# Role of Language Relatedness in Multilingual Fine-tuning of Language Models: A Case Study in Indo-Aryan Languages

**Tejas Indulal Dhamecha**[*] **Rudra Murthy V**[*]**, Samarth Bharadwaj,**
**Karthik Sankaranarayanan**
IBM Research, India
{tidhamecha,rmurthyv,samarth.b,karsank}@in.ibm.com

**Pushpak Bhattacharyya**
Department of Computer Science and Engineering, IIT Bombay, India
pb@cse.iitb.ac.in

## Abstract

We explore the impact of leveraging the relatedness of languages that belong to the same family in NLP models using multilingual fine-tuning. We hypothesize and validate that multilingual fine-tuning of pre-trained language models can yield better performance on downstream NLP applications, compared to models fine-tuned on individual languages. A first of its kind detailed study is presented to track performance change as languages are added to a base language in a graded and greedy (in the sense of best boost of performance) manner; which reveals that careful selection of subset of related languages can significantly improve performance than utilizing all related languages. The Indo-Aryan (IA) language family is chosen for the study, the exact languages being Bengali, Gujarati, Hindi, Marathi, Oriya, Punjabi and Urdu. The script barrier is crossed by simple rule-based transliteration of the text of all languages to Devanagari. Experiments are performed on mBERT, IndicBERT, MuRIL and two RoBERTa-based LMs, the last two being pre-trained by us. Low resource languages, such as Oriya and Punjabi, are found to be the largest beneficiaries of multilingual fine-tuning. Textual Entailment, Entity Classification, Section Title Prediction, tasks of IndicGLUE and POS tagging form our test bed. Compared to monolingual fine tuning we get relative performance improvement of up to 150% in the downstream tasks. The surprise take-away is that for any language there is a particular combination of other languages which yields the best performance, and any additional language is in fact detrimental.

## 1 Introduction[1]

Transformer-based ([Vaswani et al., 2017](#)) language models (LM) have been proven to be extremely useful in variety of natural language processing (NLP) tasks. Some of the most notable models are GPT ([Radford et al., 2018](#)), GPT-2 ([Radford et al., 2019](#)), GPT-3 ([Brown et al., 2020](#)), BERT ([Devlin et al., 2019](#)), RoBERTa ([Liu et al., 2019](#)), XLNet ([Yang et al., 2019](#)), and BART ([Lewis et al., 2020](#)). To fine-tune a pre-trained language model for downstream tasks has become a *de facto* approach in recent literature.

We empirically study whether (and to what extent) do the related languages accentuate the performance of models in downstream tasks with multilingual fine-tuning[2] in comparison to monolingual fine-tuning. To understand the quantitative advantage of including languages gradually, we explore the gradation of multilinguality by incrementally adding in new languages added one-by-one building up to an all-language multilingual fine-tuning.

A good approximation for *language relatedness* is their membership to the same *language family* as languages of a family often share properties such as grammar, vocabulary, etymology, and writing systems. We choose the Indo-Aryan (IA) family for the study, since constituent languages 1) include low-resource languages, 2) have similar Abugida writing system, 3) are relatively understudied, and 4) are covered in a well-defined NLP benchmark IndicGLUE ([Kakwani et al., 2020b](#)). Further, the fact that all constituent languages except one use similar Abugida writing systems (rooted in the ancient Brahmi Script[3]) presents an opportunity for script normalization via transliteration.

Overall, although the general notion of language relatedness is explored, and the multilingual fine-tuning is explored in literature, the detailed

---

[*]Equal contribution
[1]https://github.com/IBM/indo-aryan-language-family-model

[2]Fine-tuning a pre-trained model with a downstream task's training data for multiple languages.
[3]https://en.wikipedia.org/wiki/Brahmic_scripts

linguistic understanding of role of language relatedness in multilingual fine-tuning remains understudied; even more so for IA family. Further, in this context the script-conversion aspect is not explored for multilingual fine-tuning.

To summarize, in this paper we seek to answer the following research questions (employing the Indo-Aryan language family as the experimental test-bed).

- **RQ1:** Does multilingual fine-tuning with a set of related languages yield improvements over monolingual fine-tuning (FT) on downstream tasks?

- **RQ2:** Starting from monolingual FT, as each related language is gradually added for multilingual FT, to ultimately a multilingual FT with all related languages, how does the performance vary? In other words, should one use all related languages' data or only a sub-set of languages' data?

These inquiries are critical to understanding the right balance between per-language fine-tuning and massively multilingual fine-tuning, as the viable way forward. Additionally, we also explore the role of common script representation in multilingual FT of related languages.

To facilitate these inquiries, we utilize existing pre-trained models, namely IndicBERT, mBERT, and MuRIL, and also pre-train two language models for Indo-Aryan language family from scratch. We utilize various tasks of IndicGLUE (Kakwani et al., 2020b) as our test-beds.

## 2 Related Work

Multilinguality aspect has been explored in context of pre-training language models, for effective transfer from one language to other, and in multilingual fine-tuning, to an extent.

### 2.1 Multilingual Pre-training

Multilingual LMs have enabled effective task fine-tuning across various languages. Notable examples include, multilingual BERT (mBERT)[4] model trained with 104 languages, and XLM (Lample and Conneau, 2019) and XLM-RoBERTa (Conneau et al., 2020a) trained with 100 languages.

In the context of Indic languages, three recent works on multilingual LMs have been IndicBERT,

MuRIL, and Indic-Transformers language models. IndicBERT (Kakwani et al., 2020b) focuses on languages belonging to the Indo-Aryan and Dravidian language families along with English. Along with 17 Indic languages and English, Multilingual Representations for Indian Languages (MuRIL)[5](Khanuja et al., 2021), utilizes English-Indic languages parallel corpus and the Roman transliterated counterparts to train an mBERT model. Similarly, Indic-Transformers (Jain et al., 2020) presents monolingual LMs for Hindi, Bengali, and Telugu. Recently, various types of word-embeddings are also trained for each language (Kumar et al., 2020; Kakwani et al., 2020b).

These approaches focus on multilingual *pre-training* of models. This means that once a multilingual LM is pre-trained, it is fine-tuned per task separately for each language.

### 2.2 Language Transfer

It is understood that a multilingual model gains cross-lingual understanding from sharing of layers that allows the alignment of representations among languages; to the extent that large overlap of the vocabulary between the languages is not required to bridge the alignment (Conneau et al., 2020b; Wang et al., 2019). This property facilitates, zero-shot transfer between two related languages (e.g. Hindi and Urdu) reasonably well (Pires et al., 2019). Performance for zero-shot transfer further improves when multilingual model is further aligned by utilizing parallel word or sentence resources (Kulshreshtha et al., 2020). Usually, the low-resource language members in a multilingual LM benefit by presence of related languages (Liu et al., 2020). Further, it is likely that presence of unrelated languages do not aid the multilingual training, but rather may lead to negative interference rooted in conflicting gradients (Wang et al., 2020b) or yield substantially poor transfer between unrelated languages (e.g. English and Japanese) (Pires et al., 2019). A recent work by Dolicki and Spanakis (2021) focuses on establishing the connection between the effectiveness of zero-shot transfer and the linguistic feature of source and target languages; interestingly, they observe that the effectiveness of zero-shot transfer is a function of downstream task, in addition to the languages themselves.

The general understanding has been that language-specific FT serves as skyline, and, in

these set of works, pursuit has been to get zero-shot transfer from related languages(s) closer to the skyline (Wu and Dredze, 2019).

## 2.3 Multilingual Fine-tuning

Tsai et al. (2019) perform multilingual fine-tuning of 48 languages for downstream tasks of POS tagging and morphological tagging, and find these multilingual models to be slightly poorer compared to monolingual models. For morphological tagging and lemmatization tasks, Kondratyuk (2019) makes similar observation regarding poor performance for the model fine-tuned with 66 languages in multilingual setting compared to monolingual fine-tuning (although, a second stage of per-language fine-tuning yields superior performance). *These findings indicate that arbitrary collection of languages may not be suitable for improving downstream task performance; and that, a principled approach for selecting a set of languages may be preferable for multilingual fine-tuning.* To this end, we hypothesize that language relatedness should be an important aspect to consider while selecting a language set for multilingual fine-tuning.

Pires et al. (2019) briefly explore language set selection based on topological features (syntactic word order). Wang et al. (2020b) explores multilingual fine-tuning in strictly bilingual settings. Taking the language relatedness in considerations, Tran and Bisazza (2019) show that joint fine-tuning with four European language is better than fine-tuning with only English in the specific task of universal dependency parsing. Unfortunately, it doesn't provide comparison with monolingual fine-tuning of all constituent languages.

We observe that there is a void regarding the systematic analysis to understand how a presence of related languages in the multilingual fine-tuning affects the performance on the target language.

## 3 Methodology

Traditionally, a pre-trained LM (such as mBERT) is used as base model, which is fine-tuned for a downstream task for a specific language (monolingual). In this work, we aim to evaluate the role of script and language relatedness in multilingual fine-tuning by employing the Indo-Aryan language family. Therefore, we include the following components to the approach: (1) multilingual fine-tuning, (2) transliteration, and (3) language models. Next, we discuss these in detail.

## 3.1 Multilingual Fine-Tuning

As opposed to traditional monolingual fine-tuning for a downstream task, in multilingual fine-tuning the LM is fine-tuned once per task with the aggregate labelled corpus across languages. Intuitively, related languages should assist each other for a downstream task. To draw a parallel, a polyglot person (akin to a multilingual LM) good at guessing titles of passages written in one language can easily adapt this skill on another, albeit related, language with few exemplars. Arguably, when put together, a greater understanding of the downstream task arises compared to what each language would yield individually, and the relatedness of associated languages would play a key role in deciding the benefits of this approach.

Therefore, to study this systematically, for a downstream task that is relevant for a variety of languages (e.g. part-of-speech tagging or named entity recognition), first the training sets of all languages are combined to create the multilingual task training set. Then, the base LM is fine-tuned on the multilingual task training corpus. This multilingual fine-tuning now yields a model per task, and not per task-language pair.

## 3.2 Script Similarity and Transliteration

Languages of a language family often use similar writing systems. For example, in the IA family, on one hand, Hindi, Bhojpuri, Magahi, Marathi, Sanskrit, and Nepali are written in Devanagari script. On the other hand, Bengali, Gujarati, Punjabi, and Oriya are written in their respective scripts. As Indic languages have high lexical similarity (Bhattacharyya et al., 2016), having a universal script for all these languages allows for model to exploit cross-lingual similarities. For example, the verb term for "to go" is similar in Hindi (जाना jaanaa), Urdu (جانا jaanaa), Gujarati (જવું javum), Punjabi (ਜਾਣਾ jaanaa), Marathi (जाने jaane), and Oriya (ଯିବାକୁ jibaku), and Bengali (যাওয়া jao) with each language morphing it in different manners. We use indic-nlp-library (Kunchukuttan et al., 2015) (for all but Urdu) and indic-trans tools (Bhat et al., 2015) (for Urdu) for transliteration to Devanagari.

## 3.3 Language Models

For this study, we use the mBERT, IndicBERT, and MuRIL as existing pre-trained language models. Additionally, we pre-train language models

(from scratch) specifically for only the Indo-Aryan languages, as other LMs contains languages of other families too.

**Pre-training Language Model From Scratch:** We choose to pre-train RoBERTa (Liu et al., 2019) transformer-based model as it has been shown to improve over BERT (Devlin et al., 2019) recently. Existing pre-trained language models are trained on original script data. For fair study of effectiveness of transliteration, we wish to pre-train separate language models on original and transliterated corpus from scratch. Our experimentation around transliteration makes existing pre-trained models (mBERT, IndicBERT, and MuRIL) somewhat incompatible. In other words, it would be akin to fine-tuning for an unseen language, albeit in a previously seen script. Thus, we settle upon pre-training contextual LM from scratch for this purpose. Specifically, we train two LMs from scratch, one preserving the original scripts of corpora (IndoAryan-Original) and other after transliterating all corpora to Devanagari script (IndoAryan-Transliterated).

## 4 Experimental Setup

In this section, we describe the datasets used in our experiments, their pre-processing, and implementation details.

### 4.1 Data

To train the language models, we obtained text data from various sources including: Wikipedia Dump[6], WMT Common Crawl [7], WMT News CommonCrawl[8], Urdu Charles University Corpus (Bojar et al., 2014; Jawaid et al., 2014), IIT Bombay Hindi Monolingual Corpus (Kunchukuttan et al., 2018), Bhojpuri Monolingual Corpus (Kumar et al., 2018), and Magahi Monolingual Corpus[9]. Various statistics of the collected corpus are reported in Table 1. Note the major imbalance in the data with Hindi being undoubtedly a high-resource language and likes of Magahi, Punjabi, and Oriya being low-resource languages. The challenges of the data imbalance and insufficiency of data to train monolingual models for many of these languages is apparent from the statistics.

---

| Language | # Sentences | # Tokens | |
| --- | --- | --- | --- |
| | | # Total | # Unique |
| Hindi (hi) | 1552.89 | 20,098.73 | 25.01 |
| Bengali (bn) | 353.44 | 4,021.30 | 6.5 |
| Sanskrit (sa) | 165.35 | 1,381.04 | 11.13 |
| Urdu (ur) | 153.27 | 2,465.48 | 4.61 |
| Marathi (mr) | 132.93 | 1,752.43 | 4.92 |
| Gujarati (gu) | 131.22 | 1,565.08 | 4.73 |
| Nepali (ne) | 84.21 | 1,139.54 | 3.43 |
| Punjabi (pa) | 68.02 | 945.68 | 2.00 |
| Oriya (or) | 17.88 | 274.99 | 1.10 |
| Bhojpuri (bh) | 10.25 | 134.37 | 1.13 |
| Magahi (mag) | 0.36 | 3.47 | 0.15 |

Table 1: Statistics (in Millions) of monolingual corpora used in pre-training IndoAryan LMs from scratch.

### 4.2 Data Preparation and Implementation Details

As the first step, sentences are segmented from the text corpora. Then script converted version of the datasets is obtained by transliterating Bengali, Gujarati, Punjabi, Oriya, and, Urdu into Devanagari script. We additionally perform de-duplication to remove repeated sentences. The statistics of the resulting set are reported in Table 1. We identify following two challenges that can affect pre-training negatively: 1) data imbalance and 2) compute requirements.

1. **Data imbalance:** As reported in Table 1 the size of each language corpora differs up to four orders of magnitude, e.g. Hindi has 1552M sentences vs 0.36M in Magahi and 17.88M in Oriya. The language bias can creep into the tokenizer and the language model pre-training, if left unattended. To mitigate bias in tokenizer training, we utilize a re-sampling strategy to reduce the data skew (Lample and Conneau, 2019). Specifically, samples are drawn following multinomial distribution with *adjusted* probabilities. Adjusted probabilities are computed as $q_i = p_i^\alpha / \sum_j p_j^\alpha$ where $p_i = n_i / \sum_i n_i$, and $n_i$ is the number of samples in $i^{th}$ language. Before rescaling, the language distribution is in the range of 0.01%-58%, which changes to 5-12% afterwards.

2. **Compute requirements:** Depending on the computing infrastructure, running one training epoch can typically take few hundreds to (single digit) few thousands of GPU hours. To mitigate this, we utilize a variant of sharding technique outlined in Algorithm 1 to pre-train the model in infrastructure with limited memory (<50GB)

**Algorithm 1:** Economical LM training for language family

> **Result:** Trained LM checkpoint
> randomly select working language set;
> **for** *each language in working set* **do**
> > randomly select
> > least number of blocks containing
> > x% sentence of the language;
> > **for** *each block* **do**
> > > **if** *block is not cached* **then**
> > > > tokenize the block;
> > > > persist on to storage;
> > >
> > > **end**
> > > read tokenized block into dataset;
> >
> > **end**
>
> **end**
> **if** *checkpoint found* **then**
> > initialize model with checkpoint;
>
> **else**
> > randomly initialize model;
>
> **end**
> train LM to minimize
> MLM loss on balanced mini-batches;

and compute (one v100 GPU). It depends on dividing each language corpus into manageable (into memory) chunks, termed as *blocks*. Each LM is trained over ∼50 sequential executions of Algorithm 1 on a single v100 GPU machine and each execution running for a day, consuming about 1200 hours overall for pre-training.

In the re-sampling step, exponent $s = 0.1$ and scaling parameter of $\gamma = 100$ are used. Byte-level BPE tokenizer (Radford et al., 2019; Wang et al., 2020a) is used with vocabulary size of 110K. Trained LMs use 12 layers, 12 attention heads, hidden size of 768, and dropout ratio for the attention probabilities of 0.1. Our implementation uses Huggingface (Wolf et al., 2020) library. We use linear schedule for learning rate decay. Maximum sequence length is set as 128 across tokenization, training, and fine-tuning. Due to compute limitations, having higher maximum sequence length lead to out-of-memory errors. Mini-batches are created by weighted sampling based on language priors with exponent $s = 0.7$. In LM pre-training, mini-batch of 48 samples and gradient accumulation of 53 is used making the effective batch size as 2,544[10].

---
[10]Loss curves of LM pre-training in supplementary material.

Apex[11] library is used with `O1` optimization level to allow mixed precision training. In all our experiments on fine-tuning, we perform a grid-search with respect to *learning-rate* and *batch size* values of $\{1,3,5\} \times 10^{-5}$ and $\{16,32,64\}$ respectively.

## 5 Experiments

To answer the research questions, we experiment on a variety of tasks suitable for multilingual fine-tuning and analyse the results. To investigate RQ1, in §5.1, the first set of experiments are aimed to understand the utility of multilingual FT with related languages. To investigate RQ2, in §5.2, the second set of experiments are designed to track gradual performance variation with addition of assisting languages. With last set of analysis, in §5.3, we investigate the role of transliteration.

### 5.1 Effectiveness on Multilingual Tasks

We experiment on four tasks suitable for multilingual fine-tuning protocol, including three from IndicGLUE (Kakwani et al., 2020b) and POS tagging (Zeman et al., 2020).

1. **Textual Entailment** task on copa-translated and wnli-translated dataset (Hi, Gu, Mr)

2. **Title Prediction** task on wiki-section-title dataset (Hi, Bn, Gu, Mr, Or, Pa)

3. **Named Entity Recognition** task on wikiann-ner dataset (Hi, Bn, Gu, Mr, Or, Pa)

4. **Part-of-Speech Tagging** task on Universal Dependency datasets (Hi, Mr, Ur)

We do not show results on Cloze-style Question Answering task of IndicGLUE as it is meant to evaluate masked-token prediction of an LM, and does not involve downstream task training.

We utilize mBERT, IndicBERT, MuRIL, IndoAryan-Original (IA-O) and IndoAryan-Transliterated (IA-TR) – last two being pre-trained by us as detailed in §3.3. All the five LMs are fine-tuned in monolingual and multilingual modes, to pursue investigation for RQ 1. Only the IA-TR model is fine-tuned with transliterated versions of the downstream task data; remaining four models are fine-tuned with original script downstream task data. The results of this set of experiments are reported in Table 2. Along with absolute metrics, the relative difference between mono- and multi-

---
[11]https://github.com/nvidia/apex

| LG (FT →) | mBERT | | | IndicBERT | | | MuRIL | | | IA-Transliterated | | | IA-Original | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mono | Multi | $\delta_{MB}$ | Mono | Multi | $\delta_{IB}$ | Mono | Multi | $\delta_{MIR}$ | Mono | Multi | $\delta_{TR}$ | Mono | Multi | $\delta_O$ |
| **Textual Entailment / copa-translated (Accuracy)** | | | | | | | | | | | | | | | |
| hi | 0.6590 | 0.5909 | (-10.33) | 0.6250 | **0.6705** | (+7.28) | 0.5568 | 0.6477 | (+16.33) | 0.6591 | 0.5455 | (-17.24) | 0.5796 | 0.5796 | (0.00) |
| gu | 0.4318 | 0.5795 | (+34.21) | 0.5341 | 0.6591 | (+23.40) | 0.5909 | 0.5795 | (-1.93) | 0.5909 | 0.6250 | (+5.77) | 0.5113 | 0.5227 | (+2.23) |
| mr | 0.5568 | 0.5909 | (+6.12) | 0.5909 | 0.6591 | (+11.54) | 0.5682 | 0.6363 | (+11.99) | 0.5909 | 0.6023 | (+1.93) | 0.6590 | 0.5454 | (-17.24) |
| **Textual Entailment / wnli-translated (F-Score)** | | | | | | | | | | | | | | | |
| hi | 0.3604 | 0.4020 | (+11.54) | 0.3604 | **0.4855** | (+34.71) | 0.3604 | 0.3604 | (0.00) | 0.3604 | 0.3604 | (0.00) | 0.3616 | 0.3604 | (-0.33) |
| gu | 0.3604 | 0.4123 | (+14.40) | 0.3604 | **0.5071** | (+40.70) | 0.3604 | 0.3604 | (0.00) | 0.3107 | 0.3604 | (+15.99) | 0.3604 | 0.3604 | (0.00) |
| mr | 0.4167 | 0.4920 | (+18.07) | 0.3604 | **0.4685** | (+29.99) | 0.3914 | 0.3604 | (-7.92) | 0.4228 | 0.3576 | (-15.42) | 0.3604 | 0.3604 | (0.00) |
| **Entity Classification / wikiann-ner (F-Score)** | | | | | | | | | | | | | | | |
| hi | 0.8656 | 0.9197 | (+6.25) | 0.9031 | 0.9229 | (+2.19) | 0.9237 | **0.9446** | (+2.26) | 0.8720 | 0.8897 | (+2.03) | 0.8476 | 0.8776 | (+3.54) |
| bn | 0.9181 | 0.9671 | (+5.34) | 0.9339 | **0.9654** | (+3.37) | 0.9503 | 0.9645 | (+1.49) | 0.9211 | 0.9397 | (+2.02) | 0.9486 | 0.9557 | (+0.75) |
| gu | 0.6804 | 0.8893 | (+30.70) | 0.7021 | 0.9027 | (+28.57) | 0.8016 | **0.9058** | (+12.99) | 0.7306 | 0.7796 | (+6.71) | 0.8318 | 0.8650 | (+3.99) |
| mr | 0.9127 | 0.8882 | (-2.68) | 0.8871 | 0.8965 | (+1.06) | 0.9199 | **0.9274** | (+0.82) | 0.8675 | 0.8675 | (0.00) | 0.8482 | 0.8605 | (+1.45) |
| or | 0.1905 | 0.3457 | (+81.47) | 0.3509 | **0.8896** | (+153.52) | 0.3882 | 0.8848 | (+127.92) | 0.3460 | 0.6436 | (+86.01) | 0.5737 | 0.7038 | (+22.68) |
| pa | 0.5000 | 0.8667 | (+73.34) | 0.4444 | 0.8593 | (+93.36) | 0.8535 | **0.9086** | (+6.45) | 0.3491 | 0.5789 | (+65.83) | 0.6313 | 0.7456 | (+18.10) |
| **Title Prediction / wiki-section-title (Accuracy)** | | | | | | | | | | | | | | | |
| hi | 0.8012 | 0.8081 | (+0.86) | 0.7780 | 0.7976 | (+2.52) | 0.8528 | **0.8622** | (+1.10) | 0.6779 | 0.6994 | (+3.17) | 0.6761 | 0.6807 | (+0.68) |
| bn | 0.8253 | 0.8285 | (+0.39) | 0.8266 | 0.8157 | (-1.32) | 0.8781 | **0.8874** | (+1.06) | 0.6135 | 0.7111 | (+15.91) | 0.7062 | 0.7097 | (+0.49) |
| gu | 0.7452 | 0.7858 | (+5.45) | 0.6879 | 0.8074 | (+17.37) | 0.8465 | **0.8745** | (+3.31) | 0.2614 | 0.6970 | (+166.64) | 0.4044 | 0.6890 | (+70.38) |
| mr | 0.8049 | 0.8438 | (+4.83) | 0.7744 | 0.8247 | (+6.49) | 0.8529 | **0.8913** | (+4.50) | 0.5299 | 0.7167 | (+35.25) | 0.5031 | 0.7083 | (+40.78) |
| or | 0.2222 | 0.2829 | (+27.31) | 0.6825 | 0.8108 | (+18.79) | 0.8167 | **0.8845** | (+8.30) | 0.2928 | 0.7231 | (+146.96) | 0.3147 | 0.6852 | (+117.73) |
| pa | 0.7247 | 0.7648 | (+5.53) | 0.7754 | 0.7903 | (+1.92) | 0.8240 | **0.8487** | (+2.99) | 0.2817 | 0.7064 | (+150.76) | 0.6235 | 0.6800 | (+9.06) |
| **Part-of-Speech Tagging / ud-pos (F-Score)** | | | | | | | | | | | | | | | |
| hi | 0.9693 | 0.9698 | (+0.05) | 0.9755 | 0.9748 | (-0.07) | **0.9779** | 0.9778 | (-0.01) | 0.9618 | 0.9636 | (+0.19) | 0.9562 | 0.9528 | (-0.36) |
| mr | **0.9024** | 0.8932 | (-1.02) | 0.8024 | 0.8886 | (+10.74) | 0.5388 | 0.7478 | (+38.79) | 0.8114 | 0.8290 | (+2.17) | 0.7906 | 0.7926 | (+0.25) |
| ur | 0.9102 | 0.8149 | (-10.47) | 0.9047 | 0.8249 | (-8.82) | **0.9168** | 0.8497 | (-7.32) | 0.9026 | 0.9080 | (+0.60) | 0.8915 | 0.7858 | (-11.85) |

Table 2: Comparison of multilingual fine-tuning vs monolingual fine-tuning of various Indo-Aryan LMs. The figure in parenthesis is relative difference $\delta$ (Eq. 1). The monolingual results for mBERT and IndicBERT models are as reported by Kakwani et al. (2020a) except for wnli-translated and POS-Tagging tasks. LG and FT stand for language and fine-tuning, respectively.
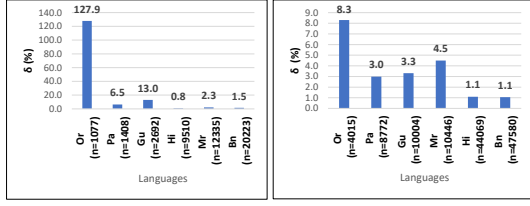
Figure 1: Relative performance difference $\delta$ (Eq. 1) for MuRIL model on (left) NER and (right) title prediction tasks. $n$ is number of training samples.

| Text | ਪੰਜਾਬ ਦੇ ਮੁੱਖ ਮੰਤਰੀਆਂ ਦੀ ਸੂਚੀ | | | | | |
|---|---|---|---|---|---|---|
| Pronounciation | pajAba | de | mukha | matarIAM | dI | sUchI |
| Word Translate | Punjab | of | Chief | Minister | of | List |
| GT | Organization | | | | | |
| Mono | Location | | | Person | | |
| Multi | Organization | | | | | |

(a) Entity tag change

| Text | ਫ਼ਰਾਂਸੀਸੀ | ਗੁਇਆਨਾ | ਫ਼ਰਾਂਸ | ਦਾ ਇੱਕ | ਵਿਦੇਸ਼ੀ | ਵਿਭਾਗ |
|---|---|---|---|---|---|---|
| Pronounciation | pharAMsIsI | guIAnA | pharAMsa | dA ika | videsal | vibhAga |
| Word Translate | French | Guyana | France | of a | Foreign | Department |
| GT | Location | | | O | O | |
| Mono | Location | | | O | Location | |
| Multi | Location | | | O | O | |

(b) Spurious entity extraction rectified

Figure 2: Examples of prediction improvement by multilingual FT compared to monolingual FT, for NER task for two examples of Punjabi written in Gurmukhi script.

lingual fine-tuning (FT) is also reported. The relative difference is calculated as

$$\delta = 100 \times \frac{M_{multi} - M_{mono}}{M_{mono}} \qquad (1)$$

where $M_{mono}$ and $M_{multi}$ are performance measures of monolingual and multilingual fine-tuning respectively. Key observations are as following:

**Monolingual vs Multilingual Fine-Tuning:** In this analysis, higher the $\delta$, the stronger is the answer to RQ1 in affirmation. In the Table 2 the positive $\delta$, shown in blue color, indicates the cases where multilingual FT improves over monolingual FT. It can be observed that for languages with limited labelled data for the downstream task, the multilingual fine-tuning is resulting in enormous improvements. Across all the five LMs, the trend is consistent. For example, on wikiann-ner task, F-Score on Oriya improved from 0.3882 and 0.3460 to 0.8848 and 0.6436, respectively for MuRIL and IA-TR models; while significant improvements are seen in other languages too. Similar trend is visible in wiki-section-title prediction task too, where improvements are seen for all the languages. Broadly, across LMs, tasks, and languages, the multilingual FT shows improvement over monolingual FT. This helps formulate the answer to RQ1 as *the multilingual fine-tuning with related languages can yield huge (up to 40% on absolute scale) improvement for low-resource languages (such as Oriya and Punjabi), and statistically significant (up to 10%) improvements on high resource languages (such as Hindi and Bengali), depending on the task.*

Note, that this is in contrast to the observations of Tsai et al. (2019) and Kondratyuk (2019) that indicate slightly poor performance with multilingual fine-tuning. They fine-tune with more than forty languages together, without considering language relatedness. We observe large improvements by selecting only the languages of the family for multilingual fine-tuning.

**Trade-off or Win-Win?:** Figure 1 visualizes the the improvements by multilingual fine-tuning relative to the monolingual one along with the task training set size. It is clearly evident, the smaller the task training data set, the higher is the relative improvement. Arguably, the data limitation of a low resource language is abridged by the related high resource languages. These improvements are not at the cost of trading-off on high resource languages; it is win-win for all languages. In fact, a decrease in performance ($\delta < 0$), indicated with red color, is observed in only 16 out of the total 105 (21 task-language pairs $\times$ 5 LMs) comparisons. Interestingly, there is no task-language pair in which $\delta$ values corresponding to all the five LMs are negative, i.e. for all task-language pairs at least one LM always showed improvement using multilingual fine-tuning. Figure 2 illustrates the types of improvements in predicting entity tags.

**Best LM Across Tasks?:** Arguably, it is unfair to compare the pre-trained LMs due to vast difference in the number of languages they are pre-trained for (in range from 11 to 104), the size of the corpora, nature of the corpora (only monolingual vs parallel corpora), model types (RoBERTA, AlBERT, BERT), number of layers (8 or 12), tokenization, pre-training objectives, and compute consumed in training. Further, mBERT model is not pre-trained with Oriya. However, it is natural to inquire if there is a clear winner LM in the experimentation. The boldface figures in Table 2 shows the best results per task per language. Most of the best metrics fall under either IndicBERT (for Textual Entailment tasks) or MuRIL models (for Title Prediction and

mostly for Entity Classification tasks). The lowest performance obtained is with IA-O and IA-TR.

## 5.2 Gradation of Multilinguality

We further dwell into understanding whether the degree of improvement varies by the language closeness within language family. Specifically, we start with the monolingual training, i.e. training set contains only the target language. Then, we experiment by adding each related language to the training set separately. The language that yields highest performance boost is selected for adding to the training set. Thus, a new training set consisting of two languages is obtained. This is repeated until all the related languages are added to the training set, resulting in the all-language multilingual FT. This approach is similar to *Sequential Forward Selection of Features* in machine learning. Further, we relate the subfamily categorization of IA family for this analysis.

Experiments are performed for NER task on MuRIL model, with Oriya and Punjabi as the evaluation language. The results are reported in Table 3. For detailed discussion, consider the case of Oriya, on one end of the spectrum, in the first row, we have a monolingual fine-tuned model with only Oriya, whereas on the other end of the spectrum, in the last row, we have a multilingually fine-tuned model with Oriya, Bengali, Hindi, Gujarati, Marathi, and Punjabi. In the middle span, we have Oriya aided by each – Bengali, Punjabi, Marathi, Gujarati, and Hindi, separately; and their combinations. Since adding Gujarati to Oriya yields best result compared to adding any other language, *or+gu* is taken as the base training set for next iteration. In the next iteration, adding Bengali to *or+gu* provides highest boost, thus *or+gu+bn* forms the base iteration for next iteration. A similar exercise is performed with Punjabi as the base language too.

In case of Oriya, adding the Gujarati data resulted in about 54 percentage points improvement (38.8% to 92.4%), which is further improved by about 0.5% with addition of Bengali (93.0%). It appears that Hindi, Punjabi, and Marathi each negatively interferes with *or+gu+bn* set, resulting in at least 1.5 percentage points performance drop. The best performance of 94.19% is obtained with the *or+gu+bn+mr+hi* set (i.e. all but Punjabi), which is 5.7 percentage points higher than considering all the languages together.

It is natural to ask if Punjabi has negative interference with Oriya for the task. *or+pa* yields 47.7 percentage points improvement over only Oriya, which indicates positive interference between them. Further, *or+gu* and *or+gu+pa* are almost similar (0.9245 and 0.9231) indicates that perhaps, *pa* is redundant to *gu* for assisting *or* on the task. However, adding *pa* to *or+gu+bn*, *or+gu+bn+mr*, and *or+gu+bn+mr+hi* results in 2-5% drop; the common denominator being Bengali, it seems that Punjabi harms the most when the base set contains Bengali. Arguably, it indicates negative interference between Bengali and Punjabi for the task.

Also, note the improvements are not correlated with increase in the training set size; instead the smaller sets (e.g. *or+gu* with 3425 samples) yields better results than larger sets (e.g. *or+bn* of 21379). Therefore, gradual deviations should be credited to the addition language rather than the training set inflation. overall, the answer to RQ 2 emerges *within the set of related languages, likely, there exists a subset of languages that yields the best performance.*

## 5.3 Transliteration

Next, we present a set of observations pertaining to the utility of transliteration to leverage the script similarity between the Indo-Aryan languages. For a fair comparison, the IA-Original and IA-Transliterated models are considered as both of them are pre-trained by us on the original script and transliterated script versions of the same corpora. Thus, in this part of analysis, higher the $\delta_{TR} - \delta_O$ in Table 2, the stronger is the role of explicit script normalization.

**Transliteration with Multilingual FT:** Comparing the relative difference ($\delta$) for transliterated and original script models, it is observed that in 16 out of 21 task-language pairs $\delta_{TR} > \delta_O$; noteworthy is $\delta_{TR} = 146.96\%$, and $\delta_O = 117.73\%$ for Oriya language on wiki-section-title prediction task. It suggests that multilingual FT is even more suitable with transliteration. Based on these results, the role of common script representation emerges as *effectiveness of multilingual fine-tuning is significantly enhanced when coupled with common-script representation via transliteration.*

**Transliteration for LM:** Comparing performance of monolingual fine-tuning of the original script LM (IA-O) and the transliterated script LM (IA-TR) reveals that the later is better in only few (8 out of 21)

| Train Set | Size | F-Score | Train Set | Size | F-Score |
|---|---|---|---|---|---|
| or | 1078 | 0.3882 | pa | 1409 | 0.8535 |
| Base set: or | | | Base set: pa | | |
| **+ gu** | 3425 | 0.9245 | **+ bn** | 21579 | 0.9156 |
| + bn | 21379 | 0.8836 | + mr | 13795 | 0.8883 |
| + hi | 10590 | 0.8795 | + hi | 10970 | 0.8759 |
| + pa | 2487 | 0.8657 | + gu | 3805 | 0.8673 |
| + mr | 13415 | 0.8649 | + or | 2487 | 0.8426 |
| Base set: or+gu | | | Base set: pa+bn | | |
| **+ bn** | 23725 | 0.9301 | **+ hi** | 31270 | 0.9286 |
| + pa | 4884 | 0.9231 | + mr | 34095 | 0.9160 |
| + hi | 12936 | 0.8836 | + or | 22838 | 0.9137 |
| + mr | 15761 | 0.8855 | + gu | 24105 | 0.9105 |
| Base set: or+gu+bn | | | Base set: pa+bn+hi | | |
| **+ mr** | 36061 | 0.9151 | **+ mr** | 43606 | 0.9211 |
| + pa | 25184 | 0.9036 | + or | 32349 | 0.9156 |
| + hi | 33236 | 0.8916 | + gu | 33616 | 0.9132 |
| Base Set: or+gu+bn+mr | | | Base set: pa+bn+hi+mr | | |
| **+ hi** | 45572 | **0.9419** | **+ gu** | 45952 | **0.9567** |
| + pa | 37520 | 0.8922 | + or | 44685 | 0.9231 |
| All | 46665 | 0.8848 | All | 46665 | 0.9086 |

Table 3: Study of graded addition of languages for NER task on low resource languages of (left) Oriya and (right) Punjabi using MuRIL. The scheme of adding the languages is similar to Greedy Forward Selection of features in Machine Learning.

experiments. This is somewhat counter intuitive, as the common script representation should have made the LM pre-train better due to the presence of cognates. We speculate following two rationales.

- Firstly, it indicates that, perhaps, even without explicit alignment of cognates (via transliteration) the model is able to align their embeddings implicitly, corroborating with (Conneau et al., 2020b; Pires et al., 2019).

- Secondly, the byte-Level BPE and the unicode block arrangements for Indo-Aryan languages may be at play underneath this phenomena also. For example, the consonant Pa in Hindi प (0xe0 0xa4 0xaa), Oriya ପ (0xe0 0xac 0xaa), and Punjabi ਪ (0xe0 0xa8 0xaa) are apart by their unicode block offset differences. Thus, potentially, a model knowing the byte level representations of the writing system could learn to map them, provided the loss function guides it.

However, we leave the further inquiry into the exact phenomena for the future work.

## 6 Conclusion

We show that multilingual fine-tuning efficiently leverages language relatedness leading to improvements over monolingual approach. We substantiate this claim on the Indo-Aryan language family with experiments on five language models. Multilingual fine-tuning is particularly effective for low-resource languages (e.g., Oriya and Punjabi show improvement up to 150% on relative scale). Also, we show that careful selection of subset of related languages, can further improve performance. Devising automatic approaches for finding optimal subset of related languages is a promising future direction. Additionally, in multilingual fine-tuning, we see some benefits of transliteration to common script.

## References

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shri-

vastava. 2015. IIIT-H System Submission for FIRE2014 Shared Task on Transliterated Search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Pushpak Bhattacharyya, Mitesh M. Khapra, and Anoop Kunchukuttan. 2016. Statistical machine translation between related languages. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, pages 17–20, San Diego, California. Association for Computational Linguistics.

Ondřej Bojar, Vojtěch Diatka, Pavel Straňák, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp 0.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam Mc-Candlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *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.

Błażej Dolicki and Gerasimos Spanakis. 2021. Analysing The Impact Of Linguistic Features On Cross-Lingual Transfer.

Kushal Jain, Adwait Deshpande, Kumar Shridhar, Felix Laumann, and Ayushman Dash. 2020. Indic-Transformers: An Analysis of Transformer Language Models for Indian Languages. *arXiv preprint arXiv:2011.02323*.

Bushra Jawaid, Amir Kamran, and Ondřej Bojar. 2014. Urdu monolingual corpus. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020a. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020b. IndicNLP-Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. MuRIL: Multilingual Representations for Indian Languages.

Dan Kondratyuk. 2019. Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–18, Florence, Italy. Association for Computational Linguistics.

Saurabh Kulshreshtha, Jose Luis Redondo Garcia, and Ching Yun Chang. 2020. Cross-lingual Alignment Methods for Multilingual BERT: A Comparative Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 933–942.

Ritesh Kumar, Bornini Lahiri, Deepak Alok, Atul Kr Ojha, Mayank Jain, Abdul Basit, and Yogesh Dawer. 2018. Automatic identification of closely-related Indian languages: Resources and experiments. *arXiv preprint arXiv:1803.09405*.

Saurav Kumar, Saunack Kumar, Diptesh Kanojia, and Pushpak Bhattacharyya. 2020. "a passage to India": Pre-trained word embeddings for Indian languages. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing*

*for Under-Resourced Languages (CCURL)*, pages 352–357, Marseille, France. European Language Resources association.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Anoop Kunchukuttan, Ratish Puduppully, and Pushpak Bhattacharyya. 2015. Brahmi-net: A transliteration and script conversion system for languages of the Indian subcontinent. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 81–85, Denver, Colorado. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training (2018). *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ke Tran and Arianna Bisazza. 2019. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 281–288, Hong Kong, China. Association for Computational Linguistics.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical BERT models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3632–3636, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020a. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9154–9160.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020b. On Negative Interference in Multilingual Models: Findings and A Meta-Learning Treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime G. Carbonell. 2019. Cross-lingual alignment vs joint training: A comparative study and A simple unified framework. *CoRR*, abs/1910.04708.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019.

Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, and Elia Ackermann *et al.* 2020. Universal dependencies 2.7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.