Technische Universität Berlin
Fakultät IV - Elektrotechnik und Informatik
Institut für Softwaretechnik und Theoretische Informatik

# Impact of Continual Multilingual Pre-training on Cross-Lingual Transferability for Source Languages

Master Thesis

for the degree of
Master of Science (M. Sc.)

submitted by: Julio Perez
Advisor: Fabio Barth
Examiners: Prof. Dr. Sebastian Möller
Prof. Dr. Georg Rehm

submitted on: .....

**Abstract.** Write your abstract here.

# Contents


**1 Introduction** **1**

**2 Theoretical Background** **1**
2.1 Machine Learning . . . . . 1
2.2 Deep Learning . . . . . 3
2.2.1 Neural Networks . . . . . 4
2.2.2 Training . . . . . 7
2.3 Natural Language Processing . . . . . 7
2.3.1 Embeddings . . . . . 7
2.4 Large Language Models . . . . . 8
2.5 Transformer Architecture . . . . . 8
2.6 Transfer Learning . . . . . 8
2.7 Continual Pre-training of Large Language Models . . . . . 8
2.8 Semantic Web . . . . . 9

**3 Research Background** **9**
3.1 Transfer Learning . . . . . 9
3.2 Cross-Lingual Transfer . . . . . 9
3.3 Text-to-SPARQL . . . . . 10

**4 Contribution** **10**

**5 Methodology** **10**
5.1 Data Collection . . . . . 10
5.2 Data Analysis . . . . . 11

**6 Experimental Setup** **11**
6.1 Data Processing . . . . . 11
6.2 Training . . . . . 11

**7 Results** **11**
7.1 Mistral . . . . . 11
7.2 Occiglot . . . . . 11

**8 Discussion** **11**
8.1 Comparison . . . . . 11
8.2 Interpretation . . . . . 11

**9 Conclusion** **11**

**Appendix** **iv**

# 1 Introduction

# 2 Theoretical Background

## 2.1 Machine Learning

[1]

Machine Learning is largely regarded as a subfield of AI. As its name suggests, the central concept is learning from data, through a process called training. Although the techniques are abundant, they all mostly draw their foundations from the fields of optimization and probabilistic inference.

When describing a machine learning system, there are some foundational concepts which they all share:

- **Problem Class:** Refers to the type of problem we are trying to solve. Some of the most common problem classes are: classification, prediction, clustering, dimensionality reduction, summarization, and translation. This will all be encompassed by the word "prediction" and the verb "predict" from this point onwards.
- **Assumptions:** It is assumed that previous data, or a subset of the data will help us to learn patterns from it that generalize to unseen data. More formally, we assume that the data we use for the learning algorithm (training data), and the rest of the data were sampled from the same probability distribution. Other assumptions that could be made depending on the problem are independence of features (characteristics of the data) and linear relationships.
- **Generalization:** As mentioned previously, the algorithm should perform well on non-training data. This is called generalization.
- **Learning Algorithm:** The learning algorithm is used to learn from data and make predictions, strongly linked to the problem class and the nature of the available training data like quantity.
- **Parameters:** Refers to the learnable part of the system. It is the piece that is used for the prediction after training, along with the architecture of the system. This vector is also called the weights of the model.
- **Hyperpameters:** These refer to adjustable values used in the learning function. Their tuning to perform better is highly coupled with the training data and the problem class. This process is usually called hyperparameter tuning.
- **Loss Function:** A mathematical function that quantifies the difference between model predictions and actual target values. It guides the training process by providing a measure to minimize.

[1] Note that this section is based on the lecture materials of *6.036 Introduction to Machine Learning* course by MIT [1], unless stated otherwise.

- **Model:** Refers to the mathematical model comprising the architecture, and the parameters.
- **Evaluation Function:** Similar to the loss function, it receives the output of the model for a subset of data points (non-training data) and the target values, outputting a number signifying the distance between them. It is used to assess the performance of the model on unseen data. Some of the most common evaluation functions are the F1-score and the BLEU score, commonly used in binary classification tasks [2] and translations [3] respectively.

  The F1-score is based on precision and recall, which are defined as follows [2]:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{1}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{2}$$

  The F1-score is given by:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

  The F1-score is useful because it penalizes imbalances between precision and recall, ensuring that a model is not rewarded for excelling in one while failing in the other. For example, a classifier that labels all instances as positive may achieve high recall but low precision, making it ineffective. The F1-score prevents such misleading results by considering both metrics together.
- **Overfitting:** This is the opposite of generalizing. When a model overfits, it means it performs well on the training data, but not on unseen data. It is conventionally said that the model memorized the training data.
- **Underfitting:** The model does not perform well on the training data or unseen data. It is usually assumed that the model either needs further training, or the nature of the data is more complex than what the model can capture (e.g., using a linear separator to classify non-linearly separable data).

Approaches in the field for training models can usually be categorized into supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. Supervised learning is the method used in this work and its core is using the labels of the data in the training process to continuously reduce the output of the loss function.

One of the most common models in machine learning is the neural network. Because of their high number of parameters, they provide great flexibility for complex tasks, a characteristic that also makes them usually require greater volumes of data than other classical machine learning algorithms such as Naive Bayes [4], a statistical model based on the Bayes Theorem [5],

## 2.2 Deep Learning

[2]

As explained in *Deep Learning* [6], modern deep learning is defined as a class of machine learning algorithms that construct complex architectures by composing simpler structures. However, most contemporary interpretations focus on neural networks [6].

Neural networks, because of their aforementioned properties, are suitable for learning patterns from complex high-dimensional data. Before starting with them, some machine learning concepts need to be defined more formally:

**Data point**

A mapping from data point to vector is needed for neural networks since most operations involve vector products. There are several ways to do this, but to illustrate this concept, and the next ones, a data point x with D features will be mapped to $\mathbb{R}^{D,1}$ (i.e. a column vector with D dimensions).

**Loss Function**

A loss function measures the discrepancy between the predicted output and the true label. Formally, a loss function for one data point is defined as:

$$\mathcal{L} : \mathbb{R}^{D,1} \times \mathbb{R}^{D,1} \to \mathbb{R} \tag{4}$$

where $D$ is the dimension of the data and the labels.

One of the most commonly used loss functions for classification tasks is the cross-entropy loss, which measures the difference between the predicted probabilities for each class and the actual class labels. Cross-entropy is especially useful in tasks where the model outputs a probability distribution over multiple classes.

In the multiclass setting, the cross-entropy loss is defined as:

$$L_{\text{CE}} = -\sum_{c=1}^{C} y_c \log(p_c) \tag{5}$$

where:

- $C$ is the total number of classes,

[2]Note that this section is based on the lecture materials of *6.036 Introduction to Machine Learning* course by MIT [1], unless stated otherwise.

- $y_c$ is a binary indicator (0 or 1) that denotes whether class $c$ is the true label for the input, and
- $p_c$ is the predicted probability that the input belongs to class $c$.

For binary classification, where the task is to distinguish between two classes (e.g., positive and negative), the cross-entropy loss simplifies to:

$$L_{\mathrm{CE}}(p, y) = -\left(y \log(p) + (1 - y) \log(1 - p)\right) \tag{6}$$

where:

- $y \in \{0, 1\}$ is the actual label (0 for the negative class and 1 for the positive class),
- $p$ is the predicted probability for the positive class.

**Parameters**

They are learned from data by the learning algorithm and are the main characters in making predictions after training. From this point on, every time the parameters are mentioned, it refers to a vector $\theta \in \mathbb{R}^{D,1}$, with D being the dimension of the data point.

**Gradient Descent**

An optimization algorithm, in which the gradient of the loss function is used to update the parameters to minimize the loss (output of the loss function). It can be pictured as going in the opposite direction of the gradient (the steepest descent). More formally, having $\nabla_\theta \mathcal{L}$, the update of the parameters can be thought of as

$$\theta :- \theta - \eta \nabla_\theta \mathcal{L} \tag{7}$$

with $\eta$ being a hyperparameter of the learning algorithm called the learning rate, which determines how big the update of $\theta$ will be.

### 2.2.1 Neural Networks

Neural Networks or Artificial Neural Networks are a machine learning model, whose architecture is inspired by the human brain. For brevity, the following descriptions in this subchapter will pertain to a simple type of neural networks called a Feed Forward Neural Network.

The architecture looks like the figure 1 and the figure 2 is a computer simulation of the neural network structure in the brain.

Both pictures show neurons or nodes propagating information.

The central component of a neural network is the artificial neural neuron (See Figure 3), whose inspiration in the human brain comes from the nerve cells (See Figure 4).

The neuron is the processing unit in the architecture. One of the earliest models of such a unit is the perceptron, introduced by Frank Rosenblatt [**block1962perceptron**],

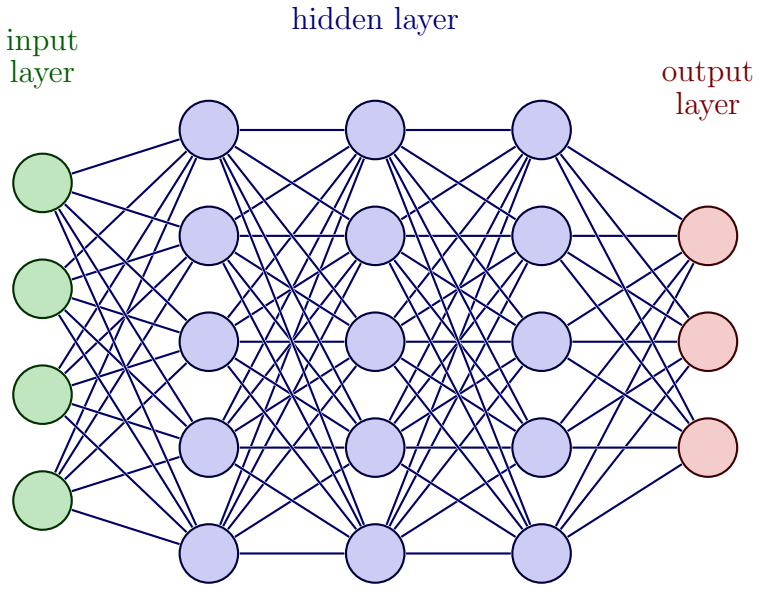


Figure 1: Feed Forward Neural Network [7]

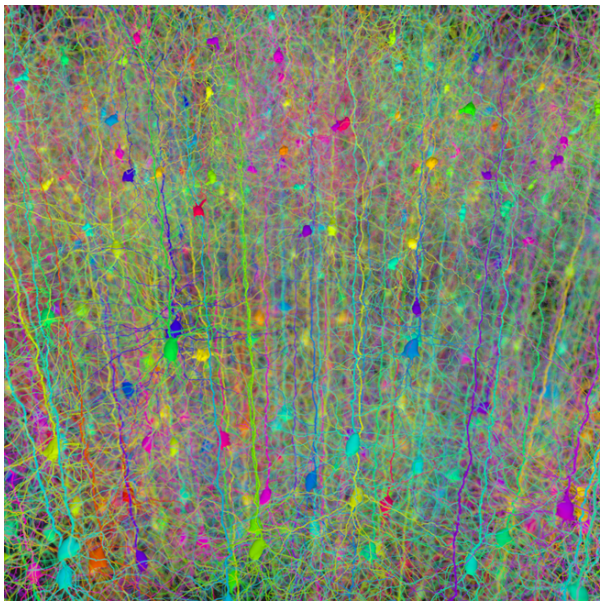

Figure 2: Brain Neural Network Simulation

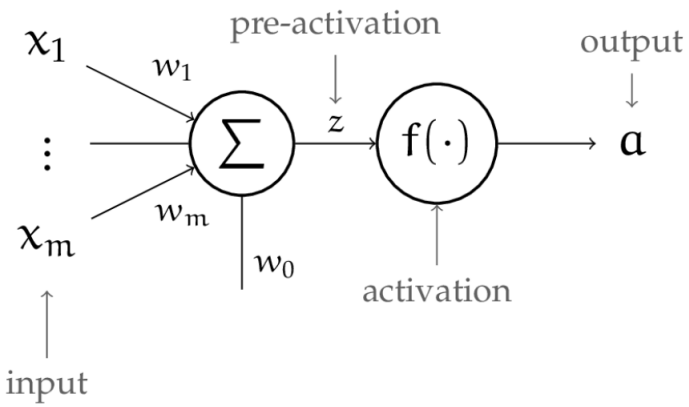


Figure 3: Artificial Neural Neuron [8]

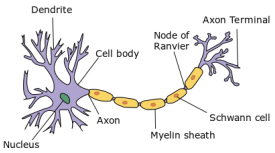


Figure 4: Human Neuron [9]

which performs a weighted sum of inputs. It is a linear classifier for binary classification that computes the sum:

$$z = \sum w_i x_i + b \tag{8}$$

where $w_i$ are the weights, $x_i$ are the inputs, and $b$ is the bias term, which prevents the decision boundary from being constrained to pass through the origin. The output is then determined by the function:

$$f(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases} \tag{9}$$

The neurons are arranged in layers. Each layer uses the previous layer's outputs for its calculations and then forwards its outputs to the next layer.

Neurons are arranged in layers. Each layer uses the previous layer's outputs for its calculations and then forwards its outputs to the next layer. For a given layer $l$, the computation follows:

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \tag{10}$$

$$a^{(l)} = \sigma(z^{(l)}) \tag{11}$$

where $W^{(l)}$ is the weight matrix, $a^{(l-1)}$ represents the outputs (activations) from the previous layer, $b^{(l)}$ is the bias vector, and $\sigma(\cdot)$ is the activation function.

There are typically three kinds of layers:

**Input Layer**

The input layer is a symbolic name since it refers to the input data points, with each node in the layer usually representing a feature of the data points. There is just one input layer.

**Hidden Layer**

This is where the bulk of operations take place. They usually compose the majority of the layers.

**Output Layer**

Aggregates the inputs of the last hidden layer making sense of the data to get an output, which should be interpretable as a result of the specific task (regression, classification).

For complex tasks usually many hidden layers are required in neural networks, which define and give the name to the field of deep learning.

### 2.2.2 Training

The following explanation refers specifically to supervised training.

Training a model can be viewed as a minimization problem where the goal is to minimize this loss function over time. This process of minimization is what enables the model to improve its predictions.

There are typically two stages in a training loop: forward pass and backpropagation.

**Forward Pass**

Defines the process of a set of inputs, called batches in the context of training, going from the input layer to the output layer, which is essentially the prediction process.

**Backpropagation**

An algorithm to update weights in all layers using gradient descent following the chain rule in order to propagate from the last layer to the first hidden layer.

## 2.3 Natural Language Processing

As the name implies, this field involves analyzing natural language (human language) and sometimes predicting from it. It is largely viewed as a subfield of machine learning since the most common approach used in modern Natural Language Processing (NLP) is deep learning.

### 2.3.1 Embeddings

Since the operations involved in training and prediction of neural networks are only defined with numerical data, a two-way representation between natural language and

tensors of numbers is needed. The numerical representation is usually called embedding.

The first step to process input text is breaking it up into tokens, which in modern approaches are usually sub-words, averaging about 4 characters in length in the English language [10].

These tokens are then usually mapped to IDs through the vocabulary, which is a look-up table mapping text tokens to numeric IDs.

Finally, additional contextual information is embedded into the IDS during training of most modern models in NLP. They are updated based on surrounding tokens and relative position. For example the token "how" could have different embeddings in the sentences "How are you?" and "That is how you do it".

## 2.4 Large Language Models

Refer to deep learning models focusing on NLP, mainly on text. They have a large number of parameters, which gives them their name and are currently the state-of-the-art solution in the field of Natural Language Processing. They are the current answer to the high complexity posed by natural language, such as cultural references, syntax, semantics, and nuance.

The current most used architecture is called the Transformer Architecture and it aims to tackle the lack of parallelization possible in the previous models, sequential in essence [11]. It reduces training time significantly, in a task like translation between languages, from several days [12] to as little as twelve hours.

## 2.5 Transformer Architecture

The main new breakthrough coming with the transformer architecture is the ability to process tokens in parallel while also keeping the dependencies of the ones far apart in the text input. This provides both a significant speed up in training as well as an increase in performance in certain tasks such as machine translation from English to German [11].

## 2.6 Transfer Learning

Refers to leveraging training in one task to train a machine learning model in another, usually related task. [13].

A special kind of transfer learning, called cross-lingual transfer, revolves around leveraging training in one language to improve the results of training in another language [14].

## 2.7 Continual Pre-training of Large Language Models

Refers to the addition of new data as knowledge base to an already trained large language model without merging the old dataset and the new one to start the training from scratch [15].

It consists mainly of the careful selection of a learning rate to minimize the loss on new data while maintaining the loss on the original data. This is a key factor in the process since the distribution shift that further training introduces can lead to a decrease in performance in previous (original) data. Research indicates that the right parameters might make continual pre-training better performing than training from scratch on the whole data [15].

### 2.8 Semantic Web

An initiative to make web content machine-readable by standardizing it [16].

Resource Description Framework (RDF) is one of the most popular frameworks for standardizing and representing web content. It was first introduced and encouraged by the World Wide Web Consortium (Organization developing the standards of the Web) in 1999 [17], proposing a graph structure linking web contents making the relationships between contents machine-readable, with SPARQL being the standard query language for this format.

but

## 3 Research Background

### 3.1 Transfer Learning

Although the idea of leveraging the knowledge of previous training in new ones was already introduced in neural networks in the 1970s [18], the first successful case of using it in NLP in the context of pre-trained LLMs was in 2015 [19].

With the introduction of GPT-3, it was shown to be possible to train an LLM on a "general-purpose text generation" downstream task, typically called fine-tuning [20]. Since then, transformer-based LLMs have been gaining in popularity and in performance because of the flexibility and relatively small set of data required to effectively train for the downstream tasks.

Different methods have been developed to cater to small datasets and low hardware resources, by lowering the tunable parameters and the model's precision [21].

### 3.2 Cross-Lingual Transfer

Before the rise in popularity of llms, Cross-lingual transfer had been continually studied within neural networks in the context of NLP for specific tasks, such as part-of-speech tagging [22].

One of the most common challenges in NLP is improving performance in low-resource languages (languages with small amounts of training data). Usual approaches include tackling the model at the word embedding level leveraging high resource languages, translating training data [23] and changing the model architecture [24].

With the rise of LLms, cross-lingual transfer kept being studied within its context, focusing initially on specific tasks such as part-of-speech tagging with zero-shot learning (no training samples) [25].

Architectural approaches such as adding adapter layers (trainable layers) to transformer-based models and training them are one of the popular techniques to improve zero-shot cross-lingual transfer [26] along with previously researched word embedding approaches that increase the similarity of word embeddings between languages [27].

Factors surrounding the degree of cross-lingual transfer developed by LLMs with multilingual pre-training have been studied before [28], and a surge in interest in it has been showing recently in current research [29].

### 3.3 Text-to-SPARQL

There are in general three approaches used for LLMs to extract knowledge from data in RDF format to answer questions in natural language [30].

The first one is to provide the dataset to the LLM along with the question, so that it answers the questions directly [30].

The second approach is to translate the question from natural language to a SPARQL query that then gets executed in the corresponding engine. This approach often relies on prompt engineering [31] (structuring of prompt to leverage Llm capabilities [32]).

The third approach is a combination of the first two. It involves getting the right parts of the data to feed them to the LLM to generate the SPARQL query with prompt engineering [33].

## 4 Contribution

## 5 Methodology

### 5.1 Data Collection

Which datasets did we draw from?

Data is drawn from all of the QALD (Question Answering over Linked Data) challenges until now, from the first edition to the tenth edition. The goal of this challenge is for teams to compete to extract information from knowledge graphs in the most accurate way [34], aligning well with the first goal of creating a multilingual dataset for text-to-SPARQL generation.

The data from the QALD challenges were combined with most of the different datasets gathered by Jian et al. [35] from different challenges regarding question-answering over knowledge graphs. Specifically, the challenges drawn upon are those from Talmor et al. [36], Cui et al. [37], Gu et al. [38], Su et al. [**s**, 39], Trivedi et al. [40], Dubey et al. [41], Kaffee et al. [42], Korablinov et al. [43], Rybin et al. [44] and Yih et al. [45]. The datasets in consideration meet the basic requirement of having the following properties available or easily inferable for each question: question in natural language, SPARQL query, language and knowledge graph referenced.

For the QALD datasets, a script was made to scrape the main QALD repository (https://github.com/ag-sc/QALD/tree/master), which has the QALD challenges from 1 to 9, with the data of all challenges, to check that the schema meets the basic requirements for the task, and merge them into the standardized schema with the following columns: text_query, language, sparql_query and knowledge_graphs. Since the datasets in the rest of the challenges, with the exception of QALD 10, had much more varied forms, each dataset had to be manually examined to check the schema and a custom script was made to extract all of its partitions, adapt the schema, and combine with the previously merged dataset.

### 5.2 Data Analysis

The QALD datasets sum up to about 16000 data points, while the rest make up around 620000. One dataset stands out because of its size, the Multilingual Compositional Wikidata Questions dataset by Cui et al. [37] with around 100000 questions in each of the following languages: English, Hebrew, Kannada and Chinese. It is worth pointing out that the non-English questions were translated from English using Google Translation.

Regarding the data quality, the QALD 10 dataset stands out for being a multilingual dataset curated by native speakers, It provides 806 questions, each one in Chinese (Mandarin), English, German and Russian with the corresponding SPARQL query for the knowledge graph Wikidata.

Numbers. How big were the QALD datasets in total, how big were the others, were some of them particularly big?

## 6 Experimental Setup

### 6.1 Data Processing

### 6.2 Training

## 7 Results

### 7.1 Mistral

### 7.2 Occiglot

common small errors and where do they come from

## 8 Discussion

### 8.1 Comparison

### 8.2 Interpretation

talk about the length of the queries and its relationship with the quality and which

## 9 Conclusion

# References


[1] Massachusetts Institute of Technology. *Introduction to Machine Learning (6.036)*. https://openlearninglibrary.mit.edu/courses/course-v1:MITx+6.036+1T2019/course/. Accessed: 2025-03-08. 2020.

[2] Zachary Chase Lipton, Charles Elkan, and Balakrishnan Narayanaswamy. "Thresholding classifiers to maximize F1 score". In: *arXiv preprint arXiv:1402.1892* (2014).

[3] Mozhgan Ghassemiazghandi. "An Evaluation of ChatGPT's Translation Accuracy Using BLEU Score". In: *Theory and Practice in Language Studies* 14.4 (2024), pp. 985–994.

[4] FY Osisanwo et al. "Supervised machine learning algorithms: classification and comparison". In: *International Journal of Computer Trends and Technology (IJCTT)* 48.3 (2017), pp. 128–138.

[5] Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. "NaıHERE!HERE!ve Bayes." In: *Encyclopedia of machine learning* 15.1 (2010), pp. 713–714.

[6] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[7] TikZ.net. *Neural Networks*. https://tikz.net/neural_networks/. Accessed: 2024-12-09. n.d.

[8] MIT Open Learning Library. *Neural Networks - Basic Element*. Massachusetts Institute of Technology, accessed on December 10, 2024. 2019. URL: https://openlearninglibrary.mit.edu/.

[9] SEER Training Modules. *Neurons and Glial Cells - Brain and CNS Tumors*. U.S. National Institutes of Health, National Cancer Institute. Accessed on December 10, 2024. 2024. URL: https://training.seer.cancer.gov/brain/tumors/anatomy/neurons.html.

[10] OpenAI. *What are tokens and how to count them?* https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them. Accessed: 2024-12-11. 2024.

[11] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[12] Dzmitry Bahdanau. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[13] Lisa Torrey and Jude Shavlik. "Transfer learning". In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[14] *Cross-Lingual Transfer*. https://paperswithcode.com/task/cross-lingual-transfer. Accessed: 2024-12-17. 2024.

[15] Kshitij Gupta et al. "Continual Pre-Training of Large Language Models: How to (re) warm your model?" In: *arXiv preprint arXiv:2308.04014* (2023).

[16] Tassilo Pellegrini and Andreas Blumauer. "Semantic web". In: *Wege zur vernetzten Wissensgesellschaft. Berlin [ua] Springer* (2006).

[17] data.europa.eu. *Introduction to RDF & SPARQL*. Online. RDF was published as a W3C recommendation in 1999. RDF was originally introduced as a data model for metadata. RDF was generalised to cover knowledge of all kinds. 2014. URL: https://data.europa.eu/sites/default/files/d2.1.2_training_module_1.3_introduction_to_rdf_sparql_en_edp.pdf.

[18] Stevo Bozinovski. "Reminder of the first paper on transfer learning in neural networks, 1976". In: *Informatica* 44.3 (2020).

[19] Xu Han et al. "Pre-trained models: Past, present and future". In: *AI Open* 2 (2021), pp. 225–250. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.08.002. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000231.

[20] Haifeng Wang et al. "Pre-Trained Language Models and Their Applications". In: *Engineering* 25 (2023), pp. 51–65. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2022.04.024. URL: https://www.sciencedirect.com/science/article/pii/S2095809922006324.

[21] Venkatesh Balavadhani Parthasarathy et al. "The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities". In: *arXiv preprint arXiv:2408.13296* (2024).

[22] Joo-Kyung Kim et al. "Cross-Lingual Transfer Learning for POS Tagging without Cross-Lingual Resources". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2832–2838. DOI: 10.18653/v1/D17-1302. URL: https://aclanthology.org/D17-1302.

[23] Sebastian Schuster et al. "Cross-lingual transfer learning for multilingual task oriented dialog". In: *arXiv preprint arXiv:1810.13327* (2018).

[24] Jonas Pfeiffer et al. "Mad-x: An adapter-based framework for multi-task cross-lingual transfer". In: *arXiv preprint arXiv:2005.00052* (2020).

[25] David Ifeoluwa Adelani et al. "Comparing LLM prompting with Cross-lingual transfer performance on Indigenous and Low-resource Brazilian Languages". In: *arXiv preprint arXiv:2404.18286* (2024).

[26] Masayasu Muraoka et al. "Cross-Lingual Transfer of Large Language Model by Visually-Derived Supervision Toward Low-Resource Languages". In: *Proceedings of the 31st ACM International Conference on Multimedia*. MM '23. Ottawa ON, Canada: Association for Computing Machinery, 2023, pp. 3637–3646. ISBN: 9798400701085. DOI: 10.1145/3581783.3611992. URL: https://doi.org/10.1145/3581783.3611992.

[27] Pavel Efimov et al. “The impact of cross-lingual adjustment of contextual word representations on zero-shot transfer”. In: *European Conference on Information Retrieval*. Springer. 2023, pp. 51–67.

[28] Yoshinari Fujinuma, Jordan Boyd-Graber, and Katharina Kann. “Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability”. In: *arXiv preprint arXiv:2203.10753* (2022).

[29] Hetong Wang, Pasquale Minervini, and Edoardo M Ponti. “Probing the Emergence of Cross-lingual Alignment during LLM Training”. In: *arXiv preprint arXiv:2406.13229* (2024).

[30] Caio Viktor S Avila et al. “Experiments with text-to-SPARQL based on ChatGPT”. In: *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*. IEEE. 2024, pp. 277–284.

[31] Hamada M Zahera et al. *Generating sparql from natural language using chain-of-thoughts prompting*. 2024.

[32] Amazon Web Services (AWS). “What is Prompt Engineering?” In: (2025). Accessed: 2025-01-07. URL: https://aws.amazon.com/what-is/prompt-engineering/.

[33] Vincent Emonet et al. “LLM-based SPARQL Query Generation from Natural Language over Federated Knowledge Graphs”. In: *arXiv preprint arXiv:2410.06062* (2024).

[34] Vanessa Lopez et al. “Evaluating question answering over linked data”. In: *Journal of Web Semantics* 21 (2013). Special Issue on Evaluation of Semantic Technologies, pp. 3–13. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2013.05.006. URL: https://www.sciencedirect.com/science/article/pii/S157082681300022X.

[35] Longquan Jiang and Ricardo Usbeck. “Knowledge Graph Question Answering Datasets and Their Generalizability: Are They Enough for Future Research?” In: *arXiv preprint arXiv:2205.06573* (2022).

[36] Alon Talmor and Jonathan Berant. “The Web as a Knowledge-Base for Answering Complex Questions”. In: *NAACL*. 2018.

[37] Ruixiang Cui et al. “Multilingual Compositional Wikidata Questions”. In: *arXiv e-prints*, arXiv:2108.03509 (Aug. 2021), arXiv:2108.03509. arXiv: 2108.03509 [cs.CL].

[38] Yu Gu et al. “Beyond IID: three levels of generalization for question answering on knowledge bases”. In: *Proceedings of the Web Conference 2021*. ACM, pp. 3477–3488.

[39] Yu Su et al. “On Generating Characteristic-rich Question Sets for QA Evaluation”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 562–572. DOI: 10.18653/v1/D16-1054. URL: https://www.aclweb.org/anthology/D16-1054.

[40] Priyansh Trivedi et al. “Lc-quad: A corpus for complex question answering over knowledge graphs”. In: *International Semantic Web Conference*. Springer. 2017, pp. 210–218.

[41] Mohnish Dubey et al. “LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia”. In: *Proceedings of the 18th International Semantic Web Conference (ISWC)*. Springer. 2019.

[42] Lucie-Aimée Kaffee et al. “Ranking Knowledge Graphs By Capturing Knowledge about Languages and Labels”. In: *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*. ACM, 2019. URL: https://doi.org/10.1145/3360901.3364443.

[43] Vladislav Korablinov and Pavel Braslavski. “RuBQ: A Russian Dataset for Question Answering over Wikidata”. In: *arXiv e-prints*, arXiv:2005.10659 (May 2020), arXiv:2005.10659. arXiv: 2005.10659 [cs.CL].

[44] Ivan Rybin et al. “Ru{BQ} 2.0: An Innovated Russian Question Answering Dataset”. In: *Eighteenth Extended Semantic Web Conference - Resources Track*. 2021. URL: https://openreview.net/forum?id=P5UQFFoQ4PJ.

[45] Wen-tau Yih et al. “The Value of Semantic Parse Labeling for Knowledge Base Question Answering”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2016, pp. 201–206.

# Appendix

## Declaration of academic integrity

I hereby declare that all written work stems from original ideas, and all of which originate from external sources, is correctly referenced and/or cited.

Berlin, the March 11, 2025 .....