical, legal, or news categories (or any other), and we want to build systems that can be robust to a change of domain or perform well on a specified set of domains [49]. The multilingual aspect applies modularity to problems with fixed tasks and domains, but where the language is the changing factor. This objective, by definition, appears in Multilingual PLMs [42, 78, 199, 26] and LLMs [160, 118, 116, 64], but also in multilingual models for machine translation [44] or Named Entity Recognition [172]. At last, we challenge modularity against a case where the Foundation Model is a changing factor, formulating and evaluating the aspect [90] as one of the contributions of this thesis.

1.1. Research Questions

In the thesis, we focus on the multi-objective aspects of modularity, tackling four main research questions (RQs):

1. How can we build a multi-task NLP system where the end user can select relevant downstream tasks?
2. Can a routing function benefit from an external multi-domain teacher model?
3. Can we leverage knowledge of multiple language-specific modules without increasing the inference cost?

4. Can we reuse the pre-trained modules from different Foundation Models? What are the limitations of current modular methods when transferring between models?

In the following chapters, we address the outlined research questions, which are divided into four distinct multi-objective modularity aspects. Each research question explores an open research problem of a specific aspect and focuses on its multi-objective while keeping the rest constant, as presented in Table 1.1.1.

Table 1.1.1. The constant (C) and changing (X) modularity objectives in the research questions and corresponding publications.

Modularity objective	Task	Domain	Language	Foundation Model
RQ 1 [P1]	X	C	C	C
RQ 2 [P2]	C	X	C	C
RQ 3 [P3]	C	C	X	C
RQ 4 [P4]	C	C	C	X

1.2. Thesis Contribution

The thesis structure is as follows. Chapter 2 introduces the Modular Deep Learning background and relevant literature, positioning our work within the broader contexts of Natural Language Processing and Modular Deep Learning. Chapters 3-6 contain the core contribution of this thesis, presented as a series of four publications described below.

1.2.1. An NLP system with user-defined modularity for a multi-task setup.

Publication [P1]:
Mateusz Klimaszewski, Alina Wróblewska.
COMBO: State-of-the-Art Morphosyntactic Analysis.
The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021, System Demonstrations).
MNiSW list: 140 pts
Contribution: 60%
Description of the contribution: design, implementation, training & evaluation of the models, running experiments, results analysis, manuscript co-writing.

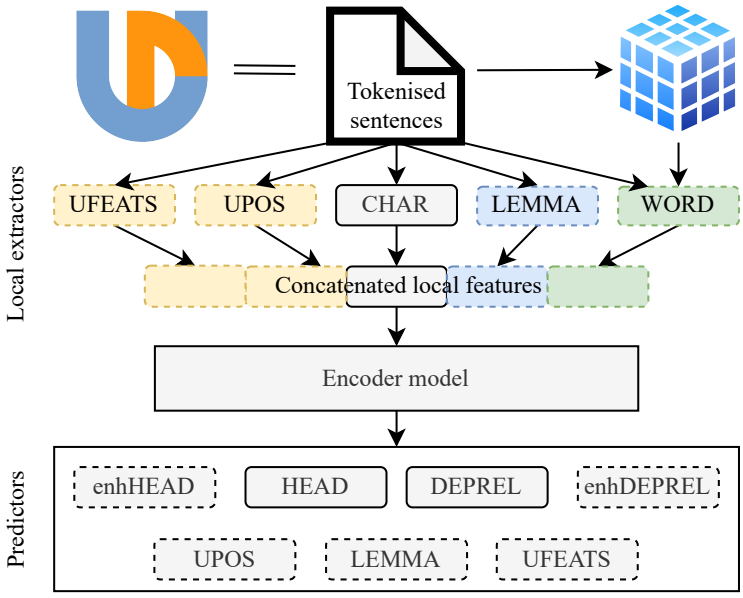


Figure 1.2.1. High-level view of the COMBO architecture. The modules marked in the solid line are required elements of the system, while the dashed lines present optional modules that the user can add or remove.

The first core contribution of the thesis [94], presented in Chapter 3, focuses on modularity that relates to the software engineering terminology (before the MDL term was forged) [174] and modularity that follows a multi-task learning paradigm [156]. The work creates an NLP system named COMBO for part-of-speech tagging, morphological analysis, lemmatisation and (enhanced) dependency parsing. The system is released as a standalone Python library with over 40 pre-trained models and a live demo.[1]

COMBO design presented in Figure 1.2.1 allows the end-user to define modularity on the input feature and prediction levels. Depending on available training data and its features, the user can enhance the raw text with, e.g. part-of-speech tags and train a COMBO model leveraging multiple inputs. On the other hand, when it comes to prediction, if the user is interested exclusively in dependency parsing, they can limit task-specific prediction modules to those responsible for the parsing task, reducing other prediction heads and system complexity and, therefore, improving efficiency.

The initial modular architecture was successfully extended, proving its design in our work [93] that built a new Enhanced Dependency Parsing Module ranked 4th at the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies [28]. Moreover, the CLARIN project[2] [140] extended COMBO with yet another module for

[1] `http://combo-demo.nlp.ipipan.waw.pl/`
[2] `https://clarin-pl.eu/`

the Named Entity Recognition (NER) task (work done without the PhD Candidate's participation).

The PhD Candidate, as the first author of this publication, played a crucial role in developing the proposed system. Specifically, the PhD Candidate took responsibility for designing and implementing the system, training and evaluating the models and conducting the experiments. Working collaboratively with the co-author, the PhD Candidate analysed the results and prepared the manuscript for publication.

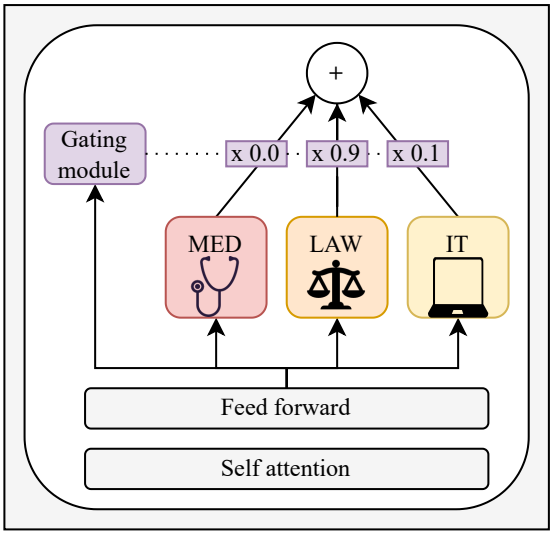
1.2.2. Knowledge sharing between domain-specific Adapters

Publication [P2]:
Mateusz Klimaszewski, Zeno Belligoli, Satendra Kumar, Emmanouil Stergiadis.
Gated Adapters for Multi-Domain Neural Machine Translation.
26th European Conference on Artificial Intelligence (ECAI 2023).
MNiSW list: 140 pts
Contribution: 65%
Description of the contribution: conceptualisation, implementation, training & evaluation of the method, results analysis, manuscript co-writing.

In the second work [92], presented in Chapter 4, we shift towards a multi-domain aspect, where the objective is to manage data coming from different domains under the same task in our study - machine translation. In particular, we focus on knowledge sharing between independently trained PEFT modules - adapters [83]. Given domain-specific adapters, we propose a mixture of expert-based architecture [162], dubbed Gated Adapters. We visualise the method in Figure 1.2.2.

Our method uses a router distilled from an external Pre-trained Language Model. The router, during inference, predicts soft labels for each domain and combines multiple adapters' outputs for each data sample instead of assigning a single adapter per sample. We base our research question in this study on the findings of Aharoni and Goldberg [4], who showcase that simply hard-labelling domains (i.e. choosing a single one out of many), given the source of origin, can be a limiting factor.

We evaluate the quality of the proposed architectures on two language pairs, English-Polish and English-Greek, with six domains each and measure quality using statistical and neural metrics. In particular, we show an improvement of over 5 COMET [153] points in the case of misclassified samples (i.e., ambiguous examples regarding their source domain). Moreover, we observe that the quality of the router distillation holds even during out-of-domain evaluation, i.e. when a new, unseen domain is taken into consideration.



Source: "Law act related to the usage of artificial intelligence..."

Figure 1.2.2. The proposed Gated Adapters overview. In the example sentence, the gates' weights lean towards law and IT Adapters and discard the medical one, as the source text concerns an AI-related law act.

As the first author, the PhD Candidate was responsible for conceptualising, implementing and evaluating the proposed method and also played a leading role in analysing the results. Working with co-authors, the PhD Candidate prepared the manuscript for publication. At last, the PhD Candidate had the honour of presenting the results of this work at the ECAI 2023 conference.

1.2.3. Model merging in a multilingual space

Publication [P3]:
Mateusz Klimaszewski, Piotr Andruszkiewicz, Alexandra Birch.
No Train but Gain: Language Arithmetic for training-free Language Adapters enhancement.
The 31st International Conference on Computational Linguistics (COLING 2025).
MNiSW list: 140 pts
Contribution: 70%
Description of the contribution: implementation, training & evaluation of the method, results analysis, manuscript co-writing

The third publication [91], Chapter 5, moves towards multilinguality and the landscape of model merging. The study investigates whether task arithmetic works under the challenge of the language beyond single-language tasks. Once again, we work with the Adapters as a PEFT method; however, in a more complex setup, i.e.

using a MAD-X [135] setup where modules are stacked, i.e. they process data via two consecutive modules: language- and task-specific.

Our findings show that model merging improves over MAD-X [136] results via merging two language Adapters, an operation we name language arithmetic (presented in Figure 1.2.3). Language arithmetic can improve existing Adapters, allowing the forging of a new module to obtain better zero-shot performance.

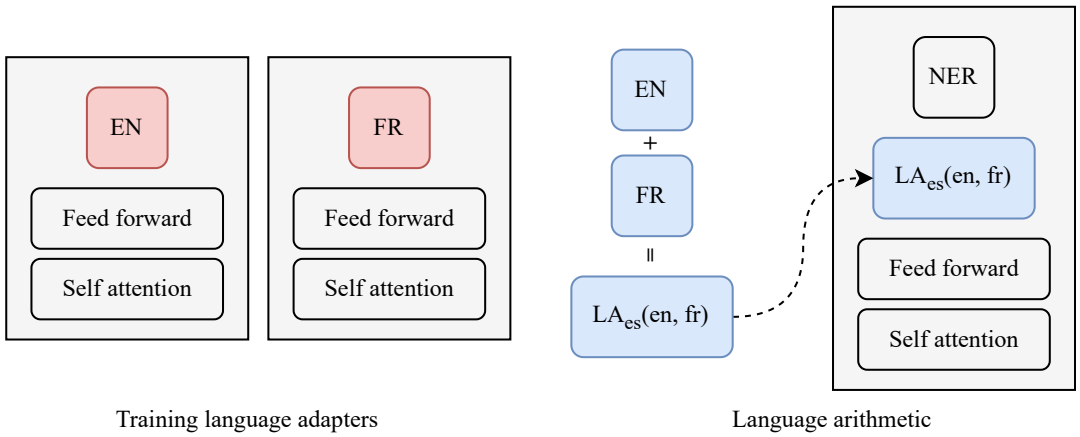


Figure 1.2.3. We propose language arithmetic – a method that combines pre-trained language adapters. The example presents a zero-shot setup where a language adapter for a target language (Spanish, es) was unavailable and is obtained by a model merging operation between English and French adapters.

Last, our analysis indicates that the newer methods that build upon the task arithmetic framework might not work for language arithmetic, as the internal representation exhibits different characteristics.

This work was led by the PhD Candidate, who conceptualised, implemented, trained and evaluated the proposed method. The PhD Candidate, working with co-authors, prepared the manuscript for publication and presented the work at the COLING 2025 conference. This study was supported by the Preludium grant from the National Science Centre, Poland, where the PhD Candidate is the Principal Investigator.

1.2.4. Modularity transfer between pre-trained models

Publication [P4]:
Mateusz Klimaszewski, Piotr Andruszkiewicz, Alexandra Birch.
Is Modularity Transferable? A Case Study through the Lens of Knowledge Distillation.
The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024).

MNiSW list: 140 pts
Contribution: 80%
Description of the contribution: conceptualisation, implementation, training & evaluation of the method, results analysis, manuscript co-writing

In the final work [90], Chapter 6, we formulate a property that would showcase the thorough modularity of the current MDL methods, i.e., the transferability of modules between Foundation Models. This study defines the property and attempts to unravel whether current MDL techniques support the property or can be easily adapted to support the transfer.

We take a closer look at a specific scenario as we examine a knowledge distillation setup [79] with PEFT presented in Figure 1.2.4. Our findings indicate that transferability, in the current MDL methods, is possible under the assumption that the teacher-student pair are a task-agnostic distillation of each other. In such a case, the student can retain partially external (teacher) knowledge on a downstream task. The evaluation was performed on two pairs of multilingual PLMs with three downstream tasks covering jointly a set of over 20 languages.

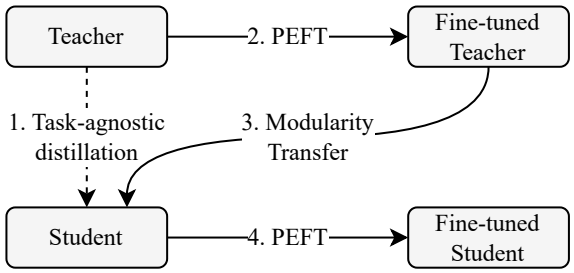


Figure 1.2.4. The schema of the transferable modularity experiment. We investigate setups where the teacher-student pair results from task-agnostic distillation (marked as the optional step 1) or are independently trained models.

This work was led by the PhD Candidate, who conceptualised, implemented, trained and evaluated the proposed method. The PhD Candidate, working with co-authors, prepared the manuscript for publication and presented the work at the LREC-COLING 2024 conference. It is important to mention that the article was prepared during the PhD Candidate's research visit to the University of Edinburgh, funded by the MOBILITY PW IX grant.

1.3. Other scientific contributions and achievements

The series of publications presented in the previous section is the core scientific contribution of this thesis. However, the scientific work has resulted in additional outcomes, i.e. research grants, open-sourced software, and other publications, which we briefly document in the following sections.

1.3.1. Own research grants

- PRELUDIUM 22, National Science Center, Poland
  Compositional modularity in Multilingual NMT
  10/2024 – 10/2026
  Budget: 139 031 PLN
- Mobility PW IX, Smart Education for Engineering Doctor NAWA STER, research visit grant
  Modularity in Neural Machine Translation
  09/2023 – 12/2023
  Budget: 39 000 PLN
- PlGrid ACK Cyfronet Helios computational grant
  Compositional modularity in Multilingual NLP
  04/2025 – 04/2026
  Budget: 120 744 GPU hours
- PCSS (Poznań Supercomputing and Networking Center) computational grant
  Knowledge Graphs application in Neural Machine Translation
  02/2022 – 02/2024
  Budget: 200 000 GPU hours

1.3.2. Participation in research grants

- HORIZON Research and Innovation Actions
  Unified Transcription and Translation for Extended Reality
  10/2022 – 09/2025
- INFOSTRATEG III, The National Centre for Research and Development, Poland
  Artificial Intelligence and Blockchain for product quality and safety control system
  08/2022 – 07/2023

- EuroHPC Extreme Scale Access computational grant
  European Large Language Model - EuroLLM
  05/2024 – 04/2025

1.3.3. Repositories and software packages

- COMBO system repository
  `https://gitlab.clarin-pl.eu/syntactic-tools/combo`
- COMBO PyPI package
  `https://pypi.clarin-pl.eu/simple/combo`
- COMBO system demo
  `https://combo-demo.nlp.ipipan.waw.pl`
- Gated adapters implementation
  `https://github.com/mklimasz/gated-adapters`
- Language arithmetic implementation
  `https://github.com/mklimasz/language-arithmetic`
- Transferable modularity implementation
  `https://github.com/mklimasz/transferable-modularity`

1.3.4. Other research publications

- **Mateusz Klimaszewski**, Alina Wróblewska
  COMBO: a new module for EUD parsing
  IWPT 2021
- Adam Dobrowolski, **Mateusz Klimaszewski**, Adam Myśliwy, Marcin Szymański, Jakub Kowalski, Kornelia Szypuła, Paweł Przewłocki, Paweł Przybysz
  Samsung R&D Institute Poland Participation in WMT 2022
  WMT 2022
- **Mateusz Klimaszewski**, Pinzhen Chen, Liane Guillou, Ioannis Papaioannou, Barry Haddow, Alexandra Birch
  AveniBench: Accessible and Versatile Evaluation of Finance Intelligence
  FinNLP @ COLING 2025
- Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M Guerreiro, Ricardo Rei, Duarte M Alves, José Pombal, Amin Farajian, Manuel Faysse, **Mateusz Klimaszewski**, Pierre Colombo, Barry Haddow, José GC de Souza, Alexandra Birch, André FT Martins
  EuroLLM: Multilingual Language Models for Europe
  EuroHPC user day 2025

- Laurie Burchell, Ona de Gibert, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajič, Erik Henriksson, **Mateusz Klimaszewski**, Ville Komulainen, Joona Kytöniemi, Veronika Laippala, Petter Mæhlum, Farrokh Mehryary, Vladislav Mikhailov, Nikita Moghe, Dayyán O'Brien, Stephan Oepen, Proyag Pal, Jousia Piha, Gema Ramírez-Sánchez, David Samuel, Pavel Stepachev, Dušan Variš, Tereza Vojtěchová, Jaume Zaragoza-Bernabeu
  An Expanded Massive Multilingual Dataset for High-Performance Language Technologies
  Accepted to ACL 2025
- Jacqueline Rowe, **Mateusz Klimaszewski**, Liane Guillou, Shannon Vallor, Alexandra Birch
  EuroGEST: Investigating gender stereotypes in multilingual language models
  Pre-print (in-review)
- Pedro Henrique Martins, João Alves, Patrick Fernandes, Nuno M. Guerreiro, Ricardo Rei, Amin Farajian, **Mateusz Klimaszewski**, Duarte M. Alves, José Pombal, Manuel Faysse, Pierre Colombo, François Yvon, Barry Haddow, José G. C. de Souza, Alexandra Birch, André F. T. Martins
  EuroLLM-9B: Technical Report
  Pre-print

2. Background

2.1. The Rise of Modular Deep Learning

Transfer learning (TL) [124, 157], which can be seen as the Modular Deep Learning ancestor [134], is a learning framework that addresses a limitation of many machine learning setups, where training and test examples are drawn from the same distribution [124]. In a real-world scenario, we often face the problem of a distribution shift [22] between training and test (also called inference or future) data unless we put effort and resources into collecting and annotating data specifically for our task or domain. The knowledge transfer between existing datasets or models, reducing the cost of labelling, is the main appeal of transfer learning.

Modular Deep Learning [134] emerged from the success of TL, where one of the core objectives of MDL is to prevent the TL limitation - negative transfer. *Negative transfer* [198], also known as *negative interference*, occurs when source data negatively impacts the target objective. For example, in a multi-task setting, it appears when a subset of tasks negatively impact each other and their training signal contradicts. A similar phenomenon has been denoted in multilingual models (termed *the curse of multilinguality*) [42]. As Pan and Yang (2012) [124] suggest, although most of the work assumes that there exists a relation (of some sort) between tasks, the assumption does not always hold; therefore, the methods might perform worse than a baseline without any TL. Furthermore, the authors identify negative transfer as the first of the challenges in their enumeration of open problems in TL research.

The TL methods that emerged during the deep learning revolution [25] have been a source of inspiration for the MDL methods. These architectural approaches align, to varying extents, with MDL properties, which we define in the following section. As an example of such architecture, in the multi-task learning paradigm, we have seen soft- and hard-parameter sharing setups [156] (Figure 2.1.1), where one takes advantage of a Foundation Model for multiple objectives, keeping task-specific modules (usually classifiers) separated. The former, hard-parameter sharing, setup

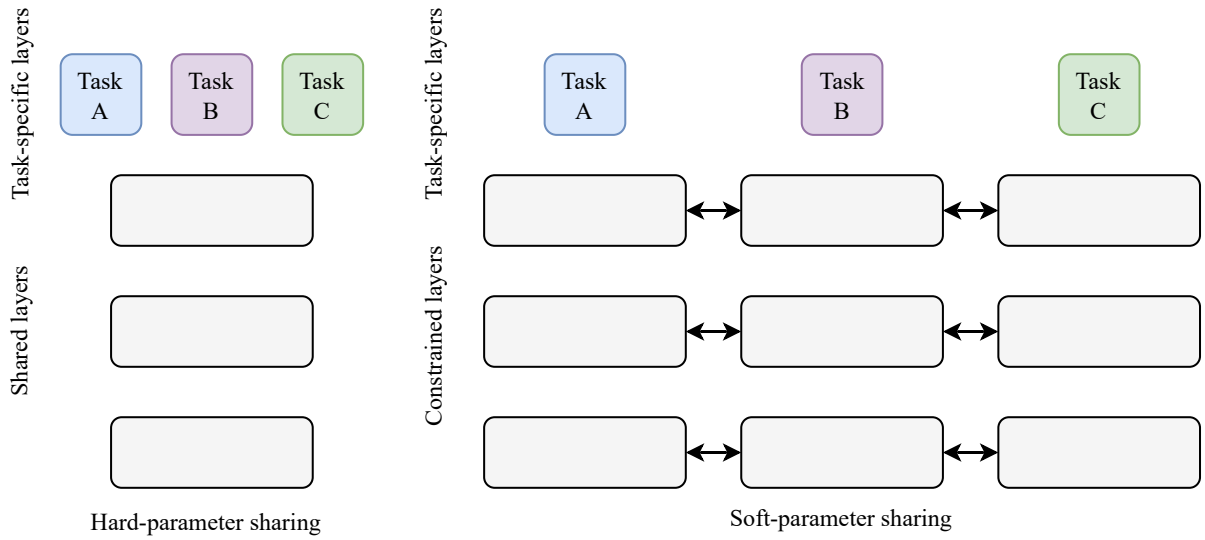


Figure 2.1.1. Example of modular transfer learning settings for a multi-task setup, where task-specific layers, usually feed-forward classifiers, are detached from a shared backbone. In the hard-sharing scenario, the task-specific modules are placed on a joint base and use the same weights. These parameters could be trained from scratch with the task layers or be a Foundation Model (e.g. a PLM or an LLM). In the soft-sharing scenario, there are multiple instances of the backbone, but all of them are constrained, e.g. with L2 loss [53]. Adapted from Ruder (2017) [156].

could be seen as a fully fledged modular system due to having explicit modular components.

It is important to mention that while the MDL can be seen as a direct successor of TL, the modularity concept is an idea that has been a part of the software engineering field [17, 27, 174] for a while. Additionally, recent findings showcase that "regular" neural networks, on their own, exhibit internal modularity, e.g. multilingual models contain language-specific subnetworks [58, 35].

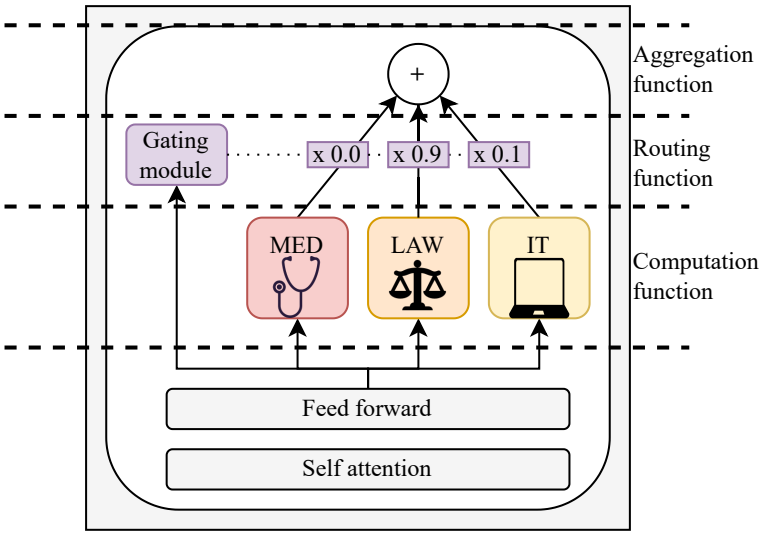
The [P1] publication is an example of the pre-MDL era system. It combines the modularity from the software engineering field (in the form of modules implemented to handle user-defined sets of features/targets) and hard-parameter sharing from the TL methods.

2.2. Modular Deep Learning

Modular Deep Learning proposes constructing neural networks with explicit modular components that can be independently updated without affecting the rest of the parameters. According to Pfeiffer et al. (2023) [134], MDL architecture comprises three principal components:

1. *computation function*,
2. *routing function*,
3. *aggregation function*.

The *computation function* determines how a module integrates into the existing model architecture. We examine the predominant architectures in Section 2.2.1. The *routing* and *aggregation functions* control information flow within the model - the former determines which modules to activate (and which to bypass), while the latter establishes how outputs from multiple modules are combined. In Figure 2.2.1, we revise Figure 1.2.2 from the [P2] publication , annotating the mentioned components.



Source: "Law act related to the usage of artificial intelligence..."

Figure 2.2.1. Annotated the three main functions of the Modular Deep Learning architecture on the example of Gated Adapters [92].

The proposed MDL functions bring distinct benefits. The computation functions are usually[1] isolated from a base model and specialised towards a specific objective. This separation mitigates negative interference, addressing one of TL's principal limitations. Furthermore, compute functions generally constitute only a small fraction of the model's total parameters [19], enhancing computational efficiency. Additionally, their specialisation toward specific objectives facilitates their combination (via routing and aggregation functions) to access novel capabilities [141, 31, 122]. Consequently, MDL is characterised by three fundamental properties [134]:

- positive transfer – to counteract TL's negative interference,
- parameter efficiency – allowing to modify even Foundation Models,

[1] We denote the outliers in Section 2.2.1.

- compositionality – to use modules as building blocks that enhance existing or new capabilities.

In the following sections, we provide additional background for two specific groups of MDL methods that relate to the core contributions of this thesis.

2.2.1. Parameter-efficient fine-tuning

The ongoing scaling of deep learning models increased the computational cost of fine-tuning (also called full fine-tuning) a model. Particularly with Large Language Models exceeding 100 billion parameters, training necessitates not merely multiple GPUs but distributed computing across multiple nodes. Moreover, the size of models increases the storage expenses. Considering multiple downstream tasks, maintaining separate fine-tuned versions for each task generates substantial costs. Parameter-efficient fine-tuning (PEFT) [133, 145, 73] emerged as a response to these mounting costs, offering a balanced compromise between computational expense and performance quality, and has played a foundational role in MDL.

Adapters, also referred to as bottleneck adapters, were introduced by Houlsby et al. (2019)[2] [83] and represent the earliest approach in the parameter-efficient fine-tuning family. Adapters are compact, small neural network layers strategically inserted within pre-trained models. They typically consist of down-projection and up-projection layers with a non-linearity in between. While the original model parameters remain frozen, only these lightweight adapter parameters are updated during training, significantly reducing the computational burden. Although several variants have been proposed that differ in the positioning within a Transfer layer and information flow (i.e. points of connection to the modified model) [83, 19, 132, 201, 76], the core concept remains unchanged. The downside of adapters is the increased computational overhead during inference – the modules are injected between existing layers, increasing the processing time of the adapted model. To avoid that limitation, Low Rank Adaptation (LoRA) [84] was proposed that yields a similar architecture as bottleneck adapter while (i) replacing sequential computation with parallel processing, (ii) removing non-linearity and approximating a single layer (i.e. a single weights' matrix rather than a complex layer as a Transformer layer). These modifications allow the LoRA's weights to be merged into the existing model during inference, and overall computational inference complexity, pre- and post-fine-tuning, stays the same. We present and compare these two techniques in Figure 2.2.2.

[2] We denote Rebuffi et al.'s (2017) [152] work that introduced the adapter concept and can be seen as the base inspiration for current bottleneck adapters.

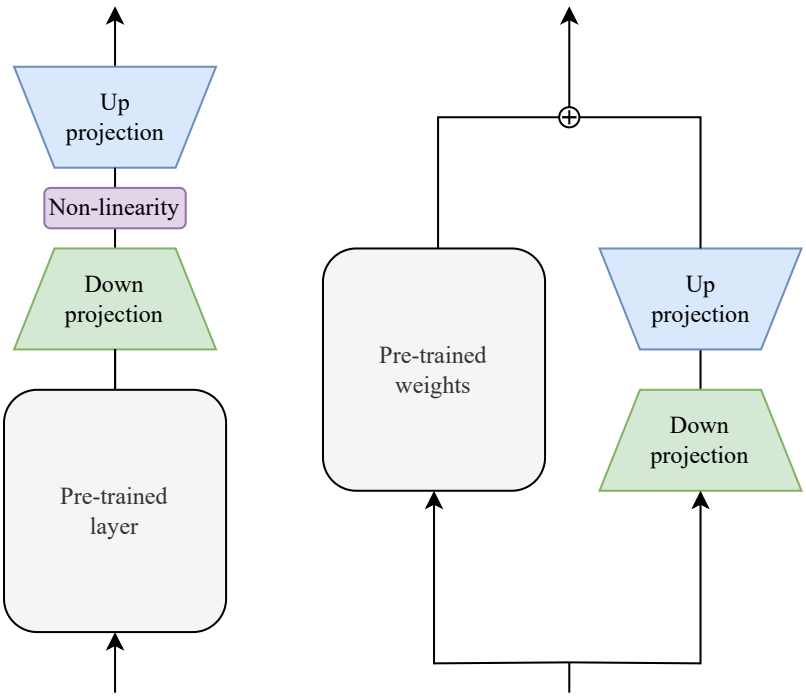


Figure 2.2.2. Schemas of flagship PEFT methods: bottleneck adapters (sequential, left) and LoRA (parallel, right). For clarity we omit residual connection. Although the methods increase complexity by roughly the same cost during training, LoRA's weight can be added to the approximated pre-trained weights after training [84], reducing the final cost of using the model. In contrast, bottleneck adapters must be applied (after the pre-trained layer) during inference.

Other methods that also represent the PEFT category are: subnetwork fine-tuning, e.g. bias weights fine-tuning - BitFit [23, 101] or sparse fine-tuning [8, 9], prefix tuning [104], prompt tuning [103] and few-shot learning (IA3) [107]. As mentioned in the previous section, most of these methods are isolated from the base model except for subnetwork fine-tuning. Subnetwork fine-tuning, while conceptually isolated (as a mask), in practice, disallows or makes it more complex [8] to use routing and aggregation functions as it trains a subset of existing parameters.

Our works [P2], [P3] and [P4] take advantage of outlined compute functions and apply them to the multi-domain, multilingual and multi-model cases. We train every single module to encode a distinct capability that depends on a downstream objective, e.g. in the multi-domain case [P2], each adapter represents a domain (law or IT or religion, etc.) for the machine translation system. Particularly, in the publications [P2] and [P3], we use bottleneck adapters to propose aggregation and routing functions, while in the [P4] work, we evaluate the proposed multi-model aspect using both adapter and LoRA modules.

2.2.2. Model merging

Model merging [191] is a set of techniques that allow combining or *merging* several existing models to obtain a new one with either novel / improved abilities or removed unwanted ones. Conceptually, these methods find their predecessors in model ensembling, e.g. averaging a few last training checkpoints of a model [179]. Task arithmetic [86] has been one of the foundational works in this field, and following Ilharco et al. (2023) [86] taxonomy, we distinguish three main applications of model merging:

- forgetting via negation,
- learning via addition,
- task analogies.

As in our work, [P3], we focus on the second application, learning via addition; we limit the introduction exclusively to this use case.

Given a pre-trained model (e.g. a Foundation Model) and its initial weights denoted as θ_{pre}, and a fine-tuned version of the model on a task t, θ^t_{ft}, we can obtain a so-called task vector τ, which is a parameter-wise difference between these two models:

$$\tau_t = \theta^t_{ft} - \theta_{pre} \tag{2.1}$$

Learning via addition allows one to take advantage of multiple task vectors and add multiple tasks to the initial model, creating a multi-task model (with a scaling parameter $\lambda \in [0,1]$) without training for this specific joint objective.

$$\theta_{multi-task} = \theta_{pre} + \lambda\tau_{t_1} + (1-\lambda)\tau_{t_2} \tag{2.2}$$

Task arithmetic allows us to forge a multi-task model from a separate, task-specific set of fine-tuned models, preserving high accuracy (although a shared pre-trained starting point is required, e.g. the same Large Language Model). While the naive method presented in Equation 2.2 performs a linear combination of the weights, many works have improved upon this baseline, considering, for example, parameter-sign interference [190] or singular value decomposition [62, 111]. These approaches have been especially successful in federated learning [33, 34], where we cannot access data to train a multi-task model jointly from scratch, and in continual learning [112, 113], where the data batches come sequentially and we, once again, cannot revisit the previous samples.

From the MDL perspective, model merging can be seen as a specific version of routing and aggregation functions where routing is done as a prior operation (i.e. the merging and its hyperparameters) while aggregation is an identity function

(after merging, the model is operating within its initial architecture; therefore, no aggregation is required). While the initial model merging works applied the techniques to fully fine-tuned models, Zhang et al. (2023) [197] evaluated the techniques under PEFT conditions, showing the same benefits.

In the publication [P3], we extend the typical use-case of the multi-task setup and explore the *learning via addition* capabilities of task arithmetic when it comes to the multilingual aspect – forging a new term: **language arithmetic**. We evaluate our approach as an extension to the MAD-X framework that leverages bottleneck adapters (details in Chapter 5).

2.3. Summary

This Chapter provides a background on the modular methods used in this thesis. We focus on a chosen subset of methods, positioning our work within the MDL field. For a more in-depth and complete overview of TL, MDL, PEFT and model merging, we refer the reader to existing surveys that summarise these fields well beyond Natural Language Processing [124, 134, 73, 191]. In the following Chapters, we present the core contributions of this thesis, each of which includes a contribution-oriented overview of related work.

3. COMBO: State-of-the-Art Morphosyntactic Analysis

Title	COMBO: State-of-the-Art Morphosyntactic Analysis
Authors	Mateusz Klimaszewski, Alina Wróblewska
Conference	The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021, System Demonstrations)
Year	2021

Abstract

We introduce COMBO – a fully neural NLP system for accurate part-of-speech tagging, morphological analysis, lemmatisation, and (enhanced) dependency parsing. It predicts categorical morphosyntactic features whilst also exposes their vector representations, extracted from hidden layers. COMBO is an easy to install Python package with automatically downloadable pre-trained models for over 40 languages. It maintains a balance between efficiency and quality. As it is an end-to-end system and its modules are jointly trained, its training is competitively fast. As its models are optimised for accuracy, they achieve often better prediction quality than SOTA. The COMBO library is available at: `https://gitlab.clarin-pl.eu/syntactic-tools/combo`.

3.1. Introduction

Natural language processing (NLP) has long recognised morphosyntactic features as necessary for solving advanced natural language understanding (NLU) tasks. An enormous impact of contextual language models on presumably all NLP tasks has slightly weakened the importance of morphosyntactic analysis. As morphosytnactic features are encoded to some extent in contextual word embeddings [e.g. 173, 105], doubts arise as to whether explicit morphosyntactic knowledge is still needed. For example, Glavaš and Vulić (2021) [63] have recently investigated an intermediate fine-tuning contextual language models on the dependency parsing task and suggested that this step does not significantly contribute to advance NLU models. Conversely, Warstadt et al. (2019) [186] reveal the powerlessness of contextual language models in encoding linguistic phenomena like negation. This is in line with our intuition about representing negation in Polish sentences (see Figure 3.1.1). It does not seem trivial to differentiate between the contradicting meanings of these sentences using contextual language models, as the word context is similar. The morphosyntactic features, e.g. parts of speech PART vs. INTJ, and dependency labels *advmod:neg* vs. *discourse:intj*, could be beneficial in determining correct reading.

In order to verify the influence of explicit morphosyntactic knowledge on NLU tasks, it is necessary to design a technique for injecting this knowledge into models or to build morphosyntax-aware representations. The first research direction was initiated by Glavaš and Vulić (2021) [63]. Our objective is to provide a tool for predicting high-quality morphosyntactic features and exposing their embeddings.

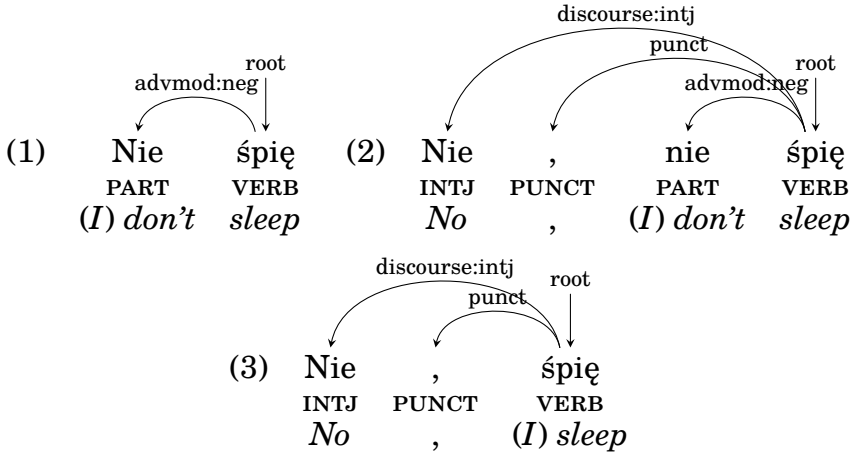


Figure 3.1.1. UD trees of Polish sentences: (1) and (2) mean a *non-sleeping* situation and (3) means *sleeping*.

These vectors can be directly combined with contextual word embeddings to build morphosyntactically informed word representations.

The emergence of publicly available NLP datasets, e.g. Universal Dependencies [196], stimulates the development of NLP systems. Some of them are optimised for efficiency, e.g. spaCy [82], and other for accuracy, e.g. UDPipe [167], the Stanford system [50], Stanza [146]. In this paper, we introduce COMBO, an open-source fully neural NLP system which is optimised for both training efficiency and prediction quality. Due to its end-to-end architecture, which is an innovation within morphosyntactic analysers, COMBO is faster in training than the SOTA pipeline-based systems, e.g. Stanza. As a result of applying modern NLP solutions (e.g. contextualised word embeddings), it qualitatively outperforms other systems.

COMBO analyses tokenised sentences and predicts morphosyntactic features of tokens (i.e. parts of speech, morphological features, and lemmata) and syntactic structures of sentences (i.e. dependency trees and enhanced dependency graphs). At the same time, its module, COMBO-vectoriser, extracts vector representations of the predicted features from hidden layers of individual predictors. COMBO user guide is in §3.4 and a live demo is available on the website `http://combo-demo.nlp.ipipan.waw.pl`.

Contributions 1) We implement COMBO (§3.2), a fully neural NLP system for part-of-speech tagging, morphological analysis, lemmatisation, and (enhanced) dependency parsing, together with COMBO-vectoriser for revealing vector representations of predicted categorical features. COMBO is implemented as a Python package which is easy to install and to integrate into a Python code. 2) We pre-train models for over 40 languages that can be automatically downloaded and directly used to

process new texts. 3) We evaluate COMBO and compare its performance with two state-of-the-art systems, spaCy and Stanza (§3.3).

3.2. COMBO Architecture

COMBO's architecture (see Figure 3.2.1) is based on the forerunner [158] implemented in the Keras framework. Apart from a new implementation in the PyTorch library [128], the novelties are the BERT-based encoder, the EUD prediction module, and COMBO-vectoriser extracting embeddings of UPOS and DEPREL from the last hidden layers of COMBO's tagging and dependency parsing module, respectively. This section provides an overview of COMBO's modules. Implementation details are in Appendix 3.6.1.

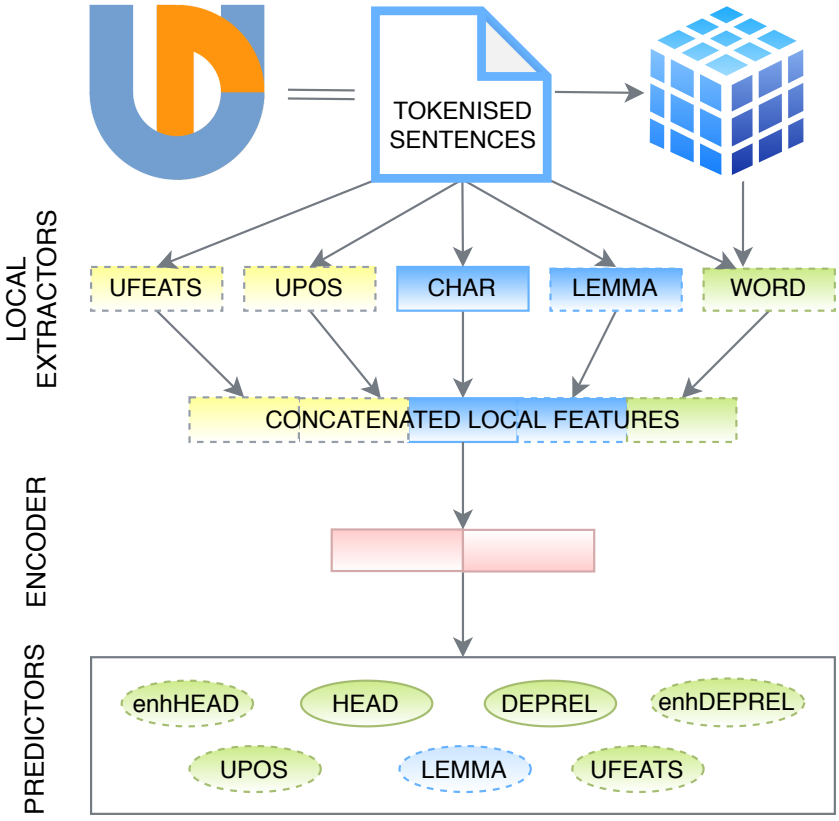


Figure 3.2.1. COMBO architecture. Explanations:
CNN FC System-trained biLSTM optional required

Local Feature Extractors Local feature extractors (see Figure 3.2.1) encode categorical features (i.e. words, parts of speech, morphological features, lemmata) into vectors. The feature bundle is configurable and limited by the requirements set for COMBO. For instance, if we train only a dependency parser, the following

features can be input to COMBO: internal character-based word embeddings (CHAR), pre-trained word embeddings (WORD), and embeddings of lemmata (LEMMA), parts of speech (UPOS) and morphological features (UFEATS). If we train a morphosyntactic analyser (i.e. tagger, lemmatiser and parser), internal word embeddings (CHAR) and pre-trained word embeddings (WORD), if available, are input to COMBO.

Words and lemmata are always encoded using character-based word embeddings (CHAR and LEMMA) estimated during system training with a dilated convolutional neural network (CNN) encoder [194, 168].

Additionally, words can be represented using pre-trained word embeddings (WORD), e.g. fastText [65], or BERT [48]. The use of pre-trained embeddings is an optional functionality of the system configuration. COMBO freezes pre-trained embeddings (i.e. no fine-tuning) and uses their transformations, i.e. embeddings are transformed by a single fully connected (FC) layer.

Part-of-speech and morphological embeddings (UPOS and UFEATS) are estimated during system training. Since more than one morphological feature can attribute a word, embeddings of all possible features are estimated and averaged to build a final morphological representation.

Global Feature Encoder The encoder uses concatenations of local feature embeddings. A sequence of these vectors representing all the words in a sentence is processed by a bidirectional LSTM [81, 66]. The network learns the context of each word and encodes its global (contextualised) features (see Figure 3.2.2). Global feature embeddings are input to the prediction modules.

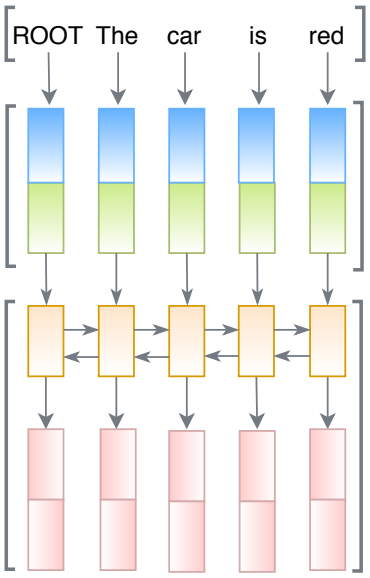


Figure 3.2.2. Estimation of global feature vectors.

Tagging Module The tagger takes global feature vectors as input and predicts a universal part of speech (UPOS), a language-specific tag (XPOS), and morphological features (UFEATS) for each word. The tagger consists of two linear layers followed by a softmax. Morphological features build a disordered set of category-value pairs (e.g. Number=Plur). Morphological feature prediction is thus implemented as several classification problems. The value of each morphological category is predicted with a FC network. Different parts of speech are assigned different sets of morphological categories (e.g. a noun can be attributed with grammatical gender, but not with grammatical tense). The set of possible values is thus extended with the NA (not applicable) symbol. It allows the model to learn that a particular category is not a property of a word.

Lemmatisation Module The lemmatiser uses an approach similar to character-based word embedding estimation. A character embedding is concatenated with the global feature vector and transformed by a linear layer. The lemmatiser takes a sequence of such character representations and transforms it using a dilated CNN. The softmax function over the result produces the sequence of probabilities over a character vocabulary to form a lemma.

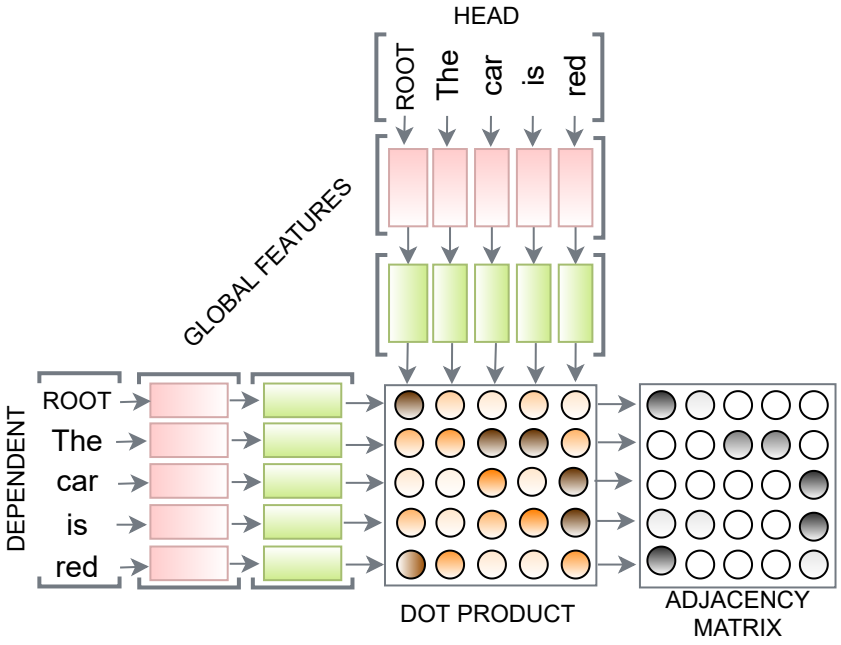


Figure 3.2.3. Prediction of dependency arcs.

Parsing Module Two single FC layers transform global feature vectors into head and dependent embeddings (see Figure 3.2.3). Based on these representations,

a dependency graph is defined as an adjacency matrix with columns and rows corresponding to heads and dependents, respectively. The adjacency matrix elements are dot products of all pairs of the head and dependent embeddings (the dot product determines the certainty of the edge between two words). The softmax function applied to each row of the matrix predicts the adjacent head-dependent pairs. This approach, however, does not guarantee that the resulting adjacency matrix is a properly built dependency tree. The Chu-Liu-Edmonds algorithm [39, 56] is thus applied in the last prediction step.

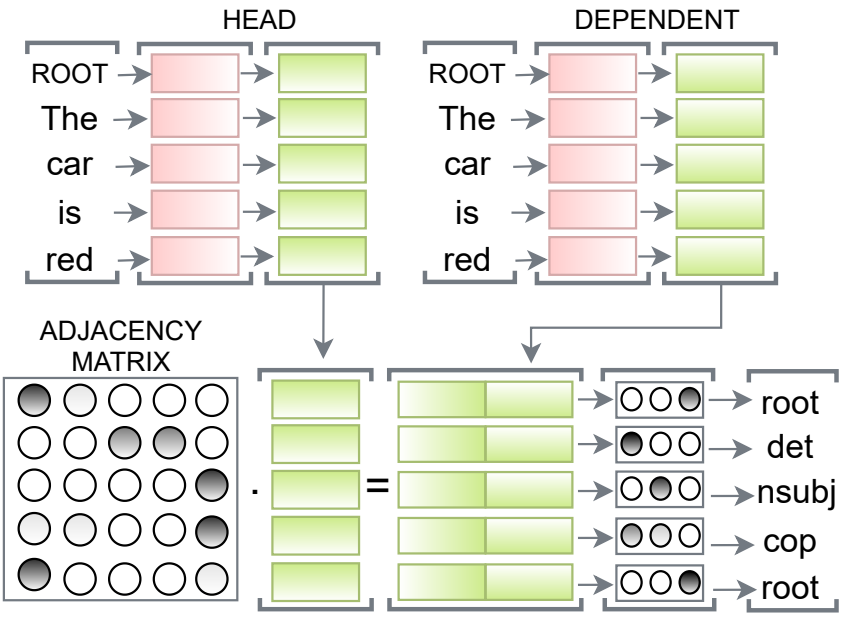


Figure 3.2.4. Prediction of grammatical functions.

The procedure of predicting words' grammatical functions (aka dependency labels) is shown in Figure 3.2.4. A dependent and its head are represented as vectors by two single FC layers. The dependent embedding is concatenated with the weighted average of (hypothetical) head embeddings. The weights are the values from the corresponding row of the adjacency matrix, estimated by the arc prediction module. Concatenated vector representations are then fed to a FC layer with the softmax activation function to predict dependency labels.

EUD Parsing Module Enhanced Universal Dependencies (EUD) are predicted similarly to dependency trees. The EUD parsing module is described in details in Klimaszewski and Wróblewska (2021) [93].

Table 3.2.1. Processing quality (F_1 scores) of spaCy, Stanza and COMBO on the selected UD treebanks (the language types are given in parentheses). The highest scores are marked in bold.

System	UPOS	XPOS	UFeat	Lemma	UAS	LAS	CLAS	MLAS	BLEX
English EWT (isolating)									
spaCy	93.79	93.10	94.89	NA	83.38	79.76	75.74	68.91	NA
Stanza	96.36	96.15	97.01	**98.18**	89.64	86.89	83.84	79.44	82.03
COMBO	95.60	95.21	96.60	97.43	88.56	85.58	82.35	76.56	79.78
$COMBO_{BERT}$	**96.57**	**96.44**	**97.24**	97.86	**91.76**	**89.28**	**86.83**	**81.71**	**84.38**
Arabic PADT (fusional)									
spaCy	90.27	82.15	82.70	NA	74.24	67.28	63.28	50.48	NA
Stanza	96.98	93.97	94.08	**95.26**	87.96	83.74	80.57	74.96	**76.80**
COMBO	96.71	93.72	93.83	93.54	87.06	82.70	79.46	73.25	73.64
$COMBO_{BERT}$	**97.04**	**94.83**	**95.05**	93.95	**89.21**	**85.09**	**82.36**	**76.82**	76.67
Polish PDB (fusional)									
spaCy	96.14	86.94	87.41	NA	86.73	82.06	79.00	65.42	NA
Stanza	98.47	94.20	94.42	97.43	93.15	90.84	88.73	81.98	85.75
COMBO	98.24	94.26	94.53	97.47	92.87	90.45	88.07	81.31	85.53
$COMBO_{BERT}$	**98.97**	**96.54**	**96.80**	**98.06**	**95.60**	**93.93**	**92.34**	**87.59**	**89.91**
Finnish TDT (agglutinative)									
spaCy	92.15	93.34	87.89	NA	80.06	74.75	71.52	61.95	NA
Stanza	97.24	97.96	95.58	**95.24**	89.57	87.14	85.52	80.52	**81.05**
COMBO	96.72	98.02	94.04	88.73	89.73	86.70	84.56	77.63	72.42
$COMBO_{BERT}$	**98.29**	**99.00**	**97.30**	89.48	**94.11**	**92.52**	**91.34**	**87.18**	77.84
Korean Kaist (agglutinative)									
spaCy	85.21	72.33	NA	NA	76.15	68.13	61.98	57.52	NA
Stanza	95.45	**86.31**	NA	**93.02**	88.42	86.39	83.97	80.64	**77.59**
COMBO	94.46	81.66	NA	89.16	87.31	85.12	82.70	78.38	72.79
$COMBO_{BERT}$	**95.89**	85.16	NA	89.95	**89.77**	**87.83**	**85.96**	**82.66**	75.89
Turkish IMST (agglutinative)									
spaCy	87.66	86.18	82.26	NA	60.43	51.32	47.74	37.28	NA
Stanza	**95.98**	**95.18**	**93.77**	96.73	74.14	67.52	64.03	58.13	61.91
COMBO	93.60	92.36	88.88	96.47	72.00	64.48	60.48	49.88	58.75
$COMBO_{BERT}$	95.14	94.27	93.56	**97.54**	**78.53**	**72.03**	**68.88**	**60.55**	**67.13**
Basque BDT (agglutinative with fusional verb morphology)									
spaCy	91.96	NA	86.67	NA	76.11	70.28	66.96	54.46	NA
Stanza	96.23	NA	93.09	**96.52**	86.19	82.76	81.30	73.56	78.27
COMBO	94.28	NA	90.44	95.47	84.64	80.44	78.82	67.33	74.95
$COMBO_{BERT}$	**96.26**	NA	**93.84**	96.38	**88.73**	**85.80**	**84.93**	**75.96**	**81.25**
Average scores									
spaCy	91.03	85.67	86.97	NA	76.73	70.51	66.60	56.57	NA
Stanza	96.67	93.96	94.66	**96.05**	87.01	83.61	81.14	75.60	77.63
COMBO	95.66	92.54	93.05	94.04	86.02	82.21	79.49	72.05	73.98
$COMBO_{BERT}$	**96.88**	**94.37**	**95.63**	94.75	**89.67**	**86.64**	**84.66**	**78.92**	**79.01**

3.3. COMBO Performance

Data COMBO is evaluated on treebanks from the Universal Dependencies repository [196], preserving the original splits into training, validation, and test sets.

The treebanks representing distinctive language types are summarised in Table 3.6.1 in Appendix 3.6.2.

By default, pre-trained 300-dimensional fastText embeddings [65] are used. We also test encoding data with pre-trained contextual word embeddings (the tested BERT models are listed in Table 3.6.2 in Appendix 3.6.2). The UD datasets provide gold-standard tokenisation. If BERT intra-tokeniser splits a word into sub-words, the last layer embeddings are averaged to obtain a single vector representation of this word.

Qualitative Evaluation Table 3.2.1 shows COMBO results of processing the selected UD treebanks.[1] COMBO is compared with Stanza [146] and spaCy.[2] The systems are evaluated with the standard metrics [195] F1, UAS (unlabelled attachment score), LAS (labelled attachment score), MLAS (morphology-aware LAS) and BLEX (bi-lexical dependency score).[3]

COMBO and Stanza undeniably outrun spaCy models. COMBO using non-contextualised word embeddings is outperformed by Stanza in many language scenarios. However, COMBO supported with BERT-like word embeddings beats all other solutions and is currently the SOTA system for morphosyntactic analysis.

Regarding lemmatisation, Stanza has an advantage over COMBO in most tested languages. This is probably due to the fact that Stanza lemmatiser is enhanced with a key-value dictionary, whilst COMBO lemmatiser is fully neural. It is not surprising that a dictionary helps in lemmatisation of isolating languages (English). However, the dictionary approach is also helpful for agglutinative languages (Finnish, Korean, Basque) and for Arabic, but not for Polish (fusional languages). Comparing COMBO models estimated with and without BERT embeddings, we note that BERT boost only slightly increases the quality of lemma prediction in the tested fusional and agglutinative languages.

Table 3.3.1. Training time of spaCy, Stanza and COMBO.

Treebank	spaCy	Stanza				COMBO	
		Tagger	Lemmatiser	Parser	Total	fastText	BERT
English EWT	00:22:34	02:08:51	02:12:17	02:29:13	06:50:21	01:26:55	1:54:11
Polish PDB	01:07:55	04:36:51	03:19:04	05:08:41	13:04:36	02:39:44	3:31:41

[1] Check the prediction quality for other languages at: `https://gitlab.clarin-pl.eu/syntactic-tools/combo/-/blob/master/docs/performance.md`.

[2] `https://spacy.io` We use the project template `https://github.com/explosion/projects/tree/v3/pipelines/tagger_parser_ud`. The lemmatiser is implemented as a standalone pipeline component in spaCy v3 and we do not test it.

[3] `http://universaldependencies.org/conll18/conll18_ud_eval.py` (CoNLL 2018 evaluation script).

For a complete insight into the prediction quality, we evaluate individual UPOS and UDEPREL predictions in English (the isolating language), Korean (agglutinative) and Polish (fusional). Result visualisations are in Appendix 3.6.3.

COMBO took part in IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies [28], where it ranked 4th.[4] In addition to ELAS and EULAS metrics, the third evaluation metric was LAS. COMBO ranked 2nd, achieving the average LAS of 87.84%. The score is even higher than the average LAS of 86.64% in Table 3.2.1, which is a kind of confirmation that our evaluation is representative, reliable, and fair.

Downstream Evaluation According to the results in Table 3.2.1, COMBO predicts high-quality dependency trees and parts of speech. We therefore conduct a preliminary evaluation of morphosyntactically informed word embeddings in the textual entailment task (aka natural language inference, NLI) in English [24] and Polish [189]. We compare the quality of entailment classifiers with two FC layers trained on max/mean-pooled BERT embeddings and sentence representations estimated by a network with two transformer layers which is given morphosyntactically informed word embeddings (i.e. BERT-based word embeddings concatenated with UPOS embeddings, DEPREL embeddings, and BERT-based embeddings of the head word). The morphosyntactically informed English NLI classifier achieves an accuracy of 78.84% and outperforms the max/mean-pooled classifiers by 20.77 pp and 5.44 pp, respectively. The Polish syntax-aware NLI classifier achieves an accuracy of 91.60% and outperforms the max/mean-pooled classifiers by 17.2 pp and 7.7 pp, respectively.

Efficiency Evaluation We also compare spaCy, Stanza and COMBO in terms of their efficiency, i.e. training and prediction speed.[5] According to the results (see Tables 3.3.1 and 3.3.2), spaCy is the SOTA system, and the other two are not even close to its processing time. Considering COMBO and Stanza, whose prediction quality is significantly better than spaCy, COMBO is 1.5 times slower (2 times slower with BERT) than Stanza in predicting, but it is definitely faster in training. The reason for large discrepancies in training times is the different architecture of these two systems. Stanza is a pipeline-based system, i.e. its modules are trained one after the other. COMBO is an end-to-end system, i.e. its modules are jointly trained and the training process is therefore faster.

[4] `https://universaldependencies.org/iwpt21/results.html`

[5] A single NVIDIA V100 card is used in all tests.

Table 3.3.2. Prediction time of Stanza and COMBO relative to spaCy (1×) on English and Polish test data.

Treebank	Stanza	COMBO	$\text{COMBO}_{\text{BERT}}$
English EWT	4.7×	6.8×	10.8×
Polish PDB	4.1×	5.8×	10.6×

3.4. Getting Started with COMBO

Prediction COMBO provides two main prediction modes: a Python library and a command-line interface (CLI). The Python package mode supports automated model download. The code snippet demonstrates downloading a pre-trained Polish model and processing a sentence:

```
from combo.predict import COMBO

nlp = COMBO.from_pretrained("polish")
sentence = nlp("Ala ma kota.")
print(sentence.tokens)
```

To download a model for another language, select its name from the list of pre-trained models.[6] The Python mode also supports acquisition of DEPREL or UPOS embeddings, for example:

```
sentence = nlp("Ala ma kota.")
chosen_token = sentence.tokens[1]
print(chosen_token.embeddings["upostag"])
```

In CLI mode, COMBO processes sentences using either a downloaded model or a model trained by yourself. CLI works on raw texts and on the CoNLL-U files (i.e. with tokenised sentences and even morphologically annotated tokens):

```
combo --mode predict \
      --model_path model.tar.gz \
      --input_file input.conllu \
      --output_file output.conllu
```

[6] The list of the pretrained COMBO models: https://gitlab.clarin-pl.eu/syntactic-tools/combo/-/blob/master/docs/models.md#pre-trained-models

Model Training COMBO CLI allows to train new models for any language. The only requirement is a training dataset in the CoNLL-U/CoNLL-X format. In the default setup, tokenised sentences are input and all possible predictors are trained:

```
combo --mode train \
      --training_data training.conllu \
      --validation_data valid.conllu
```

If we only train a dependency parser, the default setup should be changed with configuration flags: `--features` with a list of input features and `--targets` with a list of prediction targets.

3.5. Conclusion

We have presented COMBO, the SOTA system for morphosyntacic analysis, i.e. part-of-speech tagging, morphological analysis, lemmatisation, and (enhanced) dependency parsing. COMBO is a language-agnostic and format-independent system (i.e. it supports the CoNLL-U and CoNLL-X formats). Its implementation as a Python package allows effortless installation, and incorporation into any Python code or usage in the CLI mode. In the Python mode, COMBO supports automated download of pre-trained models for multiple languages and outputs not only categorical morphosyntactic features, but also their embeddings. In the CLI mode, pre-trained models can be manually downloaded or trained from scratch. The system training is fully configurable in respect of the range of input features and output predictions, and the method of encoding input data.
Last but not least, COMBO maintains a balance between efficiency and quality. Admittedly, it is not as fast as spaCy, but it is much more efficient than Stanza considering the training time. Tested on the selected UD treebanks, COMBO morphosyntactic models enhanced with BERT embeddings outperform spaCy and Stanza models.

3.6. Appendix

3.6.1. COMBO Implementation

COMBO is a Python package that uses the PyTorch [128] and AllenNLP [61] libraries. The COMBO models used in the evaluation presented in Section 3.3 are trained with the empirically set default parameters specified below. The training

parameters can be easily configured and adjusted to the specifics of an individual model.

Network Hyperparameters

Embeddings An internal character-based word embedding is calculated with three convolutional layers with 512, 256 and 64 filters with dilation rates equal to 1, 2 and 4. All filters have the kernel size of 3. The internal word embedding has a size of 64 dimensions. All external word embeddings are reduced to 100-dimensional vectors by a single FC layer. As only words are used as input features in the system evaluation, the local feature embedding is a concatenation of the 64-dimensional internal and 100-dimensional external word embedding. The global feature vectors are computed by two biLSTM layers with 512 hidden units.

Prediction modules The tagger uses a FC network with a hidden layer of the size 64 to predict UPOS and FC networks with 128-dimensional hidden layers to predict XPOS and UFEATS.
The lemmatiser uses three convolutional layers with 256 filters and dilation rates equal to 1, 2 and 4. All filters have the kernel size of 3. The fourth convolutional layer with the number of filters equal to the number of character instances in training data is used to predict the probability of each character. The final layer filters have the kernel size of 1. The 256-dimensional embeddings of input characters are concatenated with the global feature vectors reduced to 32 dimensions with a single FC layer.
The arc prediction module uses 512-dimensional head, and dependent embeddings and the labelling module uses 128-dimensional vectors.

COMBO-vectoriser currently outputs 64-dimensional UPOS and 128-dimensional DEPREL embeddings.

Activation function FC and CNN layers use hyperbolic tangent and rectified linear unit [119] activation functions, respectively.

Regularisation

Dropout technique for Variational RNNs [60] with 0.33 rate is applied to the local feature embeddings and on top of the stacked biLSTM estimating global feature embeddings. The same dropout, for output and recurrent values, is used in the context of each biLSTM layer. The FC layers use the standard dropout [164] with 0.25 rate. Moreover, the biLSTM and convolutional layers use L2 regularisation with the rate of 1×10^{-6}, and the trainable embeddings use L2 with the rate of 1×10^{-5}.

Training

The cross-entropy loss is used for all parts of the system. The final loss is the weighted sum of losses with the following weights for each task:

- 0.05 for predicting UPOS and LEMMA,
- 0.2 for predicting UFEATS and (enh)HEAD,
- 0.8 for predicting (enh)DEPREL.

The whole system is optimised with Adam [89] with the learning rate of 0.002 and $\beta_1 = \beta_2 = 0.9$. The model is trained for a maximum of 400 epochs, and the learning rate is reduced twice by the factor of two when the validation score reaches a plateau.

3.6.2. External Data Summary

Tables 3.6.1 and 3.6.2 list the UD dependency treebanks and BERT models used in the evaluation experiments presented in Section 3.3.

Table 3.6.1. The UD treebanks used in the evaluation experiments.

Language	Language Type	UD Treebank	#Words	#Trees	Reference
English	isolating	English-EWT	254,856	16,622	[163]
Arabic	fusional	Arabic-PADT	282,384	7,664	[72]
Polish	fusional	Polish-PDB	350,036	22,152	[188]
Finnish	agglutinative	Finnish-TDT	202,453	15,136	[75]
Korean	agglutinative	Korean-Kaist	350,090	27,363	[40]
Turkish	agglutinative	Turkish-IMST	57,859	5,635	[169]
Basque	agglutinative (fusional verb morphology)	Basque-BDT	121,443	8,993	[11]

Table 3.6.2. The BERT models used in the evaluation experiments.

Language	BERT model	Reference
Arabic	bert-base-arabertv2	[10]
Basque	berteus-base-cased	[2]
English	bert-base-cased	[48]
Finnish	bert-base-finnish-cased-v1	[180]
Korean	bert-kor-base	[87]
Polish	herbert-base-cased	[117]
Turkish	bert-base-turkish-cased	[161]

3.6.3. Evaluation of UPOS and UDEPREL

The comparison of the universal parts of speech predicted by the tested systems in English, Korean and Polish data is shown in the charts in Figures 3.6.1, 3.6.2 and 3.6.3, respectively. The comparison of the quality of the predicted universal dependency types in English, Korean and Polish data is presented in Figures 3.6.4, 3.6.5 and 3.6.6, respectively.

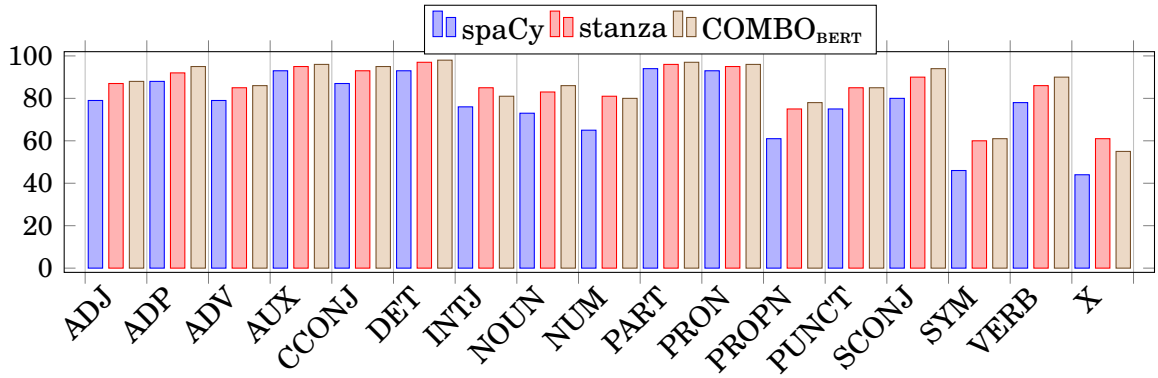


Figure 3.6.1. Evaluation of predicted universal parts of speech (UPOS) in the English test set ($F\text{-}_1$-scores).

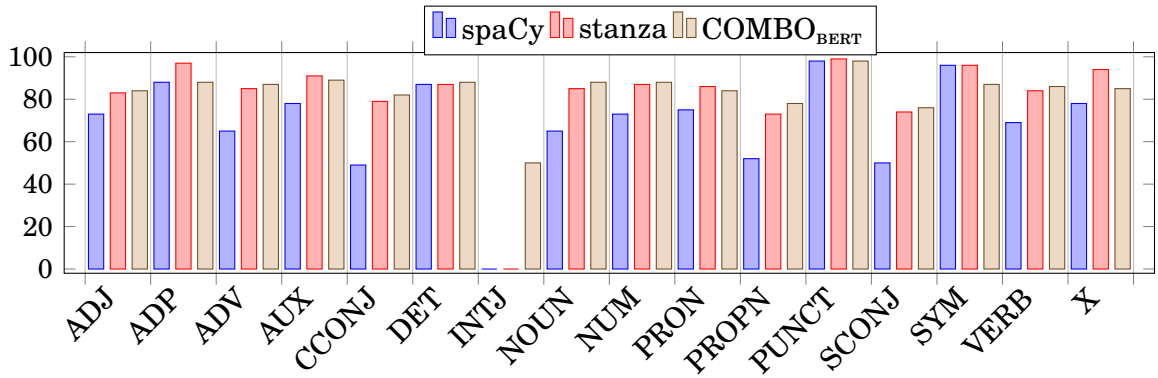


Figure 3.6.2. Evaluation of predicted universal parts of speech (UPOS) in the Korean test set ($F\text{-}_1$-scores).

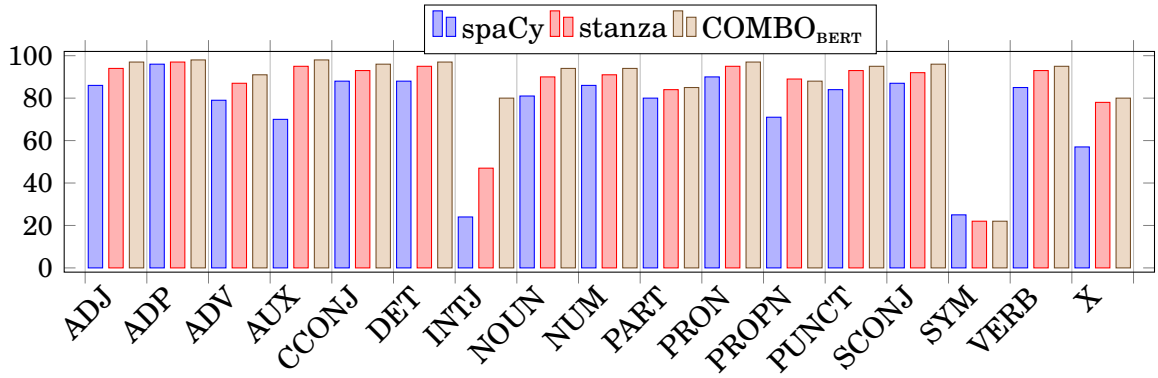


Figure 3.6.3. Evaluation of predicted universal parts of speech (UPOS) in the Polish test set ($F\text{-}_1$-scores).

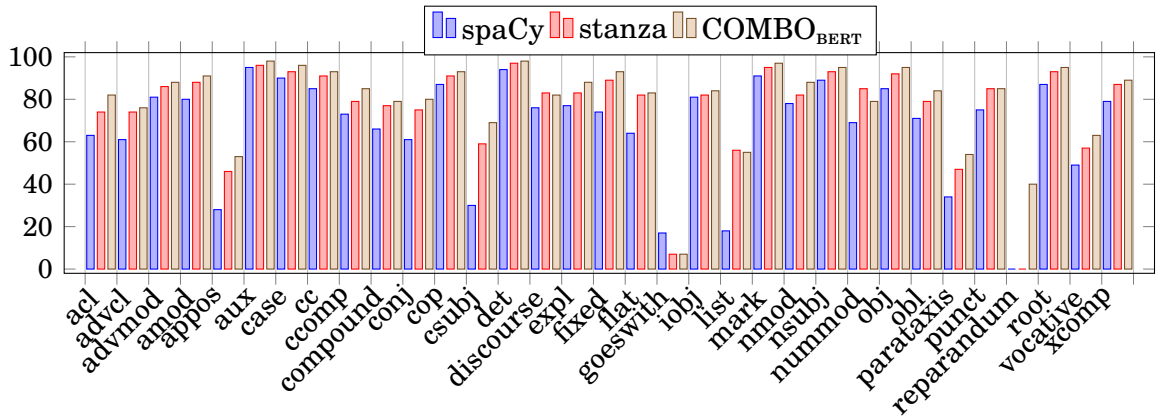


Figure 3.6.4. Evaluation of predicted grammatical functions (UDEPREL) in the English test set (F-$_1$-scores).

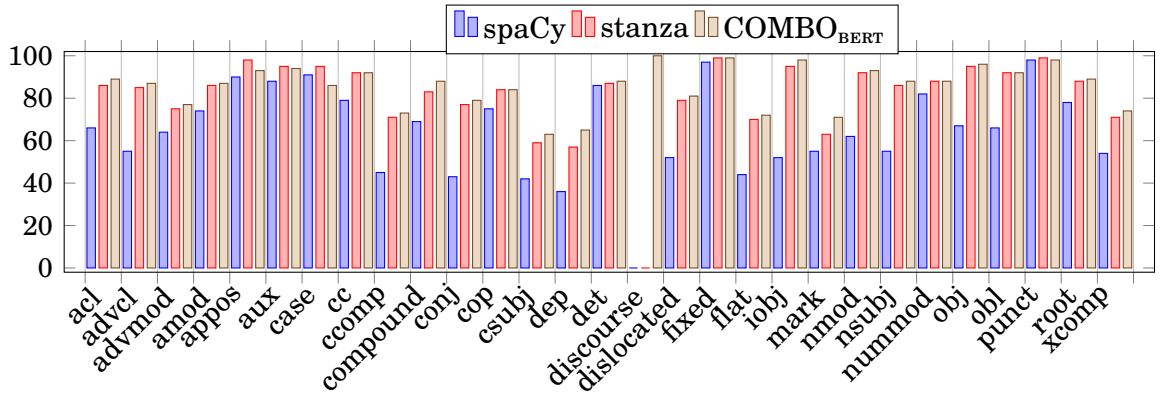


Figure 3.6.5. Evaluation of predicted grammatical functions (UDEPREL) in the Korean test set (F-$_1$-scores).

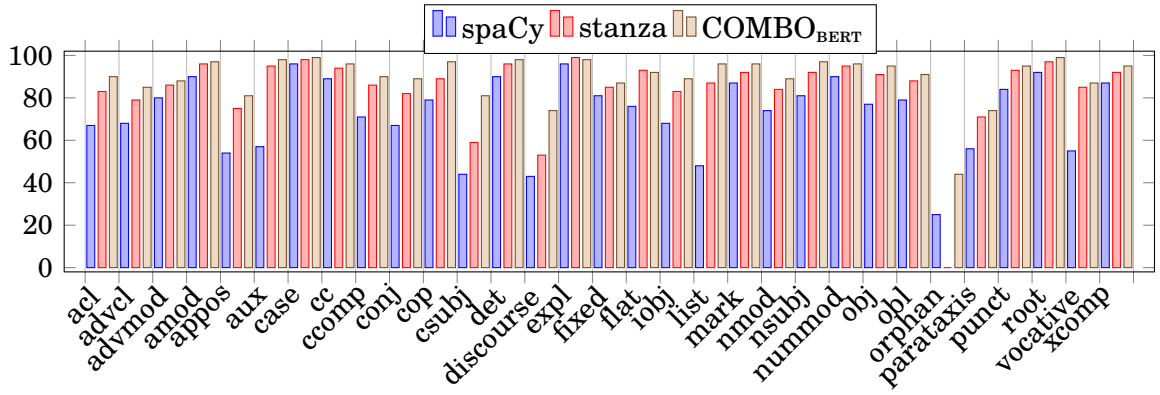


Figure 3.6.6. Evaluation of predicted grammatical functions (UDEPREL) in the Polish test set (F-$_1$-scores).

4. Gated Adapters for Multi-Domain Neural Machine Translation

Title	Gated Adapters for Multi-Domain Neural Machine Translation
Authors	Mateusz Klimaszewski, Zeno Belligoli, Satendra Kumar, and Emmanouil Stergiadis
Conference	26th European Conference on Artificial Intelligence (ECAI 2023)
Year	2023

Abstract

The Adapter framework introduces lightweight modules that reduce the complexity of Multi-Domain Machine Translation systems. Compared to fine-tuned models, Adapters train faster, do not overfit, have smaller memory requirements, and maintain the base model intact. However, just like fine-tuned models, they need prior information about the domain of the sentence. Otherwise, their performance decreases for out-of-domain and unknown-domain samples. In this work, we propose a solution that does not require the information and can decide on the sample's origin on-the-fly without compromising quality or latency. We introduce a built-in gating mechanism utilising a knowledge distillation framework to activate a subset of softly-gated, domain-specific Adapters that are relevant to the sentence. The effectiveness of the proposed solution is demonstrated through our experiments on two language pairs, using both in-domain and out-of-domain datasets. Our analysis reveals that Gated Adapters provide significant benefits, particularly in the case of ambiguous, misclassified samples, resulting in an improvement of over +5 COMET points.

4.1. Introduction

Neural Machine Translation (NMT) emerged as a go-to solution for Machine Translation, providing state-of-the-art results, especially in high-resource scenarios [14, 179, 142]. NMT models are usually trained using large, general-purpose parallel corpora. Therefore, to limit one of the known shortcomings of NMT – out-of-domain translation [98], there is a need to perform domain adaptation and improve the quality in the unknown domain, which might not be well represented in the parallel corpora.

Multi-Domain Machine Translation (MDMT) is a technique aimed at addressing the shortcomings of a general-purpose NMT model in translating text that falls outside its scope from various domains. According to Koehn and Knowles [98], a domain is characterized by a corpus from a particular source and may differ in terms of topic, genre, style, level of formality, among other things. This complexity underscores the challenge of MDMT. While fine-tuning one model for each domain is a straightforward approach that has been proven to be effective [59], it becomes challenging to implement in real-world scenarios where the number of domains and language pairs is substantial.

Recently, the Adapter framework [83] has been introduced as an alternative to regular fine-tuning. Adapters are lightweight modules injected into a pre-trained

model and fine-tuned to a specific task. This method requires training only newly introduced parameters, keeping the base model frozen. In a multi-domain setting, one Adapter per domain must be trained. However, unlike fine-tuned models, Adapters can be deployed together when they share the same base model. On the downside, the domain of each sentence must be known at inference time to activate the right Adapter. When the origin of the sentence is unknown or out-of-domain (we refer to both cases as an unannotated domain), a classifier is typically used to predict a likely domain [95]. This solution has two drawbacks: (i) it comes with a latency cost, as a pipeline approach increases the overall complexity, and (ii) it requires extra computation resources (i.e. additional GPU unit) to perform on-the-fly classification.

In this work, we propose a built-in gating mechanism, named Gated Adapters (`GAD`), to handle unannotated domains without compromising quality or latency. Gated Adapters extend the Adapter framework with the gates learnt via knowledge distillation [79]. The gates perform a fusion between sample-relevant Adapter modules. In contrast to the Adapters, `GAD` performs a soft-gating, i.e. multiple Adapters might be triggered, rather than a hard-gating when only one Adapter is used. Soft-gating in Adapter modules allows them to share relevant, cross-domain knowledge with each other (i.e. enhancing positive transfer learning). This is unlike the standard Adapters, which isolate a medical Adapter from a law one, for example. Additionally, the proposed method does not require an external classifier during inference and performs the domain prediction on-the-fly.

We evaluate the Gated Adapters on in- and out-of-domain translation, showing that the performance is on-par or better than the previous work. Moreover, our analysis reveals that in the case of ambiguous, misclassified examples (i.e. samples where the external classifier would assign an incorrect label), `GAD` outperforms other MDMT systems. To summarise, our contributions are as follows:

- We propose Gated Adapters as an extension to Adapters in the MDMT setting that does not require an external classifier at inference when the origin of the sentence to translate is unknown.
- We present an extensive evaluation of two language pairs: English to Polish and English to Greek, with six domains per pair.

4.2. Method

4.2.1. Adapters

Adapters [83] are lightweight modules injected into a pre-trained model and trained on new data while keeping the pre-trained model frozen. This means that

Adapters train only a fraction of the parameters of the initial model. Furthermore, because Adapters do not alter the base model, unlike conventional fine-tuning, there is no need to maintain a separate model for each task (e.g. domain).

In the standard NMT setup [19], an Adapter (AD) processes a transformer hidden state x at a layer i and consists of a residual connection [77], a layer norm LN [13] and two linear layers: down-project D and up-project U, creating a bottleneck with an activation function ReLU [119].

$$\mathrm{AD}_i(x_i) = U(\mathrm{ReLU}(D(\mathrm{LN}(x_i)))) + x_i \tag{4.1}$$

4.2.2. Gated Adapters

This work extends the Adapter framework by introducing a gating mechanism that allows the system to handle sentences from any domain and decide on its domain on the fly. The module provides probability-based soft gating that, given a set of domain-specific Adapters, multiplies each Adapter's output by a factor proportional to the probability of the sentence belonging to the Adapter's domain. This approach follows the mixture-of-experts (MoE) technique [162]; however, in contrast to regular MoEs, the experts in our proposed model have a pre-defined role – they are domain-specific modules.

In the following subsections, we describe (i) the gating mechanism and (ii) the knowledge distillation framework used to train the gates. The overview of our method is presented in Figures 4.2.1 and 4.2.2.

Gating mechanism

The gating mechanism is injected at each transformer layer i and acts as a weighted average over the output of each Adapter at that layer:

$$x_{\mathrm{out}} = \sum_{d}^{D} g_{\mathrm{norm}_d} \mathrm{AD}_d(x_{\mathrm{in}}) \tag{4.2}$$

where $x_{\mathrm{in}} \in R^{\mathrm{hidden\_dim}}$ is the Adapter's input, and g_{norm} is computed as:

$$g_{\mathrm{norm}} = norm(W_g \times \mathrm{agg}(x_{\mathrm{in}_T})) \tag{4.3}$$

Here $W_g \in R^{|D| \times \mathrm{hidden\_dim}}$ is a matrix of learnable weights, norm is a general normalisation function, and agg is a general aggregation function over all the time steps T $x_{\mathrm{in}_T} = x_{1:\mathrm{T}}$ for the encoder layers, and over all the steps up to the current one

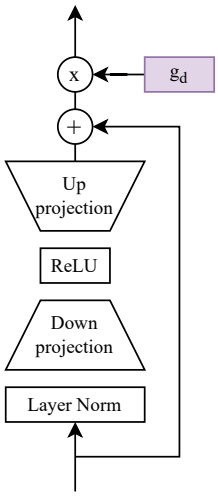


Figure 4.2.1. The schema illustrates a single Gated Adapter module, where the Adapter's output is multiplied by the probability value provided by the external gating module (g_d). This probability value indicates the degree to which a sentence belongs to the domain represented by the Adapter.

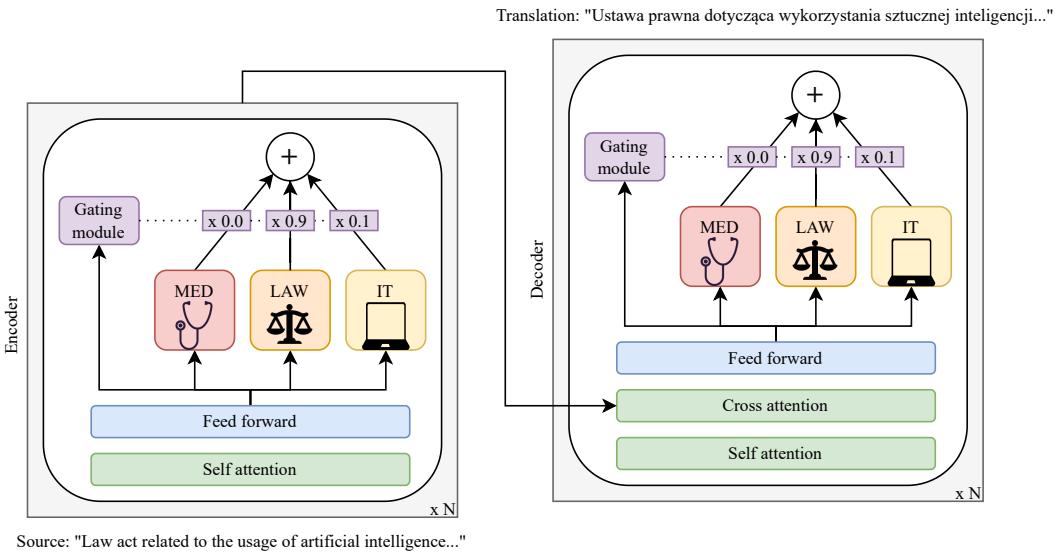


Figure 4.2.2. Overview of Gated Adapters. Given a sentence, the gating module predicts the probability of the sentence belonging to each domain. The probabilities behave as a weighting factor for the corresponding domain-specific Adapters. In the example sentence, the gates lean towards law and IT Adapters and discard the medical one, as the text concerns an AI-related law act.

$x_{\text{in}_T} = x_{1:t}$ for the decoder layers ($x_{\text{in}_T} \in R^{\text{hidden\_dim} \times |T|}$). In this work, we set norm to a softmax with a temperature parameter β and agg to a standard average operation.

Knowledge distillation

A standard NMT model is trained using a cross-entropy loss ($\mathcal{L}_{\text{CE}}$) with label smoothing [171]. We extend this setup and apply the knowledge distillation framework [79] to learn the values of the gates. Given a source sentence s, we estimate a probability distribution over domains conditioned on the sentence. As the actual distribution is unknown, we provide it as an estimation from an external classifier ($P_{\text{clf}} = P_\theta(d|s)$, implementation details in Section 4.3.2). The additional objective, Kullback–Leibler divergence ($\mathcal{L}_{\text{KL}}$), teaches the gates to mimic the teacher model.

We train the model jointly, in the same manner as Adapters, freezing everything but the parameters of Adapters and gates. The hyperparameter α weights the impact of the additional loss function, and τ is a softmax temperature used to estimate the probabilities P_g (obtained as a softmax function over the gates values g from Equation 4.3).

$$\mathcal{L}_{\text{KL}} = \tau^2 D_{\text{KL}}(P_{\text{clf}_\tau} || P_{g_\tau}) \tag{4.4}$$

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{KL}} \tag{4.5}$$

4.3. Experiments

4.3.1. Data

Our experimental setup involves two language pairs: English to Polish and English to Greek. We initiated the experiments by training a general-purpose machine translation model using ParaCrawl [18] as a baseline (`BASE`). To ensure the effectiveness of our approach, we selected a diverse set of domains from OPUS [175], including medical, legal, and IT domains, which vary significantly in terms of style, level of formality, and domain-specific terminology. By incorporating these domains, we aimed to demonstrate the robustness of our approach in handling various domain adaptation scenarios. The chosen six domains are listed below:

- `LAW`: legal documents from JRC Acquis
- `IT`: combination of KDE4 (only EN→PL), PHP, GNOME and Ubuntu localisation files
- `SUB`: a subset of OpenSubtitles 2018[1]

[1] `http://www.opensubtitles.org`

Table 4.3.1. Statistics of the training corpora as the number of parallel sentences. The table does not include 2000 parallel sentences per domain for validation and test purposes.

	English→Polish	English→Greek
BASE	33M	20,1M
LAW	838k	1244k
IT	96k	89k
SUB	1854k	1780k
TALK	206k	257k
MED	229k	257k
REL	108k	59k
SUM	3331k	3686k

- TALK: TED Talks transcripts [154]
- MED: medical documents from European Medicines Agency (EMEA)
- REL: Bible (EN→EL) [36] and Koran (EN→PL)

The statistics of the training data after pre-processing (including punctuation normalisation, ratio, language [109], length and dictionary-based filtering) are presented in Table 4.3.1. We held 2000 examples per domain for evaluation purposes (1000 for validation and 1000 as a test set).

Our analysis of the data involved utilising an SVM classifier[2] with averaged BERT [48] embeddings as features to measure the A-Proxy [22] distance between the domains. The A-Proxy distance is a measure that falls within the range of 0 to 2, where 0 indicates a perfect domain match and 2 represents complete separability. As shown in Figure 4.3.1, the domains were correctly separated, with SUB and TALK demonstrating the closest relationship and LAW exhibiting the greatest distance from the others.

4.3.2. Systems

We employed the Transformer Base [179] architecture implemented in fairseq[3] [123] for all our models. It consists of six encoder and six decoder layers, with an embedding dimension of 512, an FFN of 2048, and eight attention heads. The source and target embeddings are shared and tied with the output layer. We tokenised the data using a unigram SentencePiece model [99, 100] with a size of 32k. Table 4.3.2 presents the parameters of all the systems described in the following sections.

[2] As implemented in scikit-learn [129]

[3] fairseq architecture: `transformer_wmt_en_de`

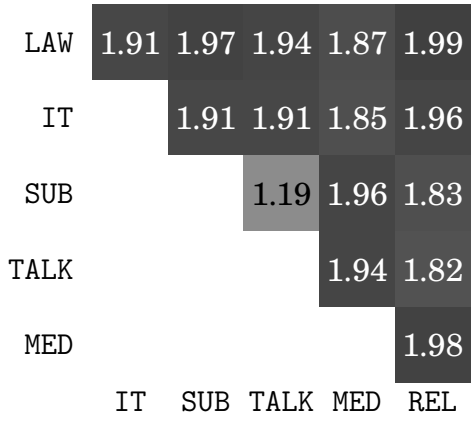


Figure 4.3.1. The A-Proxy distance between the domains.

Table 4.3.2. The rounded number of overall and trainable parameters of the evaluated models. In square brackets, we denoted the relative difference to the BASE model.

Model	Parameters	Trainable
BASE	79M	79M
FT	6×79M [$+5 \times 79$M]	6×79M
MIX	79M	79M
TAG	79M [+3k]	79M
AD	98M [+19M]	19M
GAD	98M [+19M]	19M

Baselines

The experiments begin with training a general-purpose model, labelled as BASE, using large-scale data from ParaCrawl (refer to Table 1). This model is evaluated on all domains to establish a lower bound for all MDMT systems and is used as a pre-trained Machine Translation model (i.e. MDMT systems build upon the model rather than starting from scratch). Additionally, we employ this model for fine-tuning (FT) to create a set of domain-specific models, each for a different domain. This strategy is an upper bound for MDMT systems; however, it has limitations when scaling the solution across various language pairs and domains as it produces a separate model per domain. The training details of these and the following models are described in the Appendix.

We employ two MDMT, non-adapter baselines: (i) MIX, which is straightforward training the model on a concatenation of domain corpora, (ii) TAG, which adds a domain-control mechanism in the form of domain-specific tag included into each source sentence and enables the model to differentiate between the domains [95, 165]. Both methods use BASE as a starting checkpoint.

Adapter-based systems

To assess the effectiveness of the Adapter-based systems, we examine the standard Adapters (AD) and compare their performance with the newly proposed Gated Adapters (GAD). Compared to other MDMT systems, Adapter-based systems train a fraction of parameters (refer to Table 4.3.2) as these methods freeze the NMT model and train only the Adapter modules. For AD and GAD, we rely on MIX as a starting checkpoint [138] and use Adapter modules with a bottleneck of 2 (i.e. reducing the dimensionality via the down-project layer D by 2). The rest of the training procedure is consistent with the other MDMT systems.

Gated Adapters use RoBERTa[4] [108] as a base model for an external classifier required for knowledge distillation (see Eq. 4.4). We train two classifiers, one per language pair, using the English side of the parallel corpora as the datasets are not equivalent, e.g. EN→EL uses Bible and EN→PL Koran. To prevent data leakage, only the training parallel corpora are used to train and validate the models. The evaluation of the classifiers is presented in Table 4.3.3. The classifiers serve not only as a teacher model for GAD (i.e. required only during training) but also as a means of predicting the domain for the TAG and AD baselines during inference. In the results section, we denote the systems that rely on the classifier during inference with an index CLF. Those baselines are constructed as a pipeline solution, i.e. first, the

[4] roberta-large [187]

Table 4.3.3. Quality of the RoBERTa-based classifiers in terms micro-averaged F1 score.

Model	F1
EN → PL	95.65
EN → EL	94.83

classifier predicts a domain, and then the MDMT model translates a sentence. For clarity, we present the ORACLE version as an upper bound of those systems, which always utilises the ground truth domain.

Furthermore, we up-sampled all the domains to the one with the highest sentence count. This step prevents high-resource domains from overshadowing other domains' weights. Otherwise, we noticed in preliminary experiments that a high-resource domain could harm a similar (in terms of domain closeness) lower-resource domain (i.e. `TALK` in `SUB–TALK` pair).

4.3.3. Metrics

Following the study and recommendation of Kocmi et al. (2021) [96], we use COMET[5] [153] as main evaluation metric. In addition, we provide chrF [143] and BLEU [125] scores using SacreBLEU[6,7] [144]. Due to computational and time constraints, we compute three independent runs exclusively for Adapter-based systems (`AD` and `GAD`) and report an average score with standard deviation for them.

4.3.4. Results

Table 4.3.4 presents the evaluation results. We report both per-domain scores and aggregated metrics - unweighted and weighted averages `AVG`, `wAVG`. The `AVG` metric should be treated as the primary metric determining the quality of an MDMT system in the case of balanced test distribution; the `wAVG` in the case of the test distribution matching the training one. The weights for the latter metric are derived from the ratio of domain-specific data based on the number of sentences (see Table 4.3.1).

Gated Adapters perform the best out of all MDMT systems based on aggregated metrics in both language pairs. Overall, Gated Adapters are on-par or better than not only methods that require a classifier but also their oracle version (e.g. Adapters with ground truth domain tag) while simultaneously providing the possibility of

[5] We use `wmt20-comet-da` COMET model and multiply results by 100

[6] `chrF2|#:1|c:mixed|e:yes|nc:6|nw:0|s:no|v:2.2.0`

[7] `BLEU|#:1|c:mixed|e:no|tok:13a|s:exp|v:2.2.0`

Table 4.3.4. Translation performance measured using COMET. For each system, we aggregate scores using an unweighted and weighted average, where the weights come from the ratio of domain-specific training data based on the number of sentences. We report average scores over three runs with a standard deviation for AD and GAD.

	LAW	IT	SUB	TALK	MED	REL	AVG	wAVG
				English – Polish				
BASE	84.58	39.18	30.74	46.24	54.79	11.21	44.46	46.50
FT	97.30	66.73	48.93	53.26	86.29	105.36	76.31	66.28
TAG_{ORACLE}	95.76	61.20	47.22	53.03	82.78	89.12	71.52	64.00
AD_{ORACLE}	96.11±0.27	63.47±1.90	46.98±0.53	53.36±0.80	83.26±0.38	98.83±1.25	73.67±0.36	64.38±0.31
MIX	95.79	61.93	**47.50**	52.42	82.56	86.88	71.18	64.05
TAG_{CLF}	95.73	61.30	45.90	52.41	82.12	88.86	71.05	63.16
AD_{CLF}	**95.81**±0.21	63.37±1.73	46.31±0.36	52.81±0.66	82.71±0.40	98.04±1.21	73.17±0.31	63.84±0.23
GAD	95.47±0.13	**64.78**±0.87	46.97±0.35	**53.57**±0.22	**83.55**±0.67	**103.67**±0.23	**74.67**±0.30	**64.44**±0.18
				English – Greek				
BASE	80.74	24.31	38.53	66.89	35.38	10.67	42.75	53.74
FT	87.55	73.09	53.28	77.02	74.68	78.31	73.99	68.87
TAG_{ORACLE}	88.34	62.70	51.30	75.19	72.36	46.03	65.99	67.13
AD_{ORACLE}	88.07±0.10	68.75±2.65	51.91±0.47	75.71±0.54	74.21±0.61	51.23±0.46	68.31±0.51	67.72±0.31
MIX	87.79	66.51	51.51	73.74	73.30	45.64	66.42	67.09
TAG_{CLF}	**88.19**	61.97	50.05	73.58	72.08	45.85	65.29	66.32
AD_{CLF}	88.04±0.08	68.60±2.54	50.91±0.41	73.97±0.26	73.79±0.62	50.91±0.41	67.70±0.45	67.07±0.26
GAD	87.69±0.10	**69.34**±0.35	**52.29**±0.25	**74.67**±0.59	**73.83**±0.29	**70.23**±1.22	**71.34**±0.40	**68.00**±0.14

handling unannotated domains. Especially in the case of the AVG metric, the GAD outperforms AD_{CLF} with +1.5 and +3.5 COMET point gain in English to Polish and English to Greek language pairs correspondingly. We report other automatic evaluation metrics: chrF and BLEU, in the Appendix.

4.4. Method analysis

This section dissects the Gated Adapters to examine the method's advantages and explain its performance beyond the main, in-domain results. We analyse the cross-domain and out-of-domain capabilities in Sections 4.1 and 4.3, measure the efficiency in Section 4.2 and perform an ablation study in 4.4.

4.4.1. Knowledge sharing

The preliminary analysis revealed that the SUB and TALK domains are the most related in terms of A-Proxy distance. This observation is consistent with the achieved results. In Table 4.3.4, the CLF versions of TAG and AD models have the most decrease in quality compared to the ORACLE counterpart in these two domains. Additionally, the confusion matrix of the classifier presented in Figure 4.4.1 demonstrates that those two domains were the most difficult to distinguish in the EN→PL dataset. While the other domains are classified with high accuracy, rarely making any mis-

Table 4.4.1. Translation evaluation of the misclassified sentences from the test dataset using COMET. Gated Adapters outperform both methods that require a classifier at inference whenever the classifier fails to predict a correct domain label.

	EN→PL	EN→EL
TAG_{CLF}	31.98	45.46
AD_{CLF}	42.59	55.33
GAD	**48.33**	**64.01**

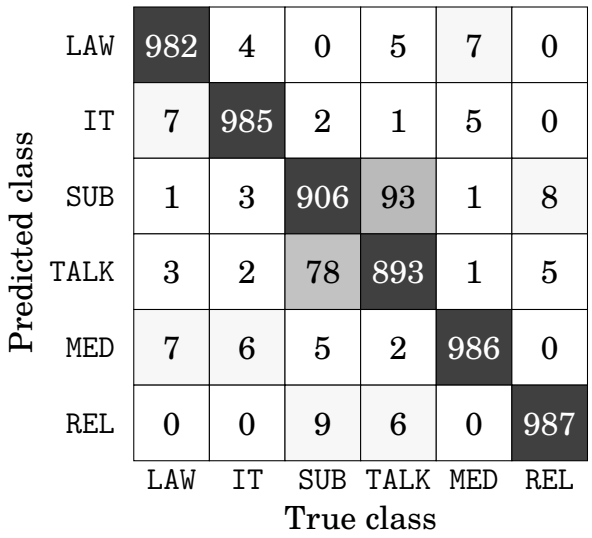


Figure 4.4.1. Confusion matrix for the classifier in the EN→PL language pair. Albeit its overall high quality, the model makes almost exclusively mistakes between the SUB–TALK domain pair.

takes, the pair of SUB and TALK is the most troublesome to both classifiers (the same phenomenon appears in EN→EL, see Appendix).

The GAD model can handle ambiguous, cross-domain examples (i.e. examples for which two or more domains are probable according to the classifier) because it has learnt a soft gating mechanism that allows knowledge sharing among outputs of different Adapters (see Equation 4.2). Considering just misclassified (i.e. with a predicted non-ground truth domain label) examples from the test dataset, the GAD outperforms its counterpart in both language pairs by over 5 and 8 COMET points. Table 4.4.1 presents the results of the evaluation. The quality of the methods that require a classifier during inference (TAG, AD) drops significantly compared to GAD. While the Gated Adapters use the same classifier during training (the classifier makes the same mistakes), GAD is aware of the uncertainty (i.e. soft-gating instead of hard-gating) and learns to handle such cases during knowledge distillation. Table 4.4.2 presents the translation examples with the impact of misclassification, showing that a wrong domain label may lead to a meaningless translation in extreme cases.

Table 4.4.2. Misclassified examples from the EN→PL test dataset. AD generates higher-quality translation when we manually provide the right domain during inference (i.e. by changing from CLF to ORACLE). At the same time, GAD does not rely on an external classifier and therefore does not suffer from the aforementioned issue. Misclassification can lead to meaningless translation, as in the second example, where the model produces a relevant translation only after providing the correct label, i.e. changing from TALK to REL ("trunk" given the context is incorrectly translated to 🚗 "bagażnik" instead of 🌴 "pień").

Source	These events are often transitory.
Reference	Zaburzenia te są często przemijające.
$\text{AD}_{\text{CLF=TALK}}$	Zdarzenia te są często przejściowe.
$\text{AD}_{\text{ORACLE=MED}}$	Zdarzenia te są często przemijające.
GAD	Zdarzenia te są często przemijające.
Source	The birth pangs brought her to the trunk of a date palm.
Reference	I doprowadziły ją bóle porodowe do pnia drzewa palmowego.
$\text{AD}_{\text{CLF=TALK}}$	Pangi narodzin przywiozły ją do bagażnika palmy randkowej.
$\text{AD}_{\text{ORACLE=REL}}$	I przyniosły ją bóle porodowe do pnia drzewa palmowego.
GAD	I doprowadziły ją bóle porodowe do pnia palmy daktylowej.

4.4.2. Efficiency

In order to evaluate the efficiency of our proposed model, we conducted experiments to compare inference time. We calculated the number of generated tokens and translation duration per domain and aggregated the values to report the number of processed sentences and tokens per second. We performed the inference per domain because each domain differs in its characteristics, such as average sentence length. All experiments were run on a single NVIDIA V100 GPU, with a batch size of 64 and with greedy decoding.

Our method introduces two additional drawbacks that affect efficiency: (i) the gating module and (ii) the requirement of using all the Adapters to perform the aggregation. To limit the impact of the latter drawback, we implemented a parallel approach instead of a sequential one. In the sequential approach, domain-specific Adapters are processed one at a time, whereas in the parallel approach, all steps are processed simultaneously, except for layer norms, via multi-channel linear layers (i.e. the down-project D and up-project U layer with the non-linear function ReLU) instead of iterating over domains.

We present the comparison between the Adapters + classifier (AD_{CLF}) pipeline versus Gated Adapters in Table 4.4.3. For reference, we also include the raw Adapters, which assume a scenario where the right domain is known. The Gated Adapters outperform the pipeline scenario of Adapters preceded by a classifier. While GAD adds an overhead over the Adapters setup, it does not require an additional classifier.

Table 4.4.3. Efficiency comparison in terms of processed sentences per second and tokens per second between the classifier and Adapters pipeline ($\texttt{AD}_{\text{CLF}}$) and Gated Adapters (GAD). For reference, we include the standalone Adapters ($\texttt{AD}_{\text{ORACLE}}$) values, that assumes prior domain knowledge for each sentence.

	sentences/s	tokens/s
$\texttt{AD}_{\text{ORACLE}}$	88.11	2091.77
$\texttt{AD}_{\text{CLF}}$	51.64	1225.90
GAD	**58.65**	**1392.12**

The Gated Adapters can use just one device (i.e., GPU) at a time, whereas the pipeline requires two devices to avoid the overhead of checkpoint loading for online translation. Additionally, compared to the Adapters without a classifier, GAD does not need information about the origin of a sample.

4.4.3. Out-of-domain evaluation

In Sections 4.3.4 and 4.4.1, we demonstrated the in-domain and cross-domain capabilities of the Gated Adapters. However, as the gates are merely a distilled version of an external classifier, the out-of-domain capabilities remain in question. Therefore, we performed an additional, out-of-domain evaluation to verify the gating mechanism's robustness. This step checks whether the GAD's quality does not decrease for out-of-domain samples and persists quality of the classifier as in $\texttt{AD}_{\text{CLF}}$. Both MDMT systems attempt to use an external classifier (RoBERTa in $\texttt{AD}_{\text{CLF}}$) or an internal one (gates in GAD) to map out-of-domain samples into one of the pre-defined domains.

We evaluate the models on two out-of-domain datasets: Flores-200 devtest [44] and WMT'20 News test [20] dataset (the latter available only for EN→PL). We report the results in Table 4.4.4. Although the gates match around 0.03% size of the classifier in terms of the number of parameters (the gates introduce less than 40k new parameters), they retain similar performance and generalizability. On both datasets, GAD presents on-par results with $\texttt{AD}_{\text{CLF}}$ while using a distilled version of the classifier embedded into the model and making domain prediction on-the-fly, verifying the gating mechanism's robustness.

4.4.4. Knowledge distillation ablation

We conducted an ablation study to examine the effect of treating gates as a regular classifier and using cross-entropy loss instead of knowledge distillation, which is in line with the method used in previous works by Britz et al. (2017) [29] and Pham et

Table 4.4.4. COMET scores for an out-of-domain evaluation on the Flores-200 devtest and News WMT'20 test dataset. We report average of three runs with the standard deviation.

	EN → PL	EN → EL
	Flores	
AD_{CLF}	$57.61_{\pm 0.24}$	$67.00_{\pm 0.46}$
GAD	$57.63_{\pm 0.24}$	$66.90_{\pm 0.32}$
	News	
AD_{CLF}	$46.44_{\pm 0.71}$	–
GAD	$46.65_{\pm 0.46}$	–

Table 4.4.5. Ablation on the EN→PL validation dataset comparing training the gates as a classifier (CE) against the knowledge distillation (KD) framework. We report AVG and wAVG for COMET score

	AVG	wAVG
CE	72.02	61.51
KD	**73.62**	**62.73**

al. (2020) [137]. The validation dataset was used to present the ablation results in Table 4.4.5. The outcomes demonstrate the advantages of the proposed approach, as it enables Gated Adapters to match and even surpass the quality of Adapters.

4.5. Related work

The mixture-of-experts models are gaining more traction in the Machine Translation field [162]. Recently, Dua et al. (2022) [51] propose a temperature heating mechanism and dense pre-training for easing the convergence of MoE MT models. The NLLB Team [44] presented a multilingual MoE model on a larger scale, breaking the 200 languages barrier.

Adapters, as a specific version of a MoE, were lately also used for the task of domain adaptation. The work of Vu et al. (2022) [181] focuses on the domain generalisation task via Adapter leave one-out strategy. In the similar, regularisation focused way, (and additionally improving overall complexity), Rücklé et al. (2021) [155] proposed AdapterDrop technique to drop out Adapter layers, similarly to removing Transformer layers [57]. The presented works can be applied to any Adapter-based MDMT system and could be applied with the GAD model.

Pfeiffer et al. (2021) [132] introduce the AdapterFusion technique, which, as our work, shares the knowledge between multiple Adapter modules. However, their

method requires additional, separate training as they extend the regular Adapter setup with a fusion layer on top of the multiple Adapters and train the new parameters with the base model and Adapter modules frozen. Moreover, they focus on the multi-task setup rather than the multi-domain one. Pham et al. (2020) [137] propose to extend a highway version of residual Adapters with domain classifiers on top of an encoder and decoder and decide on a domain on a word-per-word basis. They evaluate the solution in the MDMT setting. As in the work of Pfeiffer et al. (2021) [132], they use an additional training procedure that requires separate training of the classifiers.

4.6. Conclusions

In this work, we present an extension to the Adapters framework in the MDMT setting called Gated Adapters, which perform soft-gating over multiple domain-specific Adapters. We evaluate the validity of the proposed solution on two language pairs and across six domains.

We show that GAD not only improves upon regular Adapters but also demonstrates resistance to domain misclassification and provides high-quality translation, even when the sentences are ambiguous in terms of their domain. Moreover, the proposed solution does not require an external classifier at the inference time, making the use more efficient – it requires less computational resources than a pipeline solution of a classifier with an MDMT model (e.g. Adapters AD).

4.7. Limitations

The main limitation of our technique is the data requirements. We test our method on high-resource language pairs and domains that fall within the mid-to-high resource range. There is not enough evidence that the technique would work for (extremely) low-resource domains, considering the up-sampling required by Gated Adapters. Future work could investigate if this is a shortcoming of the proposed method. Furthermore, we rely on a classifier that is built upon a pre-trained language model [108], which may not be sufficiently robust to attain the desired level of accuracy in low-resource languages or may not be accessible at all.

Table 4.8.1. Translation performance measured using chrF. For each system, we aggregate scores using an unweighted and weighted average, where the weights come from the ratio of domain-specific training data based on the number of sentences. We report average scores over three runs with a standard deviation for AD and GAD.

	LAW	IT	SUB	TALK	MED	REL	AVG	wAVG
				English – Polish				
BASE	63.25	52.07	44.20	48.24	52.27	36.64	49.44	49.78
FT	71.03	61.42	48.42	48.33	72.44	90.79	65.41	57.51
$\text{TAG}_{\text{ORACLE}}$	69.93	58.66	48.16	48.57	66.40	80.51	62.04	56.27
$\text{AD}_{\text{ORACLE}}$	70.31±0.06	59.08±0.71	47.97±0.23	48.40±0.14	67.21±0.32	86.52±0.64	63.25±0.14	56.51±0.14
MIX	70.18	58.69	**47.91**	48.20	67.13	80.02	62.02	56.20
TAG_{CLF}	69.92	58.65	**47.91**	48.44	66.33	80.39	61.94	56.11
AD_{CLF}	**70.30**±0.06	59.06±0.69	47.83±0.13	48.27±0.19	67.17±0.29	86.24±0.62	63.14±0.12	56.41±0.08
GAD	70.16±0.08	**60.59**±0.11	47.78±0.34	**48.46**±0.06	**67.73**±0.23	**89.71**±0.08	**64.07**±0.05	**56.55**±0.18
				English – Greek				
BASE	60.20	54.60	46.00	58.65	55.66	42.90	53.00	52.50
FT	63.02	72.42	50.19	61.26	75.01	85.33	67.87	58.12
$\text{TAG}_{\text{ORACLE}}$	63.37	68.30	50.24	60.75	69.87	68.43	63.49	57.50
$\text{AD}_{\text{ORACLE}}$	63.15±0.05	69.51±1.18	50.26±0.22	60.93±0.25	71.20±0.36	71.24±0.28	64.38±0.21	57.62±0.12
MIX	63.16	67.84	**50.16**	60.27	70.27	68.09	63.30	57.37
TAG_{CLF}	**63.30**	68.11	49.64	60.41	69.81	68.33	63.27	57.16
AD_{CLF}	63.09±0.04	69.39±1.09	49.43±0.17	60.59±0.20	**71.17**±0.36	71.14±0.29	64.13±0.21	57.17±0.11
GAD	63.05±0.12	**71.10**±0.14	49.56±0.08	**60.73**±0.20	71.03±0.16	**81.45**±0.63	**66.15**±0.16	**57.43**±0.04

4.8. Appendix

4.8.1. Experiment Details

In this section, we describe the hyperparameters used during training. We use Adam optimiser [89] with betas set to 0.9 and 0.98 and a learning rate of 0.0005. The training has a warm-up phase set to 4000 steps and an inverse square root scheduling. The dropout rate is set to 0.3. We use beam search with a beam of size five during inference [170]. For Adapter-based models, we use the bottleneck of 2 (i.e. reducing the dimensionality via D to 256). For Gated Adapters, we introduced three additional hyperparameters. Those values were chosen on the EN→PL validation dataset, and we used the same values for EN→EL. The impact of the $\mathcal{L}_{\text{KL}}$ was set arbitrary to 0.5. For normalisation temperature β and $\mathcal{L}_{\text{KL}}$ temperature τ, we used the values of 2.0 and 0.1. The former was picked from $\{0.1, 0.5, 1.0, 2.0, 2.5\}$, the latter: $\{0.1, 0.5, 1.0, 2.0\}$. We trained all our models using 4 NVIDIA V100 GPUs, except the BASE model, which used 8 NVIDIA V100 GPUs.

4.8.2. Evaluation

Tables 4.8.1 and 4.8.2 present the evaluation results measured using chrF and BLEU.

Table 4.8.2. Translation performance measured using BLEU. For each system, we aggregate scores using an unweighted and weighted average, where the weights come from the ratio of domain-specific training data based on the number of sentences. We report average scores over three runs with a standard deviation for AD and GAD.

	LAW	IT	SUB	TALK	MED	REL	AVG	wAVG
				English – Polish				
BASE	41.36	29.36	20.20	19.31	24.91	12.09	24.54	25.79
FT	52.54	41.23	26.87	21.33	55.97	87.38	47.55	37.37
$\text{TAG}_{\text{ORACLE}}$	50.97	37.43	26.44	21.14	46.14	72.41	42.42	35.45
$\text{AD}_{\text{ORACLE}}$	51.37±0.03	37.91±0.87	26.29±0.24	21.36±0.19	47.32±0.46	81.06±0.81	44.22±0.23	35.85±0.19
MIX	51.28	37.36	**26.42**	21.20	47.10	71.65	42.50	35.56
TAG_{CLF}	50.96	37.46	26.21	21.03	46.07	72.23	42.33	35.30
AD_{CLF}	**51.36**±0.04	37.87±0.86	26.21±0.13	**21.27**±0.29	47.27±0.42	80.70±0.76	44.11±0.20	35.79±0.11
GAD	51.28±0.12	**40.11**±0.11	26.41±0.26	21.24±0.08	**48.31**±0.20	**85.89**±1.17	**45.54**±0.03	**36.18**±0.13
				English – Greek				
BASE	34.41	31.45	21.37	31.11	28.99	14.88	27.04	27.12
FT	39.01	55.83	28.14	36.16	61.74	75.29	49.36	36.14
$\text{TAG}_{\text{ORACLE}}$	39.38	49.32	**27.98**	35.15	52.36	46.58	41.80	34.84
$\text{AD}_{\text{ORACLE}}$	39.09±0.08	51.37±1.92	27.87±0.07	35.60±0.24	54.34±0.39	50.83±0.57	43.18±0.37	34.98±0.06
MIX	39.12	48.84	27.56	34.82	52.98	46.33	41.61	34.56
TAG_{CLF}	**39.29**	48.95	**27.32**	34.89	52.30	46.52	41.54	34.46
AD_{CLF}	39.01±0.05	51.15±1.77	26.73±0.11	35.36±0.23	**54.30**±0.40	50.75±0.57	42.88±0.37	34.38±0.10
GAD	39.05±0.18	**54.25**±0.07	26.77±0.24	**35.38**±0.18	54.13±0.24	**68.17**±1.34	**46.29**±0.27	**34.76**±0.13

4.8.3. Confusion matrix

Figure 4.8.1 presents the classifier quality for the EN→EL dataset.

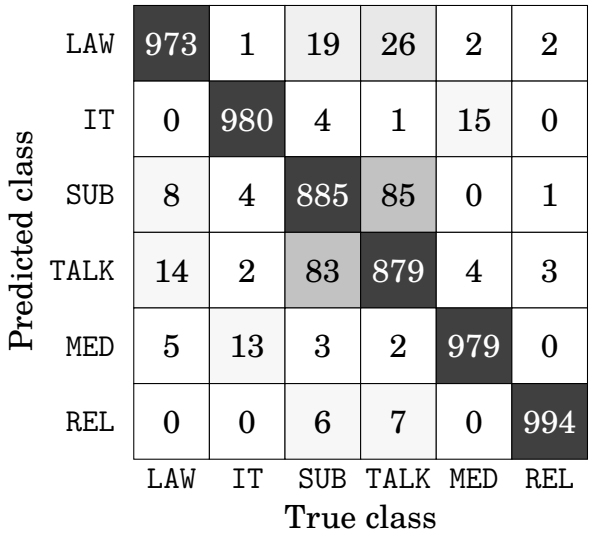


Figure 4.8.1. Confusion matrix for the classifier in the EN→EL language pair. Albeit its overall high quality, the model makes almost exclusively mistakes between the SUB–TALK domain pair.

5. No Train but Gain: Language Arithmetic for training-free Language Adapters enhancement

Title	No Train but Gain: Language Arithmetic for training-free Language Adapters enhancement
Authors	Mateusz Klimaszewski, Piotr Andruszkiewicz, and Alexandra Birch
Conference	The 31st International Conference on Computational Linguistics (COLING 2025)
Year	2025

Abstract

Modular deep learning is the state-of-the-art solution for lifting the curse of multilinguality, preventing the impact of negative interference and enabling cross-lingual performance in Multilingual Pre-trained Language Models. However, a trade-off of this approach is the reduction in positive transfer learning from closely related languages. In response, we introduce a novel method called language arithmetic, which enables training-free post-processing to address this limitation. Extending the task arithmetic framework, we apply learning via addition to the language adapters, transitioning the framework from a multi-task to a multilingual setup. The effectiveness of the proposed solution is demonstrated on three downstream tasks in a MAD-X-based set of cross-lingual schemes, acting as a post-processing procedure. Language arithmetic consistently improves the baselines with significant gains, especially in the most challenging case of zero-shot application. Our code and models are available at `https://github.com/mklimasz/language-arithmetic`.

5.1. Introduction

The recent progress of large language models has raised the question of how well they perform not just in English but across multiple languages which has spurred interest in Multilingual Pre-trained Language Models (MLLMs) [42, 160, 5]. These models serve as general-purpose solutions that can be adapted and applied to various Natural Language Processing tasks. Notably, MLLMs demonstrate zero-shot cross-lingual capabilities, allowing them to generalise effectively to downstream tasks even when pre-trained in a language different from the target language.

The positive transfer of abilities from both related languages and high-quality training data from unrelated languages has meant that MLLMs have reported state-of-the-art performance in low-resourced languages [118]. However, this benefit does not always extend to high-resourced languages [97]. In such cases, the quality of MLLMs tends to decrease compared to their monolingual counterparts [121, 114, among others] due to negative interference phenomena [185]. Additionally, the curse of multilinguality [42] reveals the existence of a trade-off between language coverage and model capacity. Consequently, MLLMs must carefully limit the number of languages included during the pre-training phase.

Modular deep learning (MDL) [134] methods help to avoid negative interference and limited model capacity , enabling the extension of MLLMs to support any number of languages [19, 177, 139, 135, 131]. MDL methods adapt the model to arbitrary tasks and languages by isolating components from each other (and the backbone

MLLM) via parameter-efficient extensions. Examples of parameter-efficient modules are adapters [152, 83], which are low-budget (in terms of parameters) bottleneck layers that increase an MLLM size by just a fraction. Language adapters [135] allow the modularisation of language-specific knowledge by training on a raw, unlabelled corpus for specific languages.

The limitation of the MDL and language adapters is their isolation. While they lift the curse of multinguality and prevent negative interference, at the same time, language adapters limit the possible impact of positive transfer. Previous attempts to address these challenges — such as training bilingual [126] or language-family [38] adapters — do not scale effectively. In our work, we tackle this limitation as a post-processing step. Leveraging recent insights from task arithmetic [86], specifically *learning via addition*, we augment language adapters with missing related language knowledge – a concept we term **language arithmetic**. Remarkably, this training-free approach can enhance not only existing language adapters but also offer zero-shot performance.

To summarise, our contributions are as follows:

- A novel training-free post-processing method named language arithmetic that enhances language adapters.
- We conduct a cross-lingual evaluation on three downstream tasks (NER, NLI and QA) and two Multilingual Pre-trained Language Models (XLM-R, mBERT) with test cases that include zero-shot and low-resource setups in a diverse group of 13 languages.
- We provide an analysis of language arithmetic internal components (including a comparison with task arithmetic) and show improvement up to 3 F1 points without any additional training involved.

5.2. Background

Our research builds upon the task arithmetic contributions of Ilharco et al. (2023) [86] and Zhang et al. (2023) [197]. The following Section provides the background and serves as a gentle introduction to the concept of task vectors and task arithmetic.

5.2.1. Task vectors & Task arithmetic

Let us assume that we have access to a pre-trained model with its weights denoted $\theta_{pre} \in R^d$ and a fine-tuned version of the same model on a task t represented by $\theta^t_{ft} \in R^d$. The task vector $\tau_t \in R^d$ is an element-wise difference between models'

weights.

$$\tau_t = \theta^t_{ft} - \theta_{pre} \tag{5.1}$$

The task vectors can be part of multiple arithmetic operations, e.g. *learning via addition*. This operation is an addition operation between two task vectors and the base model, i.e. we add two differences between the fine-tuned models and the pre-trained version with weights controlling the impact.

$$\theta_{multi-task} = \theta_{pre} + \lambda_1 \tau_{t_1} + \lambda_2 \tau_{t_2} \tag{5.2}$$

The lambdas can be further normalised to sum to one, i.e. $\lambda_2 = 1 - \lambda_1$ and simplifying notation with just λ.

$$\theta_{multi-task} = \theta_{pre} + \lambda \tau_{t_1} + (1 - \lambda) \tau_{t_2} \tag{5.3}$$

While we define learning via addition for two tasks, the same procedure can be applied to multiple tasks.

Task arithmetic allows us to forge a multi-task model from a separate, task-specific set of fine-tuned models, preserving high accuracy (although a shared pre-trained starting point is required, e.g. the same Language Model). Moreover, vectors from different tasks are typically close to orthogonal, and Ilharco et al. (2023) [86] speculate that this enables the combination of task vectors via addition with minimal interference.

In our work, we focus on parameter-efficient fine-tuning (PEFT) and use language adapters. Therefore, we reduce the task vector and underlying model weights represented by θ to newly introduced parameters (i.e. we exclude the backbone MLLM, which is frozen across all the models, following the work of Zhang et al. (2023) [197]).

5.3. Method

We propose language arithmetic that transitions the task arithmetic concept from a multi-task to a multilingual setup. In this Section, we describe the language arithmetic alongside its application as a training-free, post-processing step to a MAD-X cross-lingual framework [135].

5.3.1. Language arithmetic

We formulate a language arithmetic (LA) concept by substituting the task in task vectors and arithmetic with a language. This approach means that instead

of merging downstream tasks, we target a problem of cross-lingual performance. We propose to apply *learning via addition* to languages, and in Appendix 5.9.5, we demonstrate the discrepancies when comparing language and task vectors. Our study focuses specifically on the language adapters [139, 135]. Due to overlapping abbreviations, we use the LA exclusively as the former, i.e. language arithmetic. In the learning via addition, we limit the parameters to language adapters and simplify the notation that θ represents the adapters' weights and τ is referred to as a language vector. As we operate in a language space instead of a task, the t is replaced with a language, i.e. its language code in the notation. The example equation describes a language arithmetic operation between an English and a Spanish adapter.

$$\theta_{LA} = \theta_{pre} + \lambda\tau_{en} + (1-\lambda)\tau_{es} \tag{5.4}$$

Throughout the paper, the equation above is abbreviated as a function: $LA(en, es)$ with lambda as a default parameter. Additionally, for clarity reasons, we denote target language in a subscript to distinguish different use-cases. For example, $LA_{fr}(en, es)$ means that language arithmetic between English and Spanish is applied to a different language - in this case French (zero-shot application), or $LA_{es}(en, es)$ meaning that the target is Spanish (non-zero-shot use-case, due to Spanish being a part of the LA equation).

Language arithmetic is a training-free method, taking advantage of already pre-trained modules. The sole requirement is a validation dataset on which the λ parameter can be established. While in our work, we use a pretty fine-grained step (0.05) to determine the λ value (i.e. we run evaluations for $\lambda \in [0, 1]$ with a provided step), our analysis showcased that it is possible to increase the value and limit the computation burden even more (details in Section 5.5.1).

5.3.2. Application

We evaluate our post-processing method as an extension of the MAD-X framework [135] to challenge our solution in a cross-lingual manner. The overview of the schema is presented in Figure 5.3.1.

The MAD-X consists of the following steps:

1. Training language adapter(s)
2. Training task adapter
3. Cross-lingual inference

In our work, we introduce an additional step:

4. Post-processing via language arithmetic

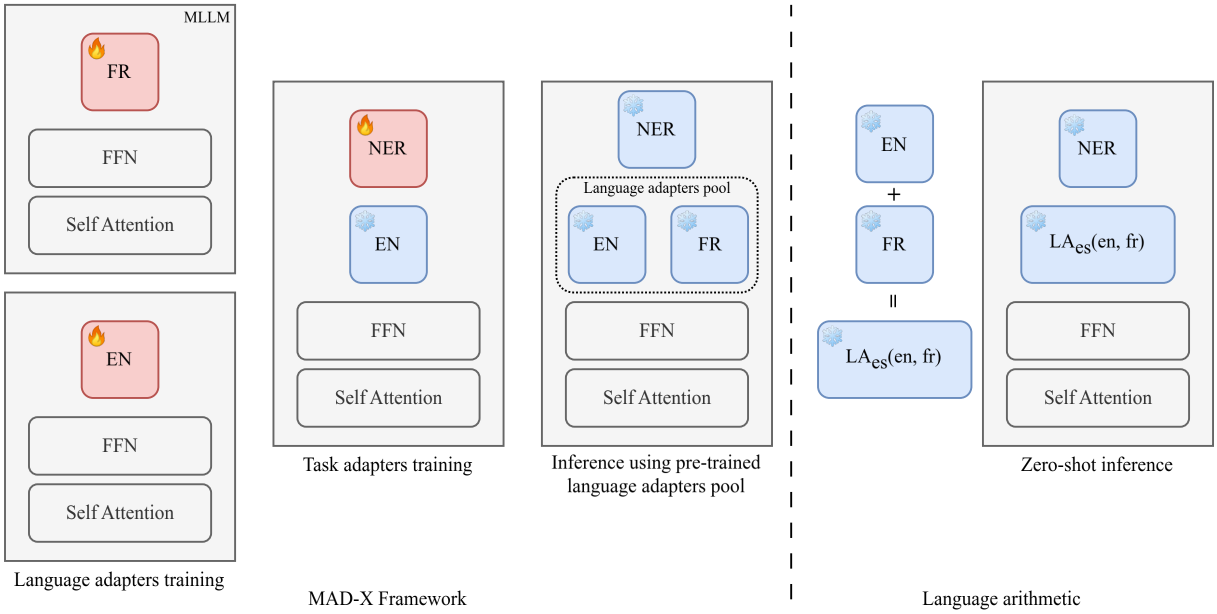


Figure 5.3.1. Language arithmetic as an extension of the MAD-X framework. Given language and task adapters (left), language arithmetic (right) enables post-processing, training-free improvement in two use-cases: (i) zero-shot where a language adapter for a target language was not trained (presented in the figure as Spanish, which was not part of existing language adapters pool, $LA_{es}(en, fr)$) or (ii) to improve existing language adapters via arithmetic with either related language or a language on which task adapter was trained (e.g. $LA_{fr}(en, fr)$).

In the following Sections 5.3.2-5.3.2, we present the framework and our proposed post-processing via language arithmetic extension, exploring two use cases: (i) a zero-shot case, where a target language adapter does not exist and (ii) an enhancement case, where we prove existing language adapters (in high- and low-resourced languages).

Training language adapter(s)

In the first step, the MAD-X framework trains language adapters. These adapters are trained on raw corpora using masked language modelling loss in a self-supervised manner. The MLLM is frozen during this step, and we only optimise the newly introduced adapter. The training must be done for languages corresponding to the downstream tasks (e.g. if we have an English NER dataset, we need an English language adapter, apart from other desired target languages). Additionally, the adapters form a pool that is leveraged during cross-lingual inference.

Training task adapters

The following step freezes a backbone MLLM and a language adapter and trains a task adapter on a downstream task dataset. Given a set of tasks or if a new task

appears, we can repeat this step as long as the required language adapter exists in the available pool, i.e. a language adapter that matches the task's language.

Cross-lingual inference

Having trained a task adapter, we can leverage a pool of pre-trained language adapters and obtain a cross-lingual performance by connecting any existing language adapter with a newly trained task adapter (i.e. routing first via language adapter and then task adapter). The growing pool of pre-trained adapters can be accessed at public repositories like AdapterHub [133] and reused for further use cases.

Post-processing via language arithmetic

Our method builds upon the MAD-X framework in two enhancement scenarios.

First, we assume a situation where the pool of language adapters does not contain a desired target language, i.e. a zero-shot scenario. In contrast to the previous works that try to improve existing adapters, this use-case is an alternative to routing via either a related language or a task language (here, by task language, we understand the language on which the task adapter was trained, in contrast to a target language - on which we want to evaluate). Instead of choosing the better-performing proxy, language arithmetic proposes to combine these two (with better results, as shown in Section 5.4.2).

In the second language adapter enhancement scenario, we apply language arithmetic as a more common goal, trying to improve existing language adapters; however, our method does that without any training. Here, we combine the existing target language adapter with either a related language (we define related languages in Section 5.4.1) or, once again, a task language.

5.4. Experiments

5.4.1. Experimental setup

Datasets

Downstream evaluation is performed on three tasks: Named Entity Recognition (NER), Natural Language Inference (NLI) and Question Answering (QA), covering jointly 13 languages,[1] while the training - to perform cross-lingual evaluation - is performed on the English data. For the NER task, we use the WikiANN [149] dataset and for NLI - XNLI [43]. The QA evaluation is done on XQuAD [12] (we

[1] ar, bg, de, el, es, fr, hi, ru, sw, tr, ur, vi, zh; XQuAD does not cover 4 languages (bg, fr, sw, ur)

split data 50/50 into valid/test datasets), and the training uses SQuAD 1.1 [150]. Additionally, to evaluate a low-resource scenario for a language not covered during MLLM pre-training, we leverage the Assamese subset from IndicXNLI [3].

Related languages To automatically establish a related language needed for language arithmetic, we query URIEL and lang2vec library [106, 110]. During the related language query, we limited the options to 13 downstream task languages for which we had already pre-computed language adapters. This limitation means that for some languages, we would be able to find a stronger performing pairing and that the ceiling for our method is higher than the presented (we denote that the performance of hypothetical paring would also depend on data availability, i.e. a paired language must be not only related but also have representative corpora; based on this we show analysis and improved performance in Section 5.5.2). However, considering the limitations of the lang2vec, we decided to keep this simplification. At last, a language can have a set of equally good related languages. Therefore, in practical terms, it is not feasible for our study to train all possible options for each language - our simplification stands as a reasonable, real-world proxy. We provide the list of related languages in the Appendix 5.9.1.

Implementation & training

In our work, we focus on two of the most popular multilingual PLMs:[2] mBERT[3] [48] and XLM-R[4] [42]. We implement our method using the AdapterHub library [133]. For language adapters, we train on the Wikipedia corpora[5] for 250k steps with a learning rate of 1e-4, an effective batch size of 64 using a single GPUand the same initialisation. For task-specific training, we train for 100 epochs with the same learning rate and a batch size set to 16. We choose the final checkpoint based on validation dataset performance (for language adapters, we evaluate on a held-out subset of Wikipedia). In our main experiments, we report the scores as an average over three independent runs with different seeds (for both language and task adapters). Additionally, to improve efficiency and reduce GPU memory utilisation, we adopt a half-precision (FP16) setting.

[2] According to downloads from Huggingface `hf.co/models?language=multilingual&sort=downloads`

[3] `bert-base-multilingual-cased`

[4] `xlm-roberta-base`

[5] 20331101.xx checkpoint `hf.co/datasets/wikimedia/wikipedia`

5.4.2. Zero-shot evaluation

The zero-shot evaluation assumes a scenario where the language adapter pool does not contain a desired target language (e.g. lack of Spanish in Figure 5.3.1). The baselines are based on routing, i.e. we proxy either by an English adapter (proxy via a task language, as the task adapter was trained using English data) or a related language (e.g. French for Spanish). Language arithmetic serves a solution that, instead of choosing a better proxy, combines the adapter's tuple: $LA_t(en, rel)$, where rel symbolises a related language and t stands for a target language (e.g. $LA_{es}(en, fr)$).

Figures 5.4.1 and 5.9.1 present the results of the zero-shot experiment. Language arithmetic consistently outperforms the proxy baselines for all the setups, reaching over 3.1 F1 points improvement in the NER task and 1.1 F1 for QA (XLM-R). These results indicate that language arithmetic is a feasible, low-cost method that one can apply in the lack of an existing target language adapter.

Additionally, we investigate how the λ parameter impacts the downstream evaluation. The goal was to understand how much weight is given to English vs related language. We looked at the validation performance over different λ thresholds. While most cases set the value to over 0.5 (i.e. preferring the English side, given $LA_t(en, rel)$), the preferred values did not showcase any consistency and pattern. We analyse this further in Section 5.5.1.

5.4.3. Improving existing language adapters

This evaluation assumes that a target language adapter exists in the adapter pool. We test two cases, i.e. $LA_t(en, t)$ and $LA_t(rel, t)$, where rel is once again a related language and t is the target language. Additionally, we provide a combination of these two approaches (referred to as $LA_t(en/rel, t)$), where for each language, we choose a better pairing (so either en or rel). This solution resembles a practical compromise between cost and performance and serves as a proxy for the ceiling of our method (discussed in Section 5.4.1).

The results are presented in Figures 5.4.2 and 5.9.2. Compared to the baseline direct application of a target language adapter (i.e., the MAD-X method), the gains are not as significant as in the case of the zero-shot scenario. Moreover, in contrast to the previous Section's study, MLLMs showcase a different behaviour, as language arithmetic provides less benefit for XLM-R than for the mBERT model (e.g. +0.38 XLM-R vs +2.41 mBERT in F1, QA).

The drop in performance of language arithmetic compared to the zero-shot use case is not surprising. Given that a target language adapter is trained on a sig-

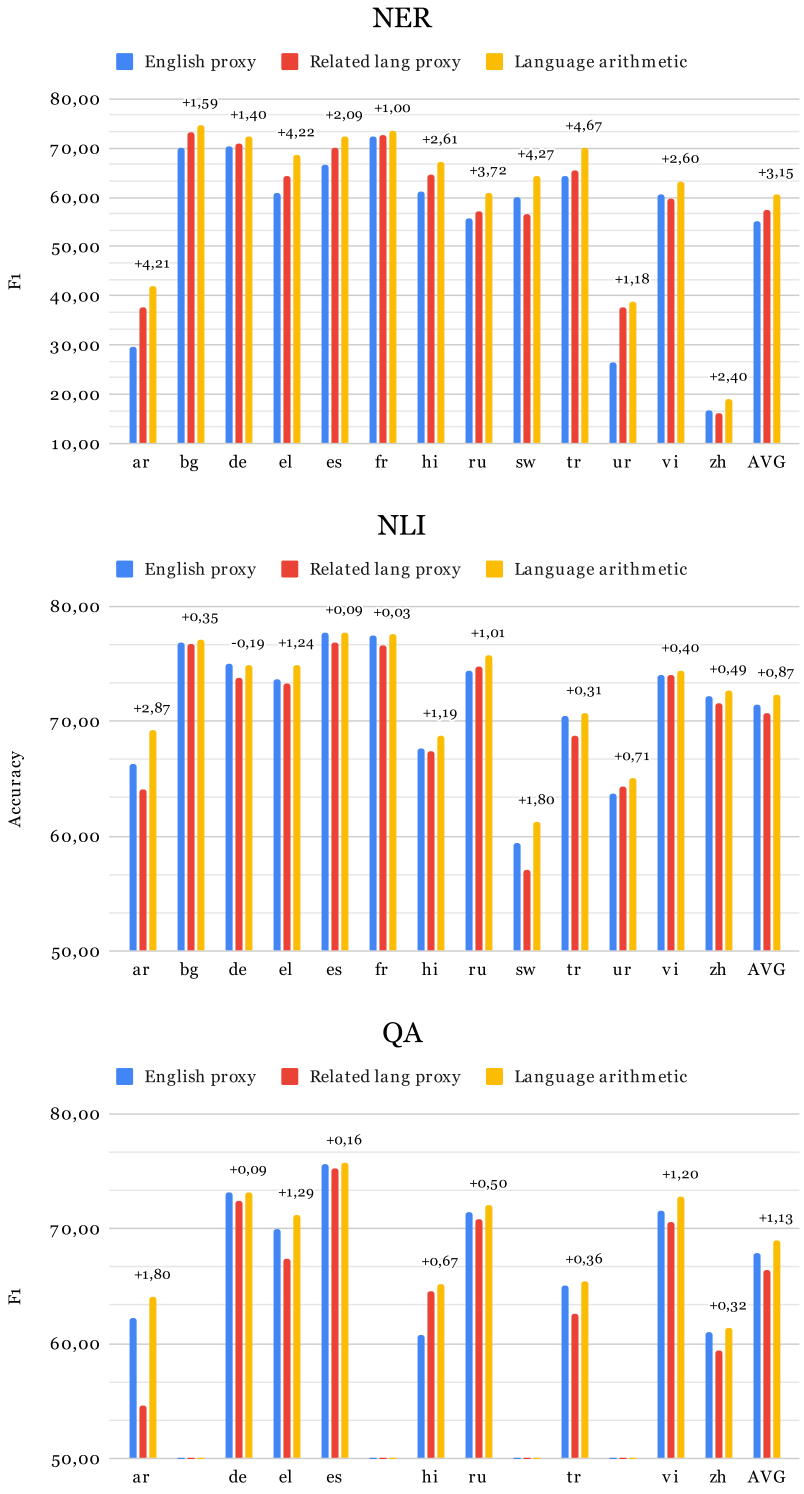


Figure 5.4.1. Zero-shot XLM-R language arithmetic evaluation, where one side of the arithmetic is an English adapter, and the other is related to the target language adapter (e.g. French for Spanish - $LA_{es}(en, fr)$). The values above bars present a relative difference to a better proxy. See Figure 5.9.1 for the mBERT model.

nificant corpus, it gives less room for improvement (this is not the case in the low-resource regimes, as shown in the following Section). This is also a potential explanation of a different behaviour between the evaluated MLLMs, considering the overall more robust performance of XLM-R over mBERT. However, considering the cost-to-performance ratio and the minimal fatigue that our post-processing method enforces on existing MAD-X pipelines, we can see a constant gain on average across all the experiments and training runs.

5.4.4. Low resource evaluation

Training a language adapter might be troublesome for high-resourced languages due to massive corpora requiring significant computational resources.[6] On the other hand, in most languages, we lack data to train a strong language adapter, i.e. language-specific corpora might be either too small or unavailable [70]. We investigate whether LA can help in such cases. We test our solution in three cases and define three (actual and simulated) evaluation scenarios:

- Assamese (as) - low resource language, additionally not used in the pre-training of a base MLLM,
- Swahili (sw) - low resource language, used in the pre-training,
- French (fr) - high resource, used in the pre-training. We simulate cases from low to high resources.

We train a series of language adapters with different token budgets for each language, from 10k to 10M (or 1B for French; we limit this particular study to the XLM-R model). Afterwards, we compare the usage of such adapters directly against language arithmetic with three adapters (we use $LA_t(t, en)$, where $t \in \{as, sw, fr\}$).

Figure 5.4.3 presents the results of the evaluation performed on the downstream tasks. The most gain is visible in the most challenging scenario, during the evaluation on the Assamese dataset. In this case, the backbone MLLMs did not encounter the language during the pre-training phase. Although the difference becomes less pronounced in the NER task as we approach the limits of available data, there remains a significant margin for NLI - the difference can be explained by the overlap in the corpora (Wikipedia) between NER and language adapter training tasks, following the findings of Gururangan et al. (2020) [68]. For Swahili, where the language is part of the pre-training, the flattening effect begins earlier and affects both tasks. Nevertheless, leveraging language arithmetic still yields improvements.

[6] Although in our experimental setup, we train each adapter for the same number of steps and choose the best checkpoint based on the validation performance, for low-resourced languages, one could apply an early stopping mechanism in a production-level pipeline.

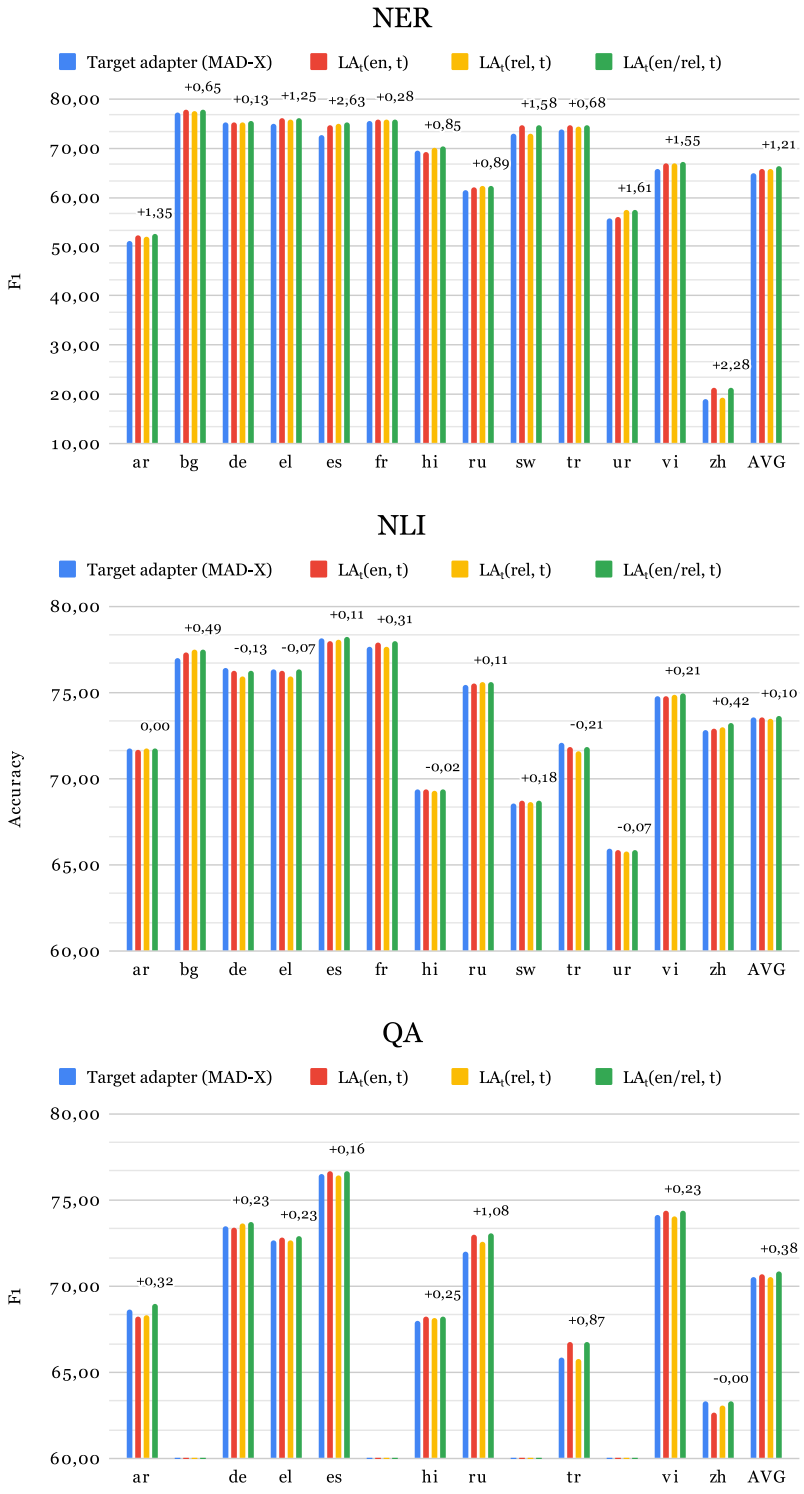
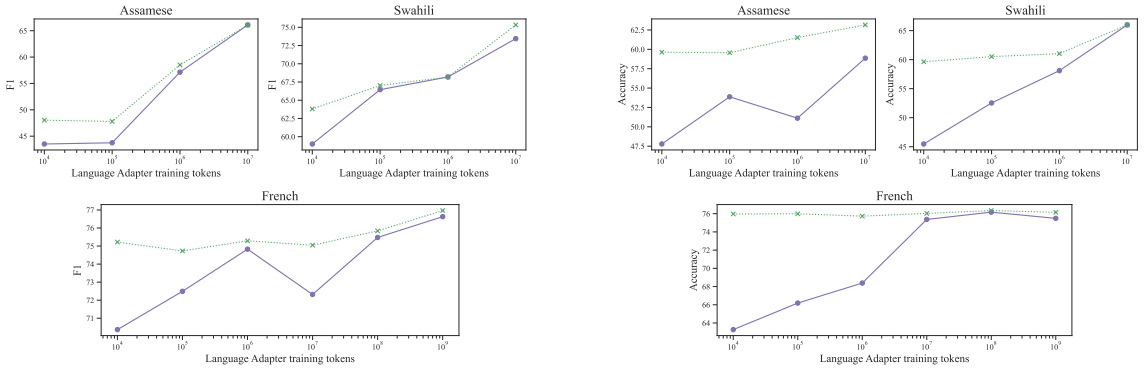


Figure 5.4.2. Variants of language arithmetic compared to the MAD-X method in the use-case to improve an existing target language adapter. The values above bars present a difference between a better LA setup and the MAD-X framework for the XLM-R model (see Figure 5.9.2 for mBERT).



(a) NER (b) NLI

Figure 5.4.3. NER and NLI evaluation of a set of adapters trained on a Wikipedia subset showcases that language arithmetic $LA_t(t, en)$ (green, dotted line) provides significant gains when compared against direct usage of the adapter (violet, solid line), especially in a very low-resource regime. The x-axis represents the token budget of each trained language adapter.

The simulated case of French showcases that even with a relatively weak language adapter (trained on 10k tokens), the language arithmetic can restore existing knowledge and results in high performance for the language. Moreover, comparing the adapters trained with a different token budget, the results remain similar, without significant fluctuations. We believe that this phenomenon happens because the MLLM has seen a much higher amount of French in the pre-training procedure than Swahili (over 35 times more tokens in XLM-R pre-training; moreover, French is in the top 15 represented languages). Therefore, even undertrained French adapters have a relatively easy task once they are merged with a robust English adapter. In practical terms, this finding allows us to prototype new languages quicker by estimating the possible end product quality or might serve as an intermediate solution (until the full-corpora adapter is trained).

5.5. Analysis

5.5.1. Lambda impact

Our study estimates the λ parameter with a small step (0.05). This analysis investigates how sensitive this parameter is in the language arithmetic. Depending on multiple variables that include model and evaluation dataset sizes or a number of languages, running 20 evaluations might be costly (especially when using neural-based metrics, e.g. COMET [153]). Therefore, we analysed the potential impact of choosing a suboptimal lambda with a decrease in evaluation count. The

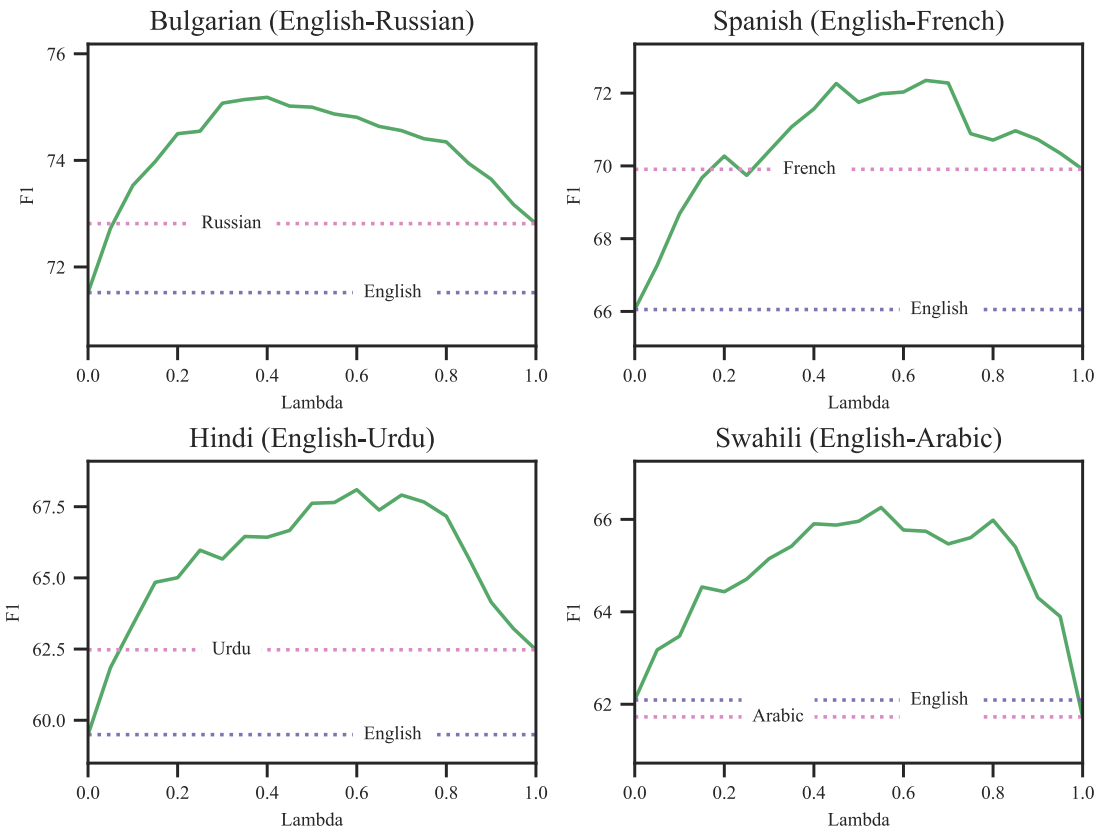


Figure 5.5.1. Interpolation of λ values for the zero-shot XLM-R scenario (NER, for NLI and QA see Appendix 5.9.4) on the validation dataset. The horizontal dashed lines represent the baseline scores for both languages used in language arithmetic.

breakdown includes a subset of languages on both tasks (using the XLM-R as a base model). We chose the zero-shot scenario where we performed LA between English and related language adapters.

In Figures 5.5.1 and 5.9.3, we plot the validation scores with the corresponding baselines, that is, the scores of using directly the adapters. The dotted lines are based on $\lambda = 0$ or $\lambda = 1$ for clarity, meaning we exclusively use the arithmetic equation's left or right side (i.e., a specific language). In most cases, a subset of valid λ values would improve over the baselines. Moreover, the analysis reveals that a coarser evaluation (e.g., with a step of 0.1 or 0.2) would be sufficient, reducing the required number of performed tests up to four times while maintaining most of the improvement. At last, setting the default $\lambda = 0.5$ would be near optimum for the analysed subset.

5.5.2. Language relatedness

Relatedness of languages is a difficult-to-define concept. At times, in our proposed framework, we might face a choice of multiple, seemingly equally related languages to use for the arithmetic operation. In this analysis, we decided to look at this aspect

Table 5.5.1. Impact of language relatedness on the language arithmetic. We compare different Romance languages as a right side of LA equation, i.e. $l2$ (both tasks use XLM-R model). We report an average over three runs.

$LA_{l_1}(l_1\downarrow, l_2\rightarrow)$		ca	es	fr	it	pt	ro
				NER			
Eval language	es	73.82	-	75.06	74.61	74.90	73.68
	fr	75.74	75.76	-	75.82	75.59	75,79
				NLI			
	es	78.23	-	78.04	77.96	77.94	78,02
	fr	77.95	77,65	-	77.70	77,47	77,60

via a glance at Romance languages. We trained an additional subset of language adapters and formed a pool of 6 languages: Catalan, French, Italian, Portuguese, Romanian and Spanish. Afterwards, we evaluated languages shared in our NER and NLI tasks (Spanish and French) by arithmetic with the entire Romance languages pool.

The results are presented in Table 5.5.1 and show that given a different related language (in this case, defined as coming from the same language family), there are minor scores fluctuation. The relative difference between the best and the worst language reaches around 1 F1 score in the NER task and around 0.3 in terms of accuracy points for the NLI task. This experiment indicates that a more sophisticated or hand-crafted language choice would improve the downstream results presented in Figure 5. However, it also shows that there is no free lunch, and results depend on a downstream task. For example, for Spanish evaluation, a French adapter is trained on the largest out of the listed languages raw corpora; therefore, for the NER task, it can leverage a bigger pool of seen during training Named Entities (at times language-independent or similar across languages) and perform the best given more data, even when there are closer related languages (according to methodology from Section 5.4.1 and Indo-European languages family tree).

5.6. Related Work

Knowledge composition from multiple, independently trained adapters has been widely discussed in the literature. However, unlike our work, the solutions require substantial changes to the vanilla adapter setup. The previous work either requires additional parameters to learn a parameterised composition function/a gating module to combine/steer the flow through the suitable adapter(s), or needs a specific training procedure that increases the complexity of the overall solution or, in

most cases, both [137, 132, 102, 126, 38, 92, 182]. Moreover, to prevent specifically negative interference, hyper-adapters [21] were proposed using hyper-networks [69], and Ansell et al. (2022) [8] applied sparse fine-tuning to compose task and language masks. Unlike the prior studies mentioned earlier, our attempt is training-free and does not modify the base architecture. The most conceptually similar work is proposed by Chronopoulou et al. (2023) [37]; however, they operate on the notion of sample similarity to a subset of domains in a domain adaptation regime. Additionally, concurrent to our work, Parović et al. (2019) [127] show intial potential of task arithemtic in cross-lingual transfer based on a full fine-tuning setup. However, in our work we focus on PEFT methods with additional, in-depth analysis. At last, we denote the rise of task arithmetic use cases, e.g. vision tasks or cross-task generalisation [166, 85].

5.7. Conclusion

We have proposed language arithmetic, which enhances language adapters based on task arithmetic learning via addition. It is a training-free method and functions as a post-processing technique for MAD-X. Our experiments have shown that LA is particularly beneficial in a zero-shot scenario, where we do not have access to a target language adapter. At last, we highlight the differences between language and task arithmetic.

In our future work, we plan to extend language arithmetic by incorporating more components into the sum. Additionally, we aim to adapt other elements of the task arithmetic framework, i.e. task analogies and forgetting via negation, to a multilingual setup with an analysis of the differences between multi-task and multilingual arithmetic context. Furthermore, we will evaluate LA's performance on various non-classification tasks.

5.8. Limitations

Our work was tested on English-centric task training and could be extended to different languages with more PEFT methods. Moreover, applying multi-source training based on the work of Ansell et al. (2023) [7] could provide better robustness of the task adapters and a more thorough analysis.

Table 5.9.1. Languages used in the experiments with corresponding related languages. Details are provided in Section 5.4.1.

Lang.	ar	bg	de	el	es	fr	hi
Related	sw	ru	fr	es	fr	es	ur
Lang.	ru	sw	tr	ur	vi	zh	
Related	bg	ar	bg	hi	ru	ar	

5.9. Appendix

5.9.1. Related languages

We present the list of related languages used in our experiments in Table 5.9.1 (details in Section 5.4.1).

5.9.2. Zero-shot evaluation

Figure 5.9.1 presents the results of the experiments described in Section 5.4.2 for the mBERT model.

5.9.3. Improving existing language adapters

Figure 5.9.2 presents the results of the experiments described in Section 5.4.3 for the mBERT model.

5.9.4. Lambda impact - NLI and QA

Figure 5.9.3 presents the analysis of lambda impact for NLI and QA tasks. For details, refer to Section 5.5.1.

5.9.5. Language vs task vectors

Task vectors exhibit high sparsity and orthogonality, as **(author?)** [86] observed. While the former characteristic can be denoted in language vectors (Figure 5.9.4), the latter displays different properties, in contrast to task vectors. In Figure 5.9.5, we visualise the cosine similarity between evaluated language vectors of language adapters. Notably, the minimal cosine similarity (0.19) surpasses the maximum (0.18) reported by previous research in the task space [86]. Interestingly, most pairs in the task space oscillate within the range of 0.01 to 0.03. At the same time, language vectors surpass 0.2 in almost each case, indicating that the orthogonality aspect is an inherent property of task adapters.

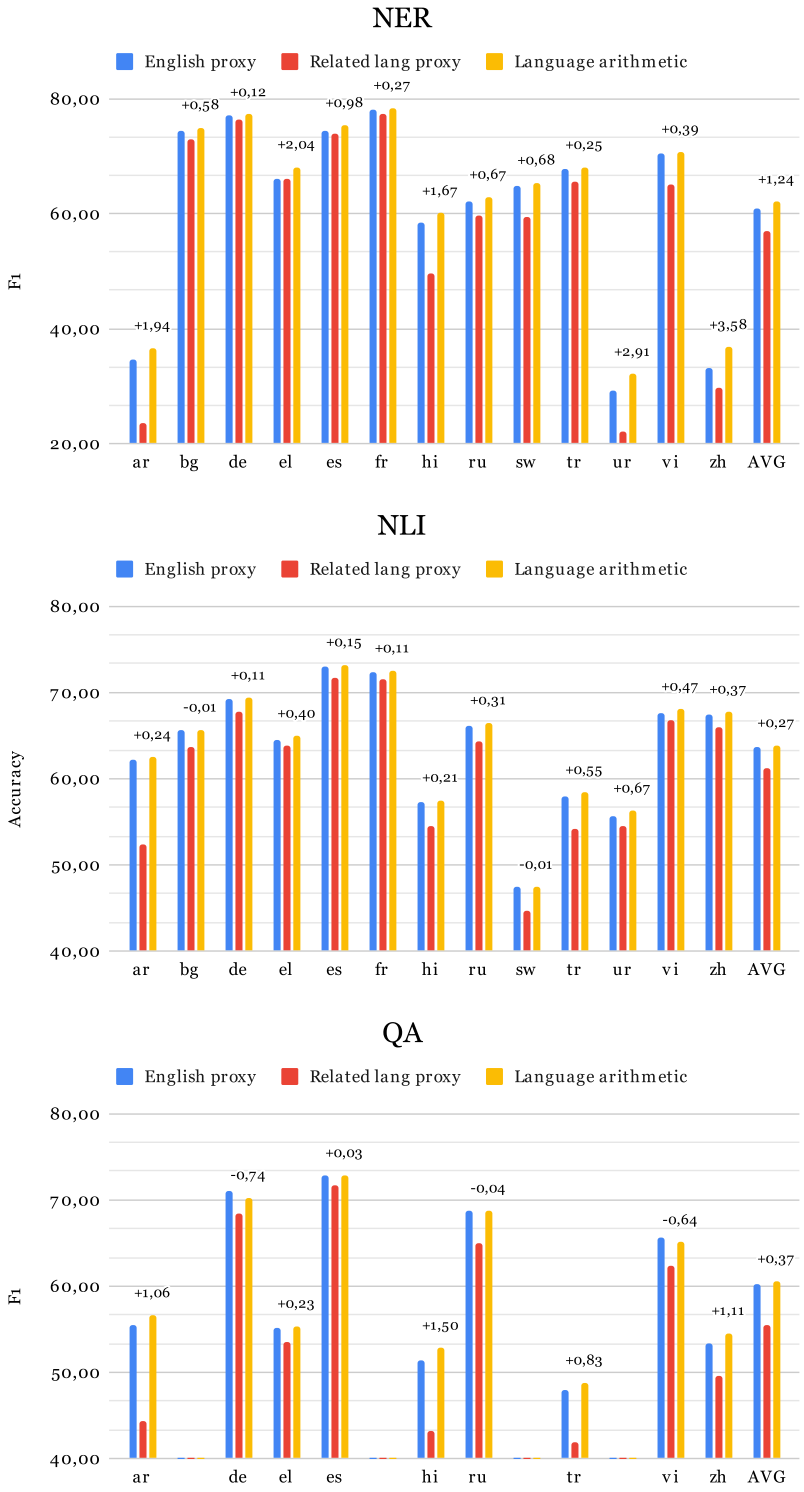


Figure 5.9.1. Zero-shot mBERT language arithmetic evaluation, where one side of the arithmetic is an English adapter, and the other is related to the target language adapter (e.g. French for Spanish - $LA_{es}(en, fr)$). The values above bars present a relative difference to a better proxy.

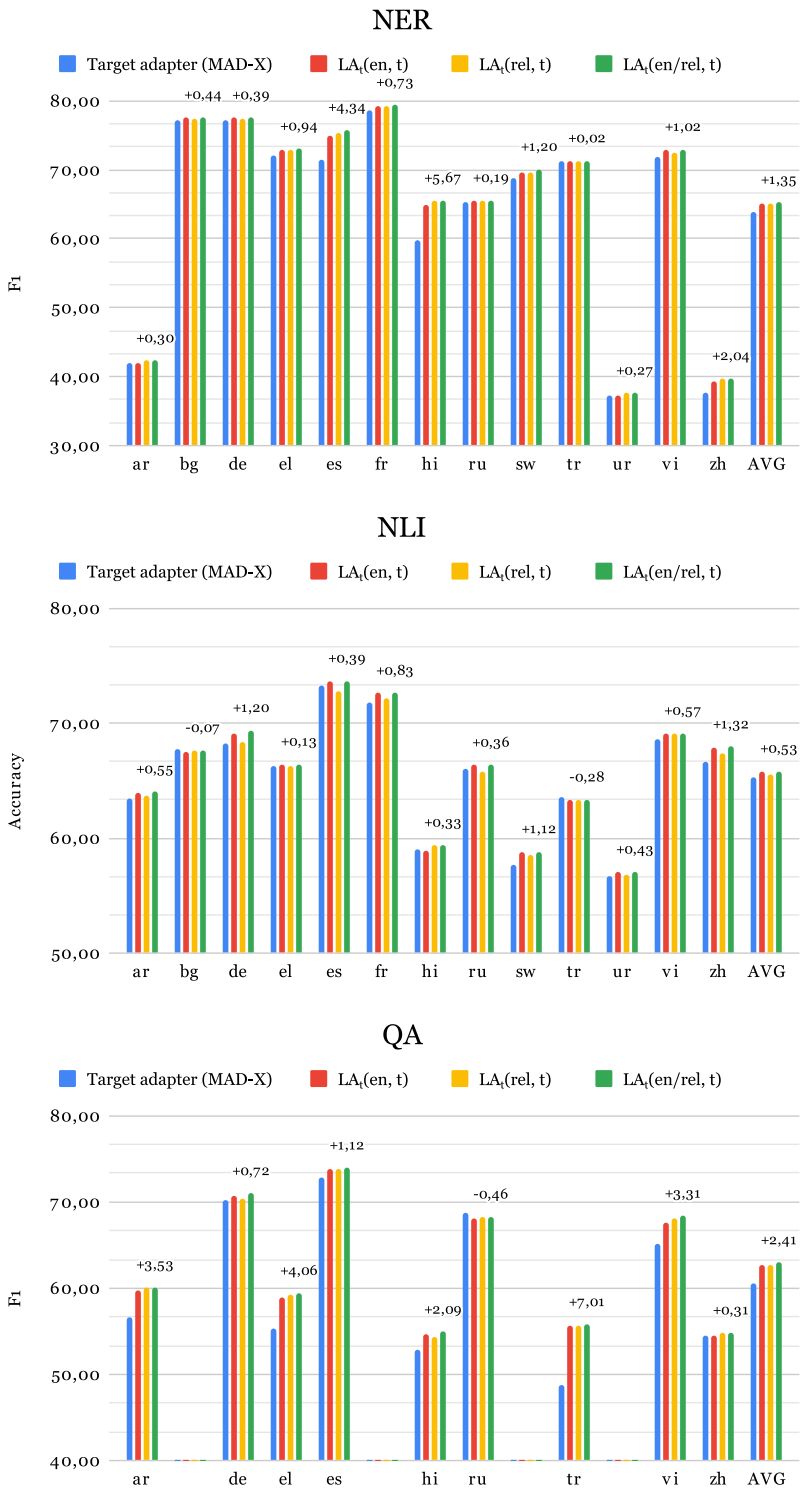


Figure 5.9.2. Variants of language arithmetic compared to the MAD-X method in the use-case to improve an existing target language adapter. The values above bars present a difference between a better LA setup and the MAD-X framework for the mBERT model.

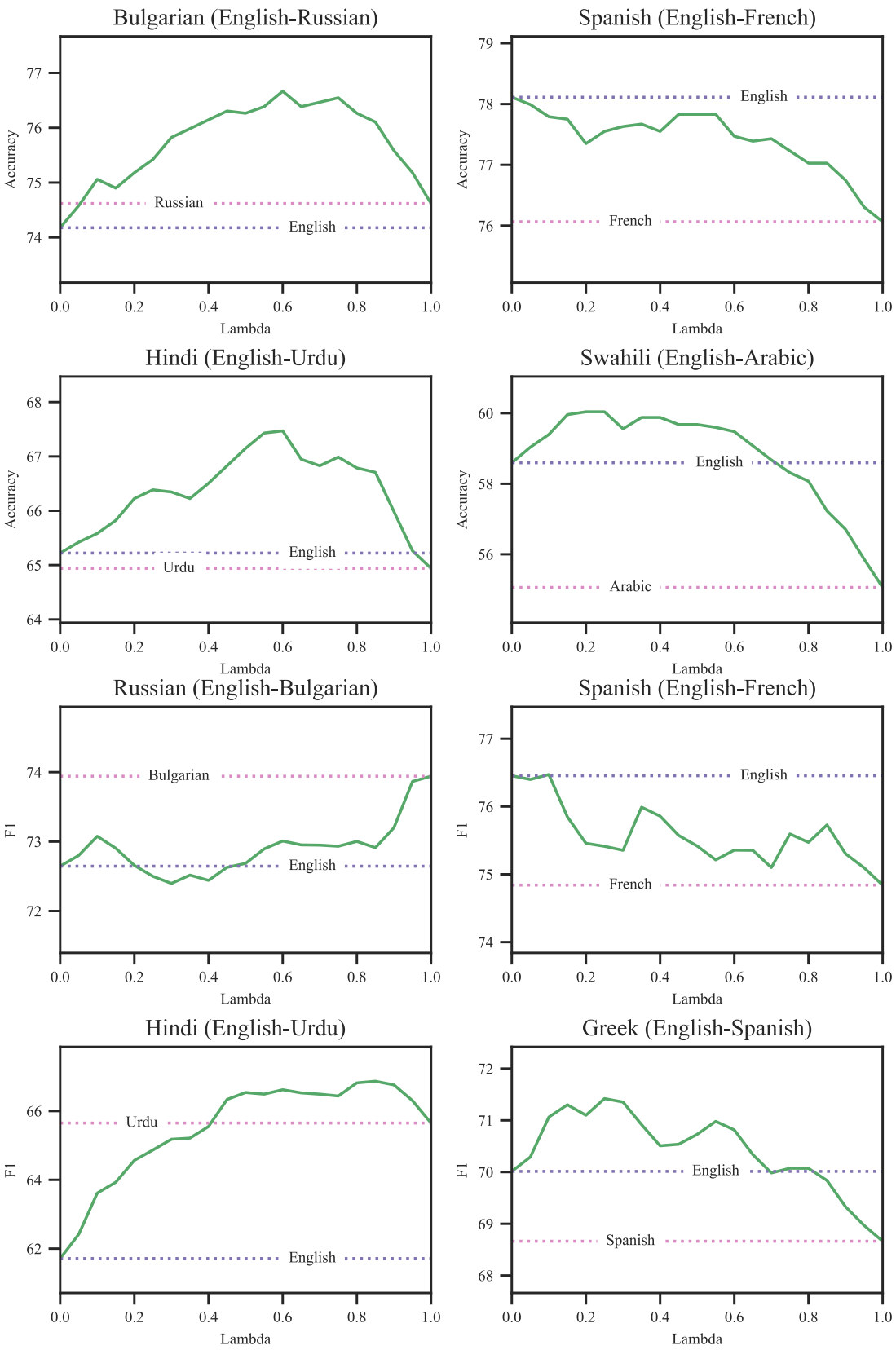


Figure 5.9.3. Interpolation of λ values for the zero-shot NLI and QA XLM-R scenario on the validation dataset. The horizontal dashed lines represent the baseline scores for both languages used in language arithmetic.

Table 5.9.2. Ties-Merging evaluation in the zero-shot setup on the NER task (XLM-R version, averaged over three runs and all evaluated languages). In the case of language arithmetic, where the language vectors have a higher overlap (i.e. higher cosine similarity), removing parameter interference decreases the overall performance.

Method	AVG F1 score
LA	60.54
Ties-Merging (Top-K% 20)	52.94
Ties-Merging (Top-K% 80)	57.57

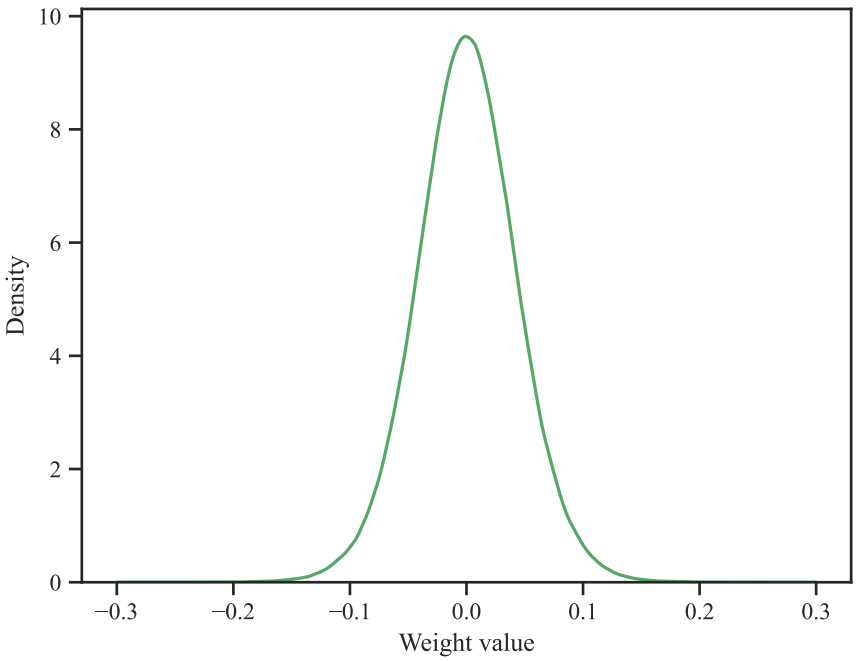


Figure 5.9.4. Language vectors, similar to task vectors, are extremely sparse. The kernel density estimate plot presents the weights of a Spanish mBERT adapter. The behaviour is consistent across sampled layers and languages.

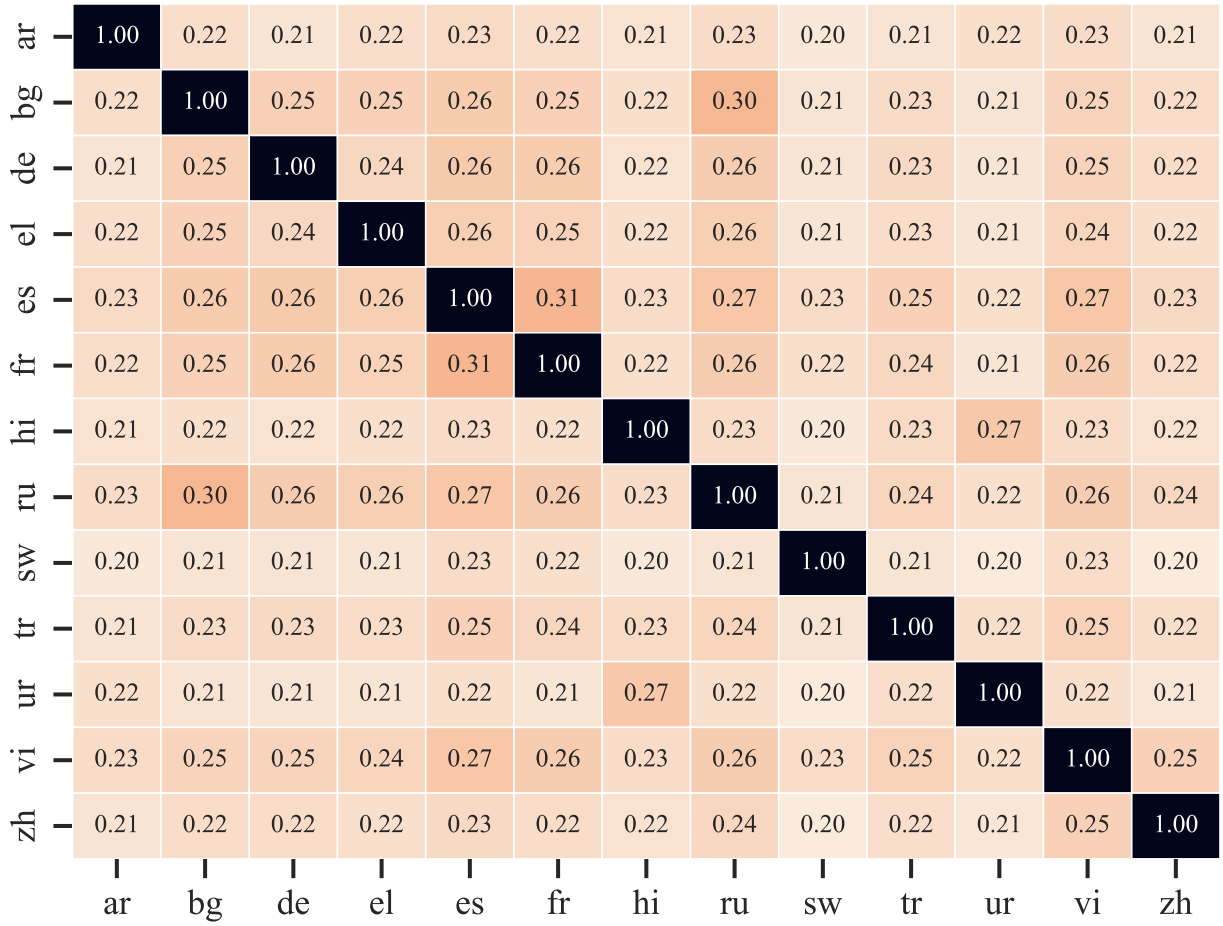


Figure 5.9.5. Cosine similarity between language vectors of language adapters.

Based on the cosine similarity observation, we investigated one of the recent task arithmetic extensions, Ties-Merging [190]. This work introduces a three-step algorithm that prevents different parameter interferences, improving upon task arithmetic. The algorithm *decreases the cosine similarity* via a pruning step and alignment of parameter signs to perform arithmetic only on relevant parameters to the merged tasks. On the experimental details note, as Ties-Merging operates on averaging, not addition, we utilise a different lambda range during validation (as suggested by Yadav et al. (2023) [190]), $\lambda \in [0.8, 1.8]$), and we set Top-K% to the default value of 20 and additionally to 80.

We report the comparison in the zero-shot setting on the NER task (XLM-R version) in Table 5.9.2. The Ties-Merging decreases the results significantly compared to the default language arithmetic. Moreover, we note that the pruning operation has the reverse effect; higher pruning (i.e. keeping Top-K% lower) decreases the performance (in contrast to task vectors) by making language vectors more sparse and, hence, closer to orthogonal.

One interpretation of the phenomena can be the different goals of the arithmetic: in the multi-task setup, we try to include multiple, often disconnected, tasks into a single task vector. In contrast, the language vectors' goal is to include the knowledge of the closely related language rather than remove the harmful artefacts. Our findings indicate that language arithmetic has different characteristics than task arithmetic, and the follow-up works that improve upon task arithmetic might not be suited for the multilingual context.

6. Is Modularity Transferable? A Case Study through the Lens of Knowledge Distillation

Title	Is Modularity Transferable? A Case Study through the Lens of Knowledge Distillation
Authors	Mateusz Klimaszewski, Piotr Andruszkiewicz, and Alexandra Birch
Conference	The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)
Year	2024

Abstract

The rise of Modular Deep Learning showcases its potential in various Natural Language Processing applications. Parameter-efficient fine-tuning (PEFT) modularity has been shown to work for various use cases, from domain adaptation to multilingual setups. However, all this work covers the case where the modular components are trained and deployed within one single Pre-trained Language Model (PLM). This model-specific setup is a substantial limitation on the very modularity that modular architectures are trying to achieve. We ask whether current modular approaches are transferable between models and whether we can transfer the modules from more robust and larger PLMs to smaller ones. In this work, we aim to fill this gap via a lens of Knowledge Distillation, commonly used for model compression, and present an extremely straightforward approach to transferring pre-trained, task-specific PEFT modules between same-family PLMs. Moreover, we propose a method that allows the transfer of modules between incompatible PLMs without any change in the inference complexity. The experiments on Named Entity Recognition, Natural Language Inference, and Paraphrase Identification tasks over multiple languages and PEFT methods showcase the initial potential of transferable modularity.

6.1. Introduction

Modular Deep Learning has recently garnered interest as a paradigm that builds upon the idea that a model is a combination of modules with control of the information flow. This paradigm allows for the transfer of learning from one task or language to another, compositionality of the modules and parameter efficiency [134]. For instance, modules allow for efficient (parameter-wise) fine-tuning of Large Language Models [84], enhance task-level generalisation [141], improve multilingual models [19], offer zero-shot capabilities [139] and enable cross-lingual [8] or cross-domain [92] knowledge transfer. Furthermore, repositories that store pre-trained modules like AdapterHub [133] promote the re-usability of previously trained components to new use cases.

The current modular approaches primarily focus on transferring knowledge to new languages, domains, or tasks. However, prior research assumes that the base model remains constant and overlooks the concept of *transferable modularity*, which entails the potential to transfer modules between different models. From a practical perspective, the effective utilisation of the *transferable modularity property* can reduce the computational burden, especially given the ongoing scaling of Large

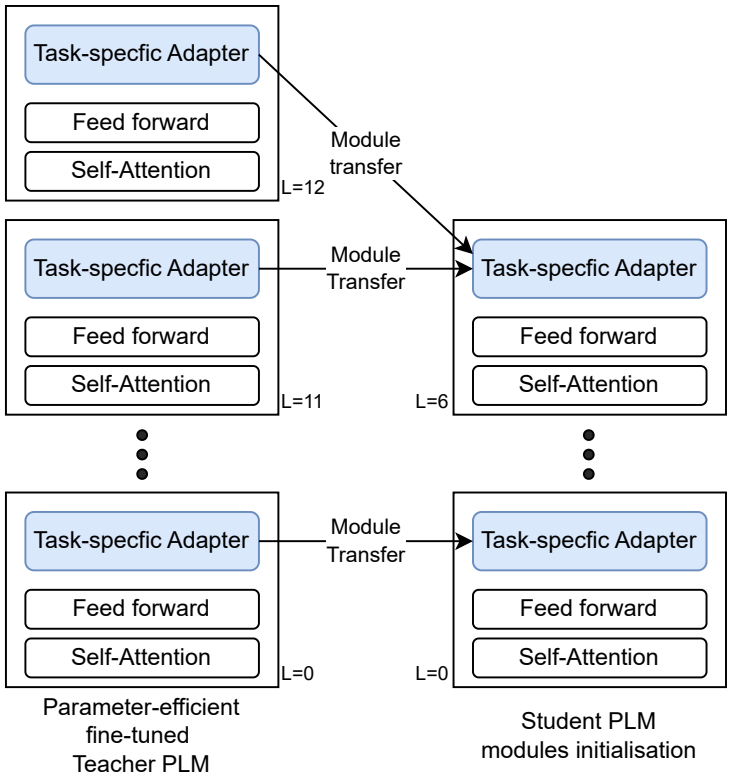


Figure 6.1.1. The most straightforward case of transferable modularity. The teacher model is first trained on a task using PEFT, e.g. Adapters, and then the student PEFT modules, prior to fine-tuning, are initialised with the teacher weights.

Language Models [30, 176], allowing for broader re-usability. Moreover, transferring modules from larger to smaller models can significantly enhance knowledge transfer. And finally, even the term "modularity" inherently implies the transfer property, suggesting that modular approaches should not be limited to a specific base model.

In this work, we aim to initialise the research objective of *transferable modularity*. We focus on a setup similar to Knowledge Distillation (KD) [79], i.e. where we have two differently sized PLMs (through the paper, we adopt the KD nomenclature, where the bigger model is called a teacher and the smaller - student). Unlike KD, we do not want to use the teacher model's output directly to train a student but use exclusively its fine-tuned PEFT modules.

We show that given matching PLMs (e.g. BERT [48] and DistilBERT [159]), it is possible to use pre-trained modules like Adapters [83, 132] or LoRA [84] as a better starting point for parameter-efficient (PE) fine-tuning of a smaller student PLM (see Figure 6.1.1). Moreover, we investigate a more challenging setup where the models are *incompatible*, i.e., have different internal dimensionality, and adapt modules via the proposed pruning and alignment method (without inference-time overhead).

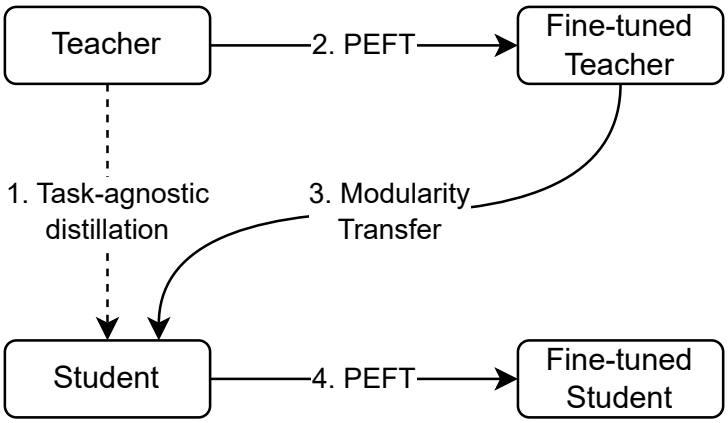


Figure 6.2.1. The schema of transferable modularity experiment. We investigate setups where the teacher-student pair result from task-agnostic distillation or are independently trained models.

To summarise, our contributions are as follows:[1]

- We define the property of transferable modularity.
- We investigate transferable modularity in matching and incompatible PLMs, proposing a pruning and alignment method for the latter.

6.2. Transferable Modularity

The high-level idea of our study is presented in Figure 6.2.1. Given a pair of PLMs, a teacher and a student, we aim to transfer the parameter-efficient (PE) modules from the teacher to the student. First, we use a PEFT technique to train the teacher and its PE modules. Then, we "move" the modules from the teacher and insert them into the student, followed by PEFT of the student. This approach means that PE modules of the student have non-random prior initialisation during training.

We consider two setups: (1) matching PLMs and (2) incompatible PLMs. The former uses a shallow version of a teacher with task-agnostic distillation as a student [88]. This case means that the models represent the same knowledge, have the same hidden dimensionality, and the only difference is the depth of the model. The latter represents a generalised version, where the models are differently parameterised (in terms of latent space size) and they are independently trained. We propose a parameter-free, sample-based pruning and alignment method to answer dimensionality mismatch.

[1] Code available at `https://github.com/mklimasz/transferable-modularity`

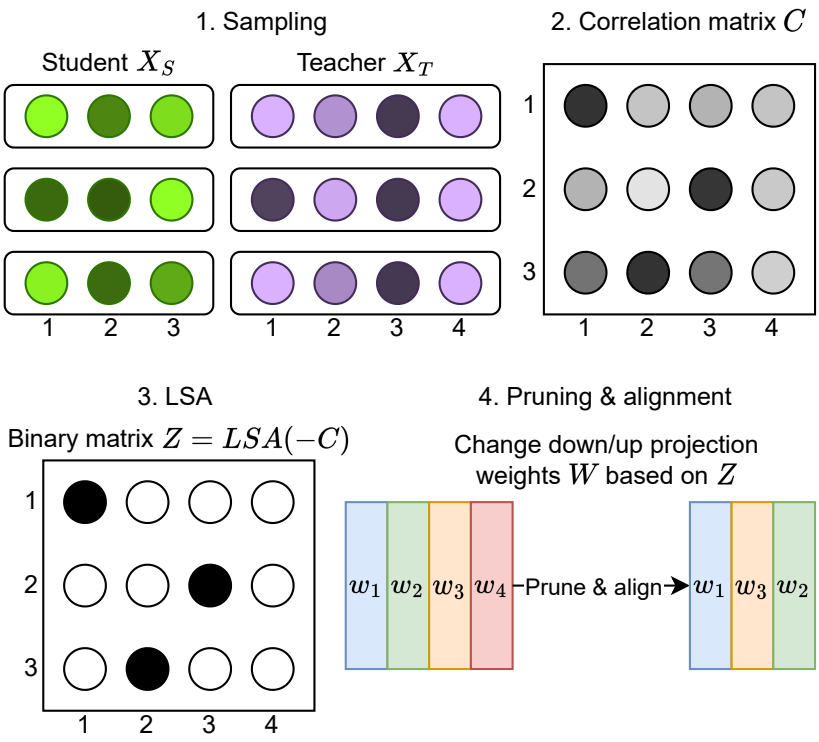


Figure 6.2.2. Toy example of adapting the PEFT modules in the case of mismatched dimensionality. Based on the sampled embeddings (1.), correlation matrix C is calculated (2.) and reduced via LSA to a binary matrix Z (3.). In the last step (4.), the pruning and alignment mapping function (derived from Z) is applied to down/up projection matrices of LoRA/Adapter modules and match dimensions.

6.2.1. Pruning and Alignment

In the case of incompatible PLMs, a dimensionality mismatch problem causes two main issues for transferable modularity. First, the module expects different (higher) dimensionality. Additionally, there exists an alignment discrepancy between the latent spaces of the two models, i.e. if the models have learned the same features, we do not have any guarantee of their placement in the latent space - their indices.

A crucial element of a successful Knowledge Distillation framework is the computational overhead; therefore, we propose an offline, parameter-free solution that does not change the final student model. The method presented in Figure 6.2.2 consists of four phases:

- sampling
- calculating correlation
- solving linear sum assignment (LSA) problem
- pruning & alignment

At first, we sample matching embeddings that would be an input to a PEFT module (we denote the set of embeddings X_s for student and X_t for teacher with

$x_s \in X_s$ and $x_t \in X_t$). We store embeddings per layer l (for clarity, we omit the notation of the layer).

In the next step, we establish a correlation matrix between latent spaces. We calculate Pearson's correlation coefficient matrix C. C_{ij} is a correlation between the i dimension of a x_s and the j dimension of a x_t embedding.

Given the correlation matrix, we attempt to find the best possible alignment. We define the problem as a linear sum assignment (LSA) [45] to establish the optimal mapping. As LSA calculates the minimum cost assignment, we use $-C$ as an input to the LSA algorithm. The algorithm produces a binary matrix Z where $Z_{ij} = 1$ means that the i index of X_s is mapped to j of X_t.

$$min \sum_i \sum_j (-C_{ij}) Z_{ij}$$

Finally, using the calculated assignment indices, we remove not-mapped weights from both down/up projection weights W of PEFT modules.

6.3. Experiments

6.3.1. Datasets

To evaluate our method, we benchmark it on three tasks: Named Entity Recognition (NER), Paraphrase Identification (PI) and Natural Language Inference (NLI) using multilingual datasets: WikiNeural [172], PAWS-X [193] and XNLI [43] covering jointly a set of over 20 languages[2].

6.3.2. Training Setup

We fine-tune multilingual models for each language/task pair using two PEFT methods: Adapter (architecture of Pfeiffer et al. (2021) [132], bottleneck size of 96) and LoRA (rank 8). We provide the training setup details for each dataset in Appendix 6.6.1.

For teacher-student pairs, we define two configurations:

- *matching*: multilingual BERT (mBERT[3], teacher) – multilingual DistilBERT (D'mBERT[4], student)

[2] Arabic, Bulgarian, Chinese, Dutch, English, French, German, Hindi, Italian, Japanese, Korean, Greek, Polish, Portuguese, Russian, Spanish, Swahili, Thai, Turkish, Urdu, Vietnamese

[3] `bert-base-multilingual-cased`

[4] `distilbert-base-multilingual-cased`

Table 6.3.1. Parameters, layer count and hidden dimension size of the evaluated models.

Model	Params	Layers	Hidden dim
D'mBERT	$135M$	6	768
mBERT	$178M$	12	768
XLM-R_{BASE}	$278M$	12	768
XLM-R_{LARGE}	$560M$	24	1024

- *incompatible*: XLM-RoBERTa Large (XLM-R_{LARGE}[5], teacher) – XLM-RoBERTa Base (XLM-R_{BASE}[6], student) [42]

We report the relevant hyper-parameters of the models in Table 6.3.1. As the models have mismatched layer counts, we test two approaches: skip modules (denoted SKIP, e.g., transfer every second module) or average them (denoted AVG, e.g., average the first and second layer's teacher module and transfer to the first module of a student).

6.3.3. Baselines and Metrics

For both *matching* and *incompatible* experiments, we define the following structure. As an upper bound of our evaluation, we provide the teacher results after PEFT (Step 2 in Figure 6.2.1). The baseline is a parameter-efficient fine-tuned student with default modules initialisation (i.e. omitting Step 3 in Figure 6.2.1).

We report F1 for NER and Accuracy for PI and NLI tasks with an average score over all languages in Section 6.4. The detailed per-language results are provided in Appendix 6.6.2.

6.4. Results and Discussion

6.4.1. Matching Models

Table 6.4.1 presents the results of the *matching* experiments. The prefix TM denotes the transfer modularity experiments. The initialisation of the modules transferred from the teacher PLM improved over a default initialisation on average in all the evaluated tasks. Moreover, the SKIP method presents consistency; the difference compared to the baseline was positive across most tasks and languages (88,7% cases). While at times the improvement was marginal (+0.02 gain in Swahili in NLI task), in most cases, as averages indicate, our approach significantly closes

[5] `xlm-roberta-large`
[6] `xlm-roberta-base`

Table 6.4.1. Results of the *matching* PLMs experiment. We report an average score (F1 or Accuracy) over all the datasets' languages and a relative performance gap to the teacher model.

	NER (F1)		PI (Acc)		NLI (Acc)	
	AVG	REL	AVG	REL	AVG	REL
			Adapter			
Teacher	95,35		82,60		67,98	
Student	92,94	−2,41	71,32	−11,28	62,12	−5,86
TM-Student$_{\text{AVG}}$	93,02	−2,32	72,96	−9,64	62,33	−5,65
TM-Student$_{\text{SKIP}}$	93,45	−1,90	75,11	−7,49	63,01	−4,97
			LoRA			
Teacher	93,27		74,68		63,00	
Student	90,09	−3,18	65,80	−8,88	60,56	−2,43
TM-Student$_{\text{AVG}}$	90,63	−2,64	68,52	−6,16	60,53	−2,47
TM-Student$_{\text{SKIP}}$	90,80	−2,47	70,69	−3,99	60,52	−2,47

Table 6.4.2. Results of the *incompatible* PLMs experiment.

	NER (F1)		PI (Acc)	
	AVG	REL	AVG	REL
			Adapter	
Teacher	95,34		88,81	
Student	93,30	−2,04	84,12	−4,69
TM-Student$_{\text{SKIP}}$	93,34	−2,00	84,27	−4,54
			LoRA	
Teacher	93,64		87,03	
Student	90,83	−2,82	78,72	−8,31
TM-Student$_{\text{SKIP}}$	90,84	−2,80	78,64	−8,39

the gap to the teacher model (e.g. +4 point improvement in Korean on PAWS-X datasets using Adapter or over +2 in Spanish LoRA on XNLI). SKIP struggles to outperform the baseline exclusively on XNLI when using LoRA. The results are on par; however, even the teacher models struggle with the task, and the knowledge that can be transferred is relatively limited.

The SKIP outperforms AVG across all the experiments. Considering the results and the findings of van Aken et al. (2019) [178] indicating that the Transformer-based models have internal modularity and each layer has its own defined task, we hypothesise that the averaging might not reflect these phenomena. Therefore, in the *incompatible* experiment, we evaluated just the SKIP method.

6.4.2. Incompatible Models

We present the results of the evaluation in Table 6.4.2. In the case of non-distilled PLMs, the TM method does not significantly outperform the baseline. The changes are uneven; while the transfer shows improvement up to almost +2 points in Korean PAWS-X, it can also decrease the performance as in French PAWS-X, losing −1.05.

The disparity between *matching* and *incompatible* experiments can be attributed to alignment challenges. Models subjected to distillation exhibit reliable alignment, thanks to the inclusion of an auxiliary loss term such as the cosine embedding loss [159] in the task-agnostic distillation process. In contrast, the correlation-based method encounters difficulties when dealing with models of greater depth. Notably, the LSA algorithm yields lower scores for deeper layers. Considering the different representations required for each language and task pair, this outcome implies that independently trained models require more robust alignment techniques to ensure consistent modularity transfer across all encoded features.

6.5. Conclusions

In this work, we present a case study of transferable modularity property. We evaluate current modular techniques in two scenarios: (1) *matching*, where a student is a shallow, task-agnostic distillation of the teacher and (2) *incompatible*, where a student is independently trained, a shallower model with mismatched internal dimensionality.

The results show that the current modular approach can be transferable as the modules from a matching teacher improve the PEFT of a student model. However, when a student is not distilled from the teacher, the evaluated techniques are inconsistent under the transfer condition, showing the limitation of the current

modular methods. We hope this study will inspire future work on modular techniques to consider the transferable modularity property under a more challenging incompatible models scenario.

6.6. Appendix

6.6.1. Experimental Setup

We use the AdapterHub library [133] for all our experiments. We train all our models using a single GPU with a batch size of 64 and a learning rate of 1e–5 for 10 epochs for NER & NLI tasks and 30 epochs for the PI task. We choose the final checkpoint based on validation dataset performance.

For PEFT hyper-parameters, we set the bottleneck size to 96 for Adapter modules and a rank of 8 for LoRA. We apply LoRA to the query and value self-attention modules.

6.6.2. Per Language Results

In Tables 6.6.1, 6.6.2 and 6.6.3, we expand the results reported in Tables 6.4.1 and 6.4.2 and provide the scores for each evaluated language.

Table 6.6.1. Named Entity Recognition results per language.

Model	de	en	es	fr	it	nl	pl	pt	ru
					Matching PLMs				
					Adapter				
Teacher	97,57	92,79	98,08	95,49	94,85	97,74	95,54	95,91	90,15
Student	95,34	90,23	95,81	93,23	92,67	95,27	93,73	94,21	85,97
TM-Student$_{\text{AVG}}$	95,53	90,21	95,87	93,26	92,74	95,39	93,88	94,40	85,92
TM-Student$_{\text{SKIP}}$	95,88	90,65	96,24	93,77	93,13	95,87	94,20	94,72	86,57
					Lora				
Teacher	95,78	90,49	96,45	93,26	93,13	95,94	93,97	94,41	85,97
Student	92,45	87,25	93,55	90,06	89,95	92,85	91,55	92,15	80,96
TM-Student$_{\text{AVG}}$	92,92	87,74	93,94	90,57	90,75	93,24	91,97	92,52	82,03
TM-Student$_{\text{SKIP}}$	93,03	87,99	93,87	90,88	91,04	93,44	92,04	92,61	82,27
					Incompatible PLMs				
					Adapter				
Teacher	97,36	92,30	97,95	95,61	94,99	97,79	96,15	96,12	89,74
Student	95,20	89,92	96,19	93,34	93,06	96,29	94,14	94,56	86,99
TM-Student$_{\text{SKIP}}$	95,30	90,03	96,21	93,40	93,00	96,09	94,19	94,78	87,01
					Lora				
Teacher	94,68	89,94	96,19	92,35	92,85	95,67	93,82	94,11	85,09
Student	92,21	86,65	92,74	89,40	90,05	93,14	91,61	92,25	81,62
TM-Student$_{\text{SKIP}}$	92,10	86,65	93,00	89,61	90,01	93,01	91,51	92,32	81,69

Table 6.6.2. Paraphrase Identification results per language.

Model	de	en	es	fr	ja	ko	zh
				Matching PLMs			
				Adapter			
Teacher	83,60	91,60	85,20	86,90	76,05	75,95	78,90
Student	73,30	75,85	72,90	74,65	67,25	65,25	70,05
TM-Student$_{\text{AVG}}$	74,15	82,35	73,35	75,10	67,10	67,40	71,25
TM-Student$_{\text{SKIP}}$	74,50	85,05	77,85	78,15	69,10	69,25	71,85
				Lora			
Teacher	75,10	83,30	78,70	77,75	67,85	67,95	72,10
Student	70,20	63,45	67,30	70,10	62,80	61,85	64,90
TM-Student$_{\text{AVG}}$	71,95	69,35	69,95	71,85	64,75	64,05	67,75
TM-Student$_{\text{SKIP}}$	72,25	74,50	72,30	74,85	66,70	64,90	69,30
				Incompatible PLMs			
				Adapter			
Teacher	90,45	94,70	91,20	92,15	82,35	85,00	85,80
Student	85,75	92,55	87,25	89,25	77,10	75,65	81,30
TM-Student$_{\text{SKIP}}$	86,15	92,05	88,50	88,20	76,80	77,45	80,75
				Lora			
Teacher	89,40	93,80	89,90	89,65	80,95	81,45	84,05
Student	80,00	88,05	82,95	83,55	72,55	68,60	75,35
TM-Student$_{\text{SKIP}}$	80,95	88,05	82,10	82,70	71,65	69,80	75,20

Table 6.6.3. Natural Language Inference results per language for the *matching* PLMs experiment.

Model	ar	bg	de	el	en	es	fr	hi	ru	sw	th	tr	ur	vi	zh
								Adapter							
Teacher	65,53	70,18	70,12	68,08	77,03	73,01	72,00	63,39	69,58	60,12	60,40	67,49	60,40	71,28	71,10
Student	60,12	63,47	65,07	63,35	69,62	66,11	65,89	57,33	62,14	56,99	56,43	61,86	55,71	63,67	64,01
TM-Student$_{\text{AVG}}$	60,54	63,57	65,19	63,27	70,38	66,69	66,13	57,54	62,53	56,17	56,39	62,00	56,59	63,33	64,59
TM-Student$_{\text{SKIP}}$	61,16	64,23	65,27	63,49	70,58	68,16	66,53	58,48	63,45	57,01	57,05	62,61	57,47	63,89	65,73
								Lora							
Teacher	61,46	63,79	66,83	63,77	70,12	67,84	66,99	59,90	64,47	54,57	55,37	61,60	56,47	65,01	66,77
Student	59,20	62,40	63,43	61,96	67,15	64,09	63,87	57,56	60,62	54,87	53,67	59,84	55,97	61,74	62,08
TM-Student$_{\text{AVG}}$	59,20	62,40	63,43	61,96	67,15	64,09	63,87	57,56	60,62	54,87	53,67	59,72	55,77	61,74	61,88
TM-Student$_{\text{SKIP}}$	59,10	62,36	63,75	61,66	67,19	64,11	64,19	57,56	60,94	54,37	53,19	59,76	56,11	61,30	62,26

7. Future work and conclusions

7.1. Future work and open research problems

7.1.1. Multilingual Large Language Models

The natural extension of the work presented in the thesis, when it comes to multilinguality, would be to expand the work beyond PLMs and apply it to multilingual LLMs. With the recent rise of high-performing and open multilingual LLMs that match the closed-sourced counterparts (e.g. EuroLLM [116, 115] – co-authored by the PhD candidate or Salamandra [64]),[1] such a study would be an appealing future step. Furthermore, considering the costs and complexity of training such LLMs, extending the model to support additional languages via model merging (i.e. continual learning scenario [183]) sounds attractive. As the multilingual LLMs support just a fraction of existing languages [70], cheaper extension procedures would drastically reduce the computational burden.

7.1.2. Language arithmetic grounding

There have been numerous successful applications of model merging in different areas. However, the grounding in the theory and understanding of all the whys and hows have been a different story. Initial studies tried to unravel the mystery in a multi-task setup [46]; however, our findings on the difference between task and language arithmetic (Section 5.9.5) suggest that depending on the model merging application, there may be a need for a deeper analysis of the phenomena.

7.1.3. Multimodal Language Models

While in this thesis, we focus on text-based approaches to NLP, there have been rapid advancements on the border of Natural Language Processing and other machine learning fields, i.e. Computer Vision and Speech Processing. Adaptation of

[1] We want to denote the existence of BLOOM [160] and BLOOMZ [118]; despite subpar performance compared to closed-source counterparts (due to the complexity and pioneering nature of these projects), the models opened a door for future multilingual LLMs.

LLMs to new modalities allowed the building of Visual LLMs and Speech LLMs, e.g. Molmo [47], Qwen-VL [15, 184, 16], SpiRit-LM [120] or SPIRE [6]. Multimodality could be another aspect evaluated by the modular approaches, combining information signals from multiple sources: text, images, videos or speech.

7.2. Conclusions

This thesis presents our contribution to modular approaches for Natural Language Processing tasks that target multiple objectives simultaneously. Through the series of publications, we investigated existing aspects: multi-task, multi-domain and multilingual setups. We developed a multi-task system COMBO for morphosyntactic analysis tasks. The system allows the end-user to control the system objectives. For the multi-domain scenario, we proposed Gated Adapters that offer knowledge transfer between domain-specific adapter modules, evaluated via machine translation task. Further, we introduce language arithmetic as an extension of model merging for multilinguality. Finally, in the fourth contribution, we proposed and evaluated a novel aspect: multi-model that stands as a new challenge for existing MDL methods.

Bibliography

[1] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan M. Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340b technical report. *CoRR*, abs/2406.11704, 2024.

[2] Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. Give your text representation models some love: the case for Basque. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France, May 2020. European Language Resources Association.

[3] Divyanshu Aggarwal, Vivek Gupta, and Anoop Kunchukuttan. IndicXNLI: Evaluating multilingual inference for Indian languages. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10994–11006, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational

Linguistics.

[4] Roee Aharoni and Yoav Goldberg. Unsupervised domain clusters in pretrained language models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online, July 2020. Association for Computational Linguistics.

[5] Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. Tower: An open multilingual large language model for translation-related tasks. In *First Conference on Language Modeling*, 2024.

[6] Kshitij Ambilduke, Ben Peters, Sonal Sannigrahi, Anil Keshwani, Tsz Kin Lam, Bruno Martins, Marcely Zanon Boito, and André F. T. Martins. From TOWER to SPIRE: adding the speech modality to a text-only LLM. *CoRR*, abs/2503.10620, 2025.

[7] Alan Ansell, Marinela Parović, Ivan Vulić, Anna Korhonen, and Edoardo Ponti. Unifying cross-lingual transfer across scenarios of resource scarcity. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3980–3995, Singapore, December 2023. Association for Computational Linguistics.

[8] Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. Composable sparse fine-tuning for cross-lingual transfer. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[9] Alan Ansell, Ivan Vulic, Hannah Sterz, Anna Korhonen, and Edoardo M. Ponti. Scaling sparse fine-tuning to large language models. *CoRR*, abs/2401.16405, 2024.

[10] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for Arabic language understanding. In Hend Al-Khalifa, Walid Magdy, Kareem Darwish, Tamer Elsayed, and Hamdy Mubarak, editors, *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May 2020. European Language Resource Association.

[11] Maria Jesus Aranzabe, Aitziber Atutxa, Kepa Bengoetxea, Arantza Diaz de Ilarraza, Iakes Goenaga, Koldo Gojenola, and Larraitz Uria. Automatic Conversion of the Basque Dependency Treebank to Universal Dependencies.

In *Proceedings of the 14th International Workshop on Treebanks and Linguistic Theories (TLT14)*, pages 233–241, 2015.

[12] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July 2020. Association for Computational Linguistics.

[13] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[15] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966, 2023.

[16] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025.

[17] Carliss Y. Baldwin and Kim B. Clark. *Design Rules, Volume 1: The Power of Modularity*. The MIT Press, 03 2000.

[18] Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online, July 2020. Association for Computational Linguistics.

[19] Ankur Bapna and Orhan Firat. Simple, scalable adaptation for neural machine translation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural*

Language Processing (EMNLP-IJCNLP), pages 1538–1548, Hong Kong, China, November 2019. Association for Computational Linguistics.

[20] Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2020 conference on machine translation (WMT20). In Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors, *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online, November 2020. Association for Computational Linguistics.

[21] Christos Baziotis, Mikel Artetxe, James Cross, and Shruti Bhosale. Multilingual machine translation with hyper-adapters. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1170–1185, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[22] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[23] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[24] Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. SICK through the SemEval Glasses. Lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Journal of Language Resources and Evaluation*, 50:95–124, 2016.

[25] Christopher M. Bishop and Hugh Bishop. *The Deep Learning Revolution*, pages 1–22. Springer International Publishing, Cham, 2024.

[26] Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte M. Alves, André F. T.

Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El Haddad, Manuel Faysse, Maxime Peyrard, Nuno Miguel Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. Eurobert: Scaling multilingual encoders for european languages. *CoRR*, abs/2503.05500, 2025.

[27] Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Connallen, and Kelli A. Houston. Object-oriented analysis and design with applications, third edition. *SIGSOFT Softw. Eng. Notes*, 33(5), August 2008.

[28] Gosse Bouma, Djamé Seddah, and Daniel Zeman. From raw text to enhanced Universal Dependencies: The parsing shared task at IWPT 2021. In Stephan Oepen, Kenji Sagae, Reut Tsarfaty, Gosse Bouma, Djamé Seddah, and Daniel Zeman, editors, *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 146–157, Online, August 2021. Association for Computational Linguistics.

[29] Denny Britz, Quoc Le, and Reid Pryzant. Effective domain mixing for neural machine translation. In Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer, editors, *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[30] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[31] Lucas Caccia, Edoardo Ponti, Zhan Su, Matheus Pereira, Nicolas Le Roux, and Alessandro Sordoni. Multi-head adapter routing for cross-task generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[32] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904,

Online, November 2020. Association for Computational Linguistics.

[33] Minghui Chen, Meirui Jiang, Xin Zhang, Qi Dou, Zehua Wang, and Xiaoxiao Li. Local superior soups: A catalyst for model merging in cross-silo federated learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 20858–20886. Curran Associates, Inc., 2024.

[34] Shutong Chen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Fedmerge: Federated personalization via model merging. *CoRR*, abs/2504.06768, 2025.

[35] Rochelle Choenni, Dan Garrette, and Ekaterina Shutova. Cross-lingual transfer with language-specific subnetworks for low-resource dependency parsing. *Computational Linguistics*, pages 613–641, September 2023.

[36] Christos Christodouloupoulos and Mark Steedman. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395, Jun 2015.

[37] Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[38] Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. Language-family adapters for low-resource multilingual neural machine translation. In Atul Kr. Ojha, Chao-hong Liu, Ekaterina Vylomova, Flammie Pirinen, Jade Abbott, Jonathan Washington, Nathaniel Oco, Valentin Malykh, Varvara Logacheva, and Xiaobing Zhao, editors, *Proceedings of the The Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023)*, pages 59–72, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[39] Y. J. Chu and T. H. Liu. On the Shortest Arborescence of a Directed Graph. *Science Sinica*, 14:1396–1400, 1965.

[40] Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. Building Universal Dependency treebanks in Korean. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language

Resources Association (ELRA).

[41] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, André F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. Saullm-7b: A pioneering large language model for law. *CoRR*, abs/2403.03883, 2024.

[42] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[43] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[44] Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672, 2022.

[45] David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

[46] Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*, 2024.

[47] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca

Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross B. Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *CoRR*, abs/2409.17146, 2024.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[49] Adam Dobrowolski, Mateusz Klimaszewski, Adam Myśliwy, Marcin Szymański, Jakub Kowalski, Kornelia Szypuła, Paweł Przewłocki, and Paweł Przybysz. Samsung R&D institute Poland participation in WMT 2022. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 251–259, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[50] Timothy Dozat and Christopher D. Manning. Simpler but more accurate semantic dependency parsing. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[51] Dheeru Dua, Shruti Bhosale, Vedanuj Goswami, James Cross, Mike Lewis, and Angela Fan. Tricks for training sparse translation models. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors,

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3340–3345, Seattle, United States, July 2022. Association for Computational Linguistics.

[52] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

[53] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China, July 2015. Association for Computational Linguistics.

[54] Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI 2020*, pages 2006–2013. IOS Press, 2020.

[55] Markus Eberts and Adrian Ulges. An end-to-end model for entity-level relation extraction using multi-instance learning. In Paola Merlo, Jorg Tiedemann,

and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3650–3660, Online, April 2021. Association for Computational Linguistics.

[56] Jack Edmonds. Optimum Branchings. *Journal of Research of the National Bureau of Standards*, 71B(4):233–240, 1967.

[57] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[58] Negar Foroutan, Mohammadreza Banaei, Rémi Lebret, Antoine Bosselut, and Karl Aberer. Discovering language-neutral sub-networks in multilingual language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7575, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[59] Markus Freitag and Yaser Al-Onaizan. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897, 2016.

[60] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1027–1035, Red Hook, NY, USA, 2016. Curran Associates Inc.

[61] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In Eunjeong L. Park, Masato Hagiwara, Dmitrijs Milajevs, and Liling Tan, editors, *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[62] Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. *CoRR*, abs/2412.00081, 2024.

[63] Goran Glavaš and Ivan Vulić. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online, April 2021. Association for Computational Linguistics.

[64] Aitor Gonzalez-Agirre, Marc Pàmies, Joan Llop, Irene Baucells, Severino Da Dalt, Daniel Tamayo, José Javier Saiz, Ferran Espuña, Jaume Prats,

Javier Aula-Blasco, Mario Mina, Iñigo Pikabea, Adrián Rubio, Alexander Shvets, Anna Salles, Iñaki Lacunza, Jorge Palomar, Júlia Falcão, Lucía Tormo-Bañuelos, Luis Vasquez-Reina, Montserrat Marimon, Oriol Pareras, Valle Ruíz-Fernández, and Marta Villegas. Salamandra technical report. *CoRR*, abs/2502.08489, 2025.

[65] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[66] Alex Graves and Jürgen Schmidhuber. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18(5):602–610, 2005.

[67] Demi Guo, Alexander Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 4884–4896, Online, August 2021. Association for Computational Linguistics.

[68] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[69] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

[70] Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732, September 2022.

[71] Stanford HAI. Introducing the center for research on foundation models (crfm). `https://hai.stanford.edu/news/introducing-center-research-foundation-models-crfm`, 2021. Accessed: (09-04-2025).

[72] Jan Hajič, Otakar Smrž, Petr Zemánek, Petr Pajas, Jan Šnaidauf, Emanuel

Beška, Jakub Kracmar, and Kamila Hassanová. Prague arabic dependency treebank 1.0, 2009. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[73] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024.

[74] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[75] Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. Building the essential resources for finnish: The turku dependency treebank. *Lang. Resour. Eval.*, 48(3):493–531, September 2014.

[76] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[78] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023.

[79] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

[80] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. Aspect-based sentiment analysis using BERT. In Mareike Hartmann and Barbara Plank, editors, *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196, Turku, Finland, September–October 2019. Linköping University Electronic Press.

[81] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[82] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. `https:`

//doi.org/10.5281/zenodo.1212303, 2020.

[83] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019.

[84] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[85] Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*, 2024.

[86] Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[87] Kiyoung Kim. Pretrained Language Models For Korean. https://github.com/kiyoungkim1/LMkor, 2020.

[88] Young Jin Kim and Hany Hassan. FastFormers: Highly efficient transformer models for natural language understanding. In Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavaš, Shafiq Joty, Alex Wang, and Thomas Wolf, editors, *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 149–158, Online, November 2020. Association for Computational Linguistics.

[89] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[90] Mateusz Klimaszewski, Piotr Andruszkiewicz, and Alexandra Birch. Is modularity transferable? a case study through the lens of knowledge distillation. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9352–9360, Torino, Italia, May 2024. ELRA and ICCL.

[91] Mateusz Klimaszewski, Piotr Andruszkiewicz, and Alexandra Birch. No train but gain: Language arithmetic for training-free language adapters

enhancement. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11121–11134, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[92] Mateusz Klimaszewski, Zeno Belligoli, Satendra Kumar, and Emmanouil Stergiadis. Gated adapters for multi-domain neural machine translation. In *ECAI 2023 - 26th European Conference on Artificial Intelligence*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 1264–1271. IOS Press, 2023.

[93] Mateusz Klimaszewski and Alina Wróblewska. COMBO: A new module for EUD parsing. In Stephan Oepen, Kenji Sagae, Reut Tsarfaty, Gosse Bouma, Djamé Seddah, and Daniel Zeman, editors, *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 158–166, Online, August 2021. Association for Computational Linguistics.

[94] Mateusz Klimaszewski and Alina Wróblewska. COMBO: State-of-the-art morphosyntactic analysis. In Heike Adel and Shuming Shi, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 50–62, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[95] Catherine Kobus, Josep Crego, and Jean Senellart. Domain control for neural machine translation. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria, September 2017. INCOMA Ltd.

[96] Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In Loic Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors, *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online, November 2021. Association for Computational Linguistics.

[97] Tom Kocmi, Dominik Macháček, and Ondřej Bojar. *The Reality of Multi-Lingual Machine Translation*, volume 21 of *Studies in Computational*

and Theoretical Linguistics. Institute of Formal and Applied Linguistics, Prague, Czechia, 2021.

[98] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In Thang Luong, Alexandra Birch, Graham Neubig, and Andrew Finch, editors, *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics.

[99] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[100] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[101] Neal Lawton, Anoop Kumar, Govind Thattai, Aram Galstyan, and Greg Ver Steeg. Neural architecture search for parameter-efficient fine-tuning of large pre-trained language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8506–8515, Toronto, Canada, July 2023. Association for Computational Linguistics.

[102] Jaeseong Lee, Seung-won Hwang, and Taesup Kim. FAD-X: Fusing adapters for cross-lingual transfer to low-resource languages. In Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 57–64, Online only, November 2022. Association for Computational Linguistics.

[103] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[104] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts

for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics.

[105] Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside BERT's linguistic knowledge. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors, *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy, August 2019. Association for Computational Linguistics.

[106] Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain, April 2017. Association for Computational Linguistics.

[107] Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[108] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[109] Marco Lui and Timothy Baldwin. langid.py: An off-the-shelf language identification tool. In Min Zhang, editor, *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[110] Chaitanya Malaviya, Graham Neubig, and Patrick Littell. Learning language representations for typology prediction. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[111] Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartlomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. *CoRR*,

abs/2502.04959, 2025.

[112] Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 379–395, Cham, 2025. Springer Nature Switzerland.

[113] Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière. Weighted ensemble models are strong continual learners. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 306–324, Cham, 2025. Springer Nature Switzerland.

[114] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July 2020. Association for Computational Linguistics.

[115] Pedro Henrique Martins, João Alves, Patrick Fernandes, Nuno M. Guerreiro, Ricardo Rei, Amin Farajian, Mateusz Klimaszewski, Duarte M. Alves, José Pombal, Manuel Faysse, Pierre Colombo, François Yvon, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. Eurollm-9b: Technical report, 2025.

[116] Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G.C. de Souza, Alexandra Birch, and André F.T. Martins. Eurollm: Multilingual language models for europe. *Procedia Computer Science*, 255:53–62, 2025. Proceedings of the Second EuroHPC user day.

[117] Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. HerBERT: Efficiently pretrained transformer-based language model for Polish. In Bogdan Babych, Olga Kanishcheva, Preslav Nakov, Jakub Piskorski, Lidia Pivovarova, Vasyl Starko, Josef Steinberger, Roman Yangarber, Michał Marcińczuk, Senja Pollak, Pavel Přibáň, and Marko Robnik-Šikonja, editors, *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine, April 2021. Association for Computational Linguistics.

[118] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hai-

ley Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada, July 2023. Association for Computational Linguistics.

[119] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[120] Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoît Sagot, and Emmanuel Dupoux. SpiRit-LM: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.

[121] Debora Nozza, Federico Bianchi, and Dirk Hovy. What the [mask]? making sense of language-specific BERT models. *CoRR*, abs/2003.02912, 2020.

[122] Oleksiy Ostapenko, Zhan Su, Edoardo Maria Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. Towards modular llms by building and reusing a library of loras. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[123] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[124] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[125] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[126] Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799, Seattle, United States, July 2022. Association for Computational Linguistics.

[127] Marinela Parović, Ivan Vulić, and Anna Korhonen. Investigating the potential of task arithmetic for cross-lingual transfer. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 124–137, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[128] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[129] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[130] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[131] Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. Lifting the curse of multilinguality by pre-training modular transformers. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3479–3495, Seattle, United States, July 2022. Association for Computational Linguistics.

[132] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online, April 2021. Association for Computational Linguistics.

[133] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapting transformers. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online, October 2020. Association for Computational Linguistics.

[134] Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Ponti. Modular deep learning. *Transactions on Machine Learning Research*, 2023. Survey Certification.

[135] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online, November 2020. Association for Computational Linguistics.

[136] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. UNKs everywhere: Adapting multilingual language models to new scripts. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[137] Minh Quang Pham, Josep Maria Crego, François Yvon, and Jean Senellart. A study of residual adapters for multi-domain neural machine translation. In Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors, *Proceedings of the Fifth Conference on Machine Translation*, pages 617–628, Online, November

2020. Association for Computational Linguistics.

[138] MinhQuang Pham, Josep Maria Crego, and François Yvon. Revisiting multi-domain machine translation. *Transactions of the Association for Computational Linguistics*, 9:17–35, 2021.

[139] Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. Monolingual adapters for zero-shot neural machine translation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online, November 2020. Association for Computational Linguistics.

[140] Maciej Piasecki, Agnieszka Dziob, Arkadiusz Janz, Jan Kocoń, Tomasz Naskrt, Marcin Oleksy, Ewa Rudnicka, Tomasz Walkowiak, Jan Wieczorek, and Krzysztof Hwaszcz. Clarin-pl: a user centred language technology infrastructure. *Language Resources and Evaluation*, May 2025.

[141] Edoardo Maria Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. Combining parameter-efficient modules for task-level generalisation. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 687–702, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[142] Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1):4381, Sep 2020.

[143] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[144] Matt Post. A call for clarity in reporting BLEU scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[145] Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon

Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. Adapters: A unified library for parameter-efficient and modular transfer learning. In Yansong Feng and Els Lefever, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, Singapore, December 2023. Association for Computational Linguistics.

[146] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online, July 2020. Association for Computational Linguistics.

[147] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. `https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf`, 2018. Accessed: (21-05-2025).

[148] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. `https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`, 2019. Accessed: (21-05-2025).

[149] Afshin Rahimi, Yuan Li, and Trevor Cohn. Massively multilingual transfer for NER. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy, July 2019. Association for Computational Linguistics.

[150] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[151] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digital Medicine*, 4, 2020.

[152] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[153] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics.

[154] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online, November 2020. Association for Computational Linguistics.

[155] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. AdapterDrop: On the efficiency of adapters in transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[156] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[157] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In Anoop Sarkar and Michael Strube, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[158] Piotr Rybak and Alina Wróblewska. Semi-supervised neural system for tagging, parsing and lematization. In Daniel Zeman and Jan Hajič, editors, *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 45–54, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[159] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[160] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel

Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022.

[161] Stefan Schweter. BERTurk - BERT models for Turkish. `https://doi.org/10.5281/zenodo.3770924`, April 2020.

[162] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[163] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. A gold standard dependency corpus for English. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[164] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[165] Emmanouil Stergiadis, Satendra Kumar, Fedor Kovalev, and Pavel Levin. Multi-domain adaptation in neural machine translation through multidimensional tagging. In Janice Campbell, Ben Huyck, Stephen Larocca, Jay Marciano, Konstantin Savenkov, and Alex Yanishevsky, editors, *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, pages 396–420, Virtual, August 2021. Association for Machine Translation in the Americas.

[166] George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations*, 2024.

[167] Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In

Daniel Zeman and Jan Hajič, editors, *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[168] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[169] Umut Sulubacak, Memduh Gokirmak, Francis Tyers, Çağrı Çöltekin, Joakim Nivre, and Gülşen Eryiğit. Universal Dependencies for Turkish. In Yuji Matsumoto and Rashmi Prasad, editors, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3444–3454, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[170] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[171] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[172] Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[173] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019.

[174] Mbugua Samuel Thaiya, Korongo Julia, and Samuel Mbugua. On software modular architecture: Concepts, metrics and trends. *International Journal of*

Computer & Organization Trends, 2022.

[175] Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[176] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[177] Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. UDapter: Language adaptation for truly Universal Dependency parsing. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online, November 2020. Association for Computational Linguistics.

[178] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1823–1832, New York, NY, USA, 2019. Association for Computing Machinery.

[179] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[180] Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: BERT for finnish. *CoRR*, abs/1912.07076, 2019.

[181] Thuy-Trang Vu, Shahram Khadivi, Dinh Phung, and Gholamreza Haffari. Domain generalisation of NMT: Fusing adapters with leave-one-domain-out training. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 582–588, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[182] Junjie Wang, Yicheng Chen, Wangshu Zhang, Sen Hu, Teng Xu, and Jing Zheng. AdapterDistillation: Non-destructive task composition with knowledge distillation. In Mingxuan Wang and Imed Zitouni, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 194–201, Singapore, December 2023. Association for Computational Linguistics.

[183] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(8):5362–5383, August 2024.

[184] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *CoRR*, abs/2409.12191, 2024.

[185] Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. On negative interference in multilingual models: Findings and a meta-learning treatment. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online, November 2020. Association for Computational Linguistics.

[186] Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China, November 2019. Association for Computational Linguistics.

[187] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[188] Alina Wróblewska. Extended and enhanced Polish dependency bank in Universal Dependencies format. In Marie-Catherine de Marneffe, Teresa Lynn, and Sebastian Schuster, editors, *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 173–182, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[189] Alina Wróblewska and Katarzyna Krasnowska-Kieraś. Polish evaluation dataset for compositional distributional semantics models. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 784–792, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[190] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[191] Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *CoRR*, abs/2408.07666, 2024.

[192] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *CoRR*, abs/2306.06031, 2023.

[193] Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China, November 2019. Association for Computational Linguistics.

[194] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. *CoRR*, abs/1511.07122, 2016.

[195] Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. CoNLL 2018 shared task: Multilingual

parsing from raw text to Universal Dependencies. In Daniel Zeman and Jan Hajič, editors, *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[196] Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika

Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, Kyung-Tae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu,

Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Manying Zhang, and Hanzhi Zhu. Universal dependencies 2.5, 2019. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[197] Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[198] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2023.

[199] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[200] Hao Zheng and Mirella Lapata. Sentence centrality revisited for unsupervised summarization. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy, July 2019. Association for Computational Linguistics.

[201] Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. Counter-interference adapter for multilingual machine translation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2812–2823, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.