🧪LinguAlchemy: Fusing Typological and Geographical Elements for Unseen Language Generalization

Anonymous ACL submission

Abstract

Pretrained language models (PLMs) have become remarkably adept at task and language generalization. Nonetheless, they often fail dramatically when faced with unseen languages, posing a significant problem for diversity and equal access to PLM technology. In this work, we present LINGUALCHEMY, a regularization technique that incorporates various aspects of languages covering typological, geographical, and phylogenetic constraining the resulting representation of PLMs to better characterize the corresponding linguistics constraints. LINGUALCHEMY significantly improves the accuracy performance of mBERT and XLM-R on unseen languages by ∼18% and ∼2%, respectively compared to fully fine-tuned models and displaying a high degree of unseen language generalization. We further introduce ALCHEMYSCALE and ALCHEMYTUNE, extension of LINGUALCHEMY which adjusts the linguistic regularization weights automatically, alleviating the need for hyperparameter search. LINGUALCHEMY enables better cross-lingual generalization to unseen languages which is vital for better inclusivity and accessibility of PLMs.

1 Introduction

Significant advancements in language processing technology have been achieved through the development of PLMs, leading to a commendable proficiency in language comprehension and generation (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020; Sanh et al., 2022; Lewis et al., 2019; Raffel et al., 2023; Li et al., 2021; Cahyawijaya et al., 2021; Wilie et al., 2020). However, there remains a notable deficiency in the ability of these models to generalize effectively to unseen languages, resulting in a considerable performance reduction of PLMs across thousands of unseen languages. To mitigate this problem, efforts to develop efficient language adaptation approaches are underway, focusing on the incorporation of these unseen languages to PLMs (Pfeiffer et al., 2021b; Alabi et al., 2022; Ebrahimi et al., 2022; Goyal et al., 2021).

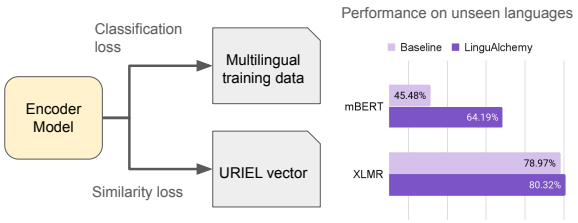

Figure 1: LINGUALCHEMY enhances performance in unseen languages by allowing the model to predict the linguistic vector and then fitting it via a similarity loss towards the specific language's URIEL vector.

Incorporating new unseen languages has been a longstanding problem in natural language processing (NLP) research, especially given that most of these unseen languages are low-resource and underrepresented making PLMs difficult to adapt to these languages. MAD-X (Pfeiffer et al., 2020b) employs a language adapter to learn new unseen languages by incorporating language adapters that mitigate the risk of forgetting pre-trained knowledge which is known as the curse-of-mulitlinguality. Nonetheless, this approach requires training for generalizing to new unseen languages which makes it costly and difficult to scale to thousands of languages. MAD-G (Ansell et al., 2021) and Udapter (Üstün et al., 2020) further generalize this approach by utilizing a linguistic-driven contextual parameter generator (CPG) module to generate language-specific parameters allowing the models to generalize to other languages with similar linguistic characteristics. Recently, Rathore et al. (2023) introduced ZGUL, which combines representations over multiple language adapters to generate the unseen language representation. Despite the effectiveness, all these approaches largely rely on two assumptions, i.e., 1) strict categorization of languages and 2) knowing the language category of

the query apriori. The first assumption disregards the fact that linguistic phenomena such as code-mixing may occur in the query. While the second assumption might cause performance degradation due to the error propagation from the language identification module (Adilazuarda et al., 2023).

To overcome those limitations, in this work, we introduce LINGUALCHEMY. Unlike adapter-based approaches which isolate the capability to understand different languages over multiple language-specific adapters, LINGUALCHEMY learns a shared representation across multiple languages, allowing the model to leverage shared knowledge between languages. Such behavior is attainable by eliminating the language-specific module from the model and instead utilizing a regularization to learn language-specific knowledge. LINGUALCHEMY allows the model to perform inference without knowing the language of the query prior. Our evaluation suggests that LINGUALCHEMY can improve mBERT and XLM-R generalization in unseen languages while maintaining the performance on the high-resource languages without requiring information on the language of the query apriori.

In summary, our contributions are as follows:

1. We propose LINGUALCHEMY, a regularization method that improves unseen language performance on language models and aligns them to arbitrary languages.

2. We demonstrate strong performance on unseen languages for models trained with LINGUALCHEMY.

3. We introduced a dynamic scaling method to scale the classification and auxiliary loss factors used in the fine-tuning stage.

2 Related Works

PLMs with their transformer-based architectures have been demonstrating exceptional capabilities in language comprehension and generation. These models excel in abstract linguistic generalization by capturing complex linguistic patterns and understanding structural positions and thematic roles, which are crucial for interpreting language semantics. Research in this area (Ganesh et al., 2021) has provided critical insights that enable these models to process and generate human language effectively. The studies have explored how these models grasp intricate linguistic features, including syntax and semantics, thereby enhancing their performance across a wide range of language tasks (Rathore et al., 2023).

In parallel, the development of resources like publicly available URIEL vector and lang2vec utility (Littell et al., 2017) has been instrumental in extending the reach of multilingual NLP, particularly for less-resourced languages. These tools provide vector representations of languages, leveraging typological, geographical, and phylogenetic data, thus offering a structured approach to understanding linguistic diversity. Complementing this, recent research has conducted a comprehensive survey on the utilization of typological information in NLP, highlighting its potential in guiding the development of multilingual NLP technologies (Ponti et al., 2019). This survey emphasized the underutilization of typological features in existing databases and the need for integrating data-driven induction of typological knowledge into machine learning algorithms.

Recent advancements in prefix tuning (Li and Liang, 2021) and subspace learning (Zhang et al., 2020) have contributed significantly to improving generalization in PLMs. These methods focus on learning prefix subspaces to stabilize the direct learning of embeddings, addressing the instability issues present in earlier approaches. Two notable methods are the MAD-X (Pfeiffer et al., 2020b) and MAD-G (Ansell et al., 2021) frameworks, which employ adapter-based techniques for multi-task cross-lingual transfer, highlighting modularity and parameter efficiency. However, it inherits the limitations of the pretrained multilingual models, such as the limited capacity to adapt effectively to low-resource and unseen languages. Furthermore, while the framework facilitates adaptation to specific target languages, it may bias the model towards these languages, potentially impacting its performance on other languages.

However, despite these advancements, PLMs still face significant challenges in generalizing to unseen languages, particularly when adapting to low-resource and unseen languages. These challenges stem from the vast structural and semantic variation across languages (Bender, 2011; Jurafsky and Martin, 2019), the scarcity of resources (Mohammad, 2019; Lewis et al., 2020), and the limitations inherent in the models themselves (Lin et al., 2017). This situation highlights the complexity of scaling and generalizing these models effectively

and underscores the need for more sophisticated approaches in model training and adaptation to ensure broader and more equitable language coverage.

3 Unseen Languages Adaptation with LINGUALCHEMY

In this section, we provide an overview of how LINGUALCHEMY can capture linguistic constraint and how is the intuition behind LINGUALCHEMY. We also discuss in detail how do we align model representations with the linguistic vector.

3.1 Does Multilingual LMs capture Linguistic Constraint?

In this work, we define the linguistic knowledge as a vector gathered from URIEL vector (Littell et al., 2017). We chose three distinct linguistic knowledge from the database, namely `'syntax_knn'`, `'syntax_average'`[1], and `'geo'` features. The choice of `'syntax_knn'` and `'syntax_average'` is motivated by the typological nature of syntax. Syntax in languages varies widely; hence, by using aggregate measures like averages and k-nearest neighbors (kNN), we can capture a more general representation of syntactic features across languages. These features include consensus values, like averages, and predicted values, such as kNN regressions based on phylogenetic or geographical neighbors.

We excluded phonological features, as our focus is not on speech but on textual data. Phonology primarily pertains to spoken languages and would not add significant value to our analysis of written language structures. Additionally, we excluded language family features (`'family'`). These features are typically binary, indicating whether a language belongs to a particular subfamily. While they can be useful for language identification tasks, they tend to be sparse and may not provide the granularity needed for our study. By focusing on syntax and geographical features, our approach aims to encompass a broader and more nuanced understanding of linguistic variations in a multilingual context.

Syntax Feature These feature vectors denote a typological feature that is adapted from several sources including World Atlas of Language Structures (WALS), Syntactic Structures of World Languages (Collins, 2010), and short prose descriptions on typological features in Ethnologue (Lewis, 2009). Syntax vectors captures information about the syntactic properties of languages which derived from large-scale typological databases, which document the structural and semantic variation across different languages. These syntax features in the URIEL vector are utilized to represent languages in vector form, allowing for the analysis and comparison of languages based on their syntactic properties.

Geographical Feature On the other hand, geographical features represent languages in terms of their geographical properties. The inclusion of `'geo'` features aims to capture geographical attributes of languages. Geographic factors can significantly influence language evolution and structure, making them crucial for understanding linguistic variations. This feature expresses geographical location with a fixed number of dimensions that each represents the "great circle" distance—from the language in question to a fixed point on the Earth's surface. By incorporating geographical information into language vectors, URIEL and lang2vec provide a more comprehensive view of languages, considering not only their structural and semantic properties but also their geographical context.

3.2 Proof of Concept

Linguistic Separability in LMs To explore whether multilingual language models (MLMs), such as Multilingual BERT (mBERT), capture the linguistic constraints as defined by the URIEL vectors, we align the mBERT language embeddings to the linguistic knowledge vectors. This projection is quantitatively measured to assess the representation of both seen and unseen languages within the model. The fundamental question we address is the extent to which mBERT's embeddings correspond to the typological and geographical features encapsulated in the URIEL vectors.

Figure 2 presents a visual analysis facilitated by UMAP (McInnes et al., 2018), showing the correlation between mBERT language representation and the linguistic vectors from the URIEL database (R^2 = 0.816). By leveraging UMAP, the plot accentuates the principal variances within the joint feature space of the embeddings and vectors. The spatial representation of languages on this plot mirrors their linguistic and geographical relatedness,

[1] In this work, we chose the `'knn'` and `'average` syntax features. These include consensus values (like averages) and predicted values (such as kNN regressions based on phylogenetic or geographical neighbors)

Figure 2: Alignment between mBERT Representation with URIEL Language Representation. The green-shaded areas indicate the sentence representations of mBERT while the brown dots represent the URIEL representations of the corresponding language.

as encapsulated by mBERT. This visualization underscores the model's ability to mirror linguistic typologies, with languages sharing common roots such as 'de-DE' and 'nl-NL' naturally clustering together. The density and arrangement of these clusters potentially reflect mBERT capacity to capture and represent language family traits. Conversely, the presence of sparser clusters or outliers prompts a closer examination of mBERT's coverage and consistency in representing diverse linguistic features. We also formally defined the language representation alignment in the Appendix B.

3.3 LINGUALCHEMY

We introduce LINGUALCHEMY as an approach that intuitively aligns model representations with linguistic knowledge, leveraging URIEL vectors. This is operationalized through an auxiliary loss function, involving the training process with a nuanced understanding of linguistic characteristics.

In LINGUALCHEMY, we enhance the fine-tuning of encoder models such as mBERT for downstream tasks by not only using the regular classification loss but also introducing a novel linguistic regularization term. This is achieved through the implementation of a URIEL loss, designed to align the model's representations with linguistic knowledge derived from URIEL vectors. Specifically, this process involves applying a linear projection to the model's pooled output, which aligns it with the URIEL vector space. The URIEL loss is quantified as the mean squared error (MSE) between the projected model outputs and the corresponding URIEL vectors. This dual approach, combining classification loss and URIEL loss, allows for a more linguistically informed model training, enhancing the model's ability to capture and reflect complex linguistic patterns and relationships.

$$L_{uriel}(R, U) = \frac{1}{N} \sum_{i=1}^{N} \|R_i - U_i\|^2$$

where R represents the model-generated representations, U denotes the URIEL vectors, and N is the number of data points. To generate the model representation, we take the output representation from the CLS token and multiply it with a new, trainable projection layer to transform the vector size so that they are compatible.

Note that there may be discrepancies between the scales of the standard classification loss and the URIEL loss. To address this, we introduce an optional hyperparameter, denoted as λ, to scale the URIEL loss appropriately.

Dynamic Scaling Approaches In addition to the fixed scaling factor, we also explore dynamic adjustment of this scaling factor at each training step. This aims to maintain a balance between the classification and URIEL losses, and even considers making the scale trainable. The final loss formula when training with LINGUALCHEMY is given by:

$$L = \lambda_{cls} * L_{cls} + \lambda_{uriel} * L_{uriel}(R, U)$$

We define two methods to implement dynamic scaling:

1. **ALCHEMYSCALE**: This method dynamically adjusts the scaling factor λ during training. It initiates with scaling factors set relative to the mean of initial losses, ensuring proportional importance to each loss component. Subsequently, these factors are updated periodically using an Exponential Moving Average (EMA) method, with a conservative adjustment for specific loss components to maintain stability. This approach ensures an optimal balance between different loss components, adjusting the contribution of each to the total loss based on predefined criteria.

2. **ALCHEMYTUNE**: Here, λ is conceptualized as a trainable parameter within the model's architecture. Initialized as part of the model's

parameters, λ undergoes optimization during the training process. This method applies the scaling factors to loss components, and an additional *mini_loss*—representing the deviation of the sum of scaling factors from unity—is computed. This *mini_loss* is used for backpropagation, enabling the scaling factors to adapt based on the training dynamics and dataset specifics.

Both methods aim to enhance model performance by dynamically and intelligently scaling loss components, with the first method relying on predefined, periodically updated scaling mechanisms, and the second integrating the scaling factor into the model's learning parameters for adaptive adjustments.

4 Experiment Setting

Datasets In our experiments, we utilize the publicly available MASSIVE Dataset (Xu et al., 2022), which is a comprehensive collection of multilingual data incorporating intent classification tasks. This dataset is particularly notable for its inclusion of a diverse range of languages, with various language families, genera, and scripts. Specifically, we split MASSIVE into 25 languages that are 'seen' during finetuning and the rest 27 languages that are 'unseen', which we exclusively used for evaluation. This splitting is based on the language adapters availability as outlined in the prior research of (Pfeiffer et al., 2020a), which we utilized in the AdapterFusion experiment for our baseline model. For a detailed breakdown of the languages used, including their respective families, genera, and script can be found in Appendix A.

Models Our research employs two state-of-the-art language models: Multilingual BERT Base (mBERT$_{\text{BASE}}$) and XLM-RoBERTa Base (XLM-R$_{\text{BASE}}$). These models are chosen for their robust performance across a wide range of multilingual NLP tasks, making them ideal for our intent classification and language identification objectives.

Hyperparameters The following hyperparameters are used in our training process:

- Learning rate: 5×10^{-5}
- Training epochs: 30
- Performance metric: Accuracy
- Early stopping criterion: 3 epochs without improvement

Each training takes about 5 hours using a single A100 GPU.

5 Results and Discussion

5.1 LINGUALCHEMY Performance

To evaluate the effectiveness of our proposed technique on unseen languages, we trained mBERT and XLM-R on the MASSIVE dataset. Specifically, we train on 25 languages and test on 27 different, unseen languages. Our results are summarized in Table 1. We compared our method with zero-shot generalization, where the model is fully tuned on seen languages and then tested on unseen languages (referred to as Full FT in the Table). Additionally, we explored AdapterFusion (Pfeiffer et al., 2021a) as another baseline. AdapterFusion has shown better adaptation to unseen languages than naive zero-shot generalization. Unfortunately, many language adapters that we need for AdapterFusion is not available for XLM-R.

From Table 1, it is shown that LINGUALCHEMY achieves better generalization for unseen languages. We observed a significant improvement for mBERT and a modest average improvement for the stronger XLM-R model. For mBERT, LINGUALCHEMY can significantly increase performance in truly unseen languages of am-ET, km-KH, mn-MN, in which mBERT has never seen during the pre-training stage nor fine-tuning. These findings show that LINGUALCHEMY can be useful in truly zero-shot settings.

We note that LINGUALCHEMY tends to even out the performance across all unseen languages, leading to a massive boost in weaker languages such as cy-GB or sw-KE. However, in languages where zero-shot performance is already strong, LINGUALCHEMY does not seem to provide benefits and in some cases degrade performance. This is more evident in XLM-R, where LINGUALCHEMY flattens performance to the 80-82% range, in which many of zero-shot performance is already reached that level or more. Regardless, the potential of our method is clear, showing it to be beneficial in cases where zero-shot performance is poor.

Method	Unseen Language Performance													
	am-ET*	cy-GB	af-ZA	km-KH*	sw-KE	mn-MN*	tl-PH	kn-IN	te-IN	sq-AL	ur-PK	az-AZ	ml-IN	ms-MY
mBERT														
AdapterFusion	4.6%	25.1%	**57.7%**	7.8%	22.2%	27.6%	40.3%	41.0%	34.4%	49.5%	47.1%	63.8%	35.8%	65.8%
Zero-shot CL	5.5%	23.8%	52.7%	8.3%	19.8%	27.2%	37.5%	34.2%	35.3%	44.8%	42.8%	61.6%	27.7%	66.5%
Ours	**58.1%**	**30.0%**	50.2%	**59.9%**	**54.9%**	**57.4%**	**66.5%**	**67.8%**	**71.9%**	**70.7%**	**69.4%**	**69.2%**	**67.8%**	**67.9%**
XLM-R														
Zero-shot CL	**78.6%**	64.4%	**82.7%**	**84.6%**	58.1%	**87.5%**	**85.9%**	80.5%	**84.6%**	67.9%	73.6%	80.2%	78.9%	**83.0%**
Ours	77.0%	**69.0%**	75.7%	78.7%	**74.9%**	76.3%	80.4%	**81.2%**	82.6%	**82.2%**	**81.8%**	**82.0%**	**81.8%**	81.8%
	ca-ES	sl-SL	sv-SE	ta-IN	nl-NL	it-IT	he-IL	pl-PL	da-DK	nb-NO	ro-RO	th-TH	fa-IR	Average
mBERT														
AdapterFusion	**73.1%**	49.3%	64.1%	41.7%	70.0%	**71.9%**	51.2%	62.3%	71.3%	**68.8%**	58.7%	30.4%	59.4%	48.0%
Zero-shot CL	**73.1%**	47.2%	60.1%	34.9%	**70.7%**	70.8%	48.2%	60.0%	**71.7%**	68.5%	54.2%	24.2%	56.9%	45.5%
Ours	68.4%	**68.5%**	**68.4%**	**68.6%**	68.6%	68.1%	**68.1%**	**67.1%**	66.4%	65.7%	**64.9%**	**64.4%**	**64.4%**	**64.2%**
XLM-R														
Zero-shot CL	**87.4%**	**86.3%**	**85.4%**	**84.4%**	82.0%	78.3%	**88.7%**	61.3%	76.5%	78.2%	**82.8%**	73.3%	77.2%	79.0%
Ours	82.0%	82.2%	82.2%	82.4%	**82.3%**	**82.1%**	82.3%	**81.6%**	**81.4%**	**81.3%**	81.3%	**81.1%**	**81.0%**	**80.3%**

Table 1: Performance of LINGUALCHEMY in MASSIVE dataset for unseen languages. These models have not seen these languages during fine-tuning. For languages in *, mBERT model also never seen the languages during pre-training.

5.2 Effect of Scaling URIEL loss

The classification and URIEL losses are not on the same scale. Therefore, simply adding both losses together means that the model will give more weight to the loss with the higher magnitude. When observing both the classification and URIEL losses during the early stages of training, we note that the classification loss is around 10 times larger than the URIEL loss. In this part, we explore the effect of different scaling factors for the URIEL loss.

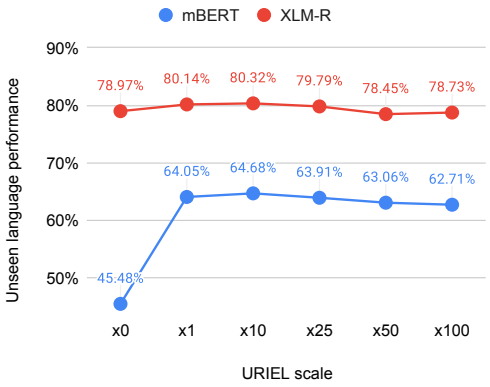

Figure 3: Average performance of unseen languages under various URIEL loss scales.

Constant Scaling We explore consistently scaling up the URIEL loss across various scaling factors. The results can be seen in Figure 3. It is important to note that, as we use the scale-invariant optimizer AdamW, we don't have to worry about gradients becoming too large due to extremely large losses. Generally, we observe that a scaling factor of 10x slightly outperforms other scaling factors, and the performance appears to decline with higher scale factors.

Dynamic and Trainable Scaling One issue with introducing a scaling factor is the addition of another tunable hyperparameter. Intuitively, we might aim for a balanced weight between the classification and URIEL losses. Therefore, instead of expensively testing different scaling factors, an adaptive scaling factor might be more cost-effective and beneficial. Here, we explore two ideas: dynamic and trainable factors. The results of these approaches can be seen in Table 2.

URIEL scaling	mBERT	XLM-R
Constant 10x	**64.68%**	80.32%
ALCHEMYSCALE	62.97%	**80.43%**
ALCHEMYTUNE	63.24%	79.10%

Table 2: Performance Comparison Across Different URIEL Scaling Methods.

Interestingly, these dynamic scale factors do not significantly outperform a constant factor. In contrast, a 10x scaling achieves the best performance in mBERT, while dynamic scaling barely outperforms the 10x scaling in XLM-R. Therefore, in a limited budget scenario, a suggested 10x scaling factor should suffice, and one may explore different scaling factors given more computational resources.

5.3 Generalization Across Language Family

We investigate LINGUALCHEMY across language families to further analyze the generalization capabilities of BERT and XLM-R models. This experiment offers insight into how adaptable LIN-

Train Group	Lang. Family	Languages	Num. Languages
1	Indo-European	af-ZA, bn-BD, ca-ES, cy-GB, da-DK, de-DE, el-GR, en-US, es-ES, fa-IR, fr-FR, hi-IN, hy-AM, is-IS, it-IT, lv-LV, nb-NO, nl-NL, pl-PL, pt-PT, ro-RO, ru-RU, sl-SL, sq-AL, sv-SE, ur-PK	26
2	Dravidian	Train Group 1 + kn-IN, ml-IN, ta-IN, te-IN	30
3	Afro-Asiatic	Train Group 2 + am-ET, ar-SA, he-IL	33
4	Sino-Tibetan	Train Group 3 + my-MM, zh-CN, zh-TW	36
Unseen Languages		sw-KE, km-KH, vi-VN, id-ID, jv-ID, ms-MY, tl-PH, ja-JP, ka-GE, ko-KR, mn-MN, th-TH, az-AZ, tr-TR, fi-FI, hu-HU	16

Table 3: Language family distribution used in the language family generalization experiment (§5.3)

GUALCHEMY is to a variety of linguistic features. We perform our experiment by splitting the languages in MASSIVE according to their language families and train the model on a subset of language families while testing on the rest, unseen language families. We explore on including different subset of language families, as seen in Table 3.

The "others unseen" category includes additional language families not incorporated in the training set, serving as an "unseen" testbed. As illustrated in Figure 4, LINGUALCHEMY demonstrates generalization towards these unseen language families. Perhaps unsurprisingly, adding more languages and, importantly, diversity to the training data improves generalization performance. Notably, the inclusion of the Afro-Asiatic language group—consisting of languages such as 'am-ET,' 'ar-SA,' and 'he-IL,' each featuring unique scripts—has significantly enhanced performance from the second to the third training group iteration. This improvement underscores LINGUALCHEMY's capability to adapt to scripts not presented during the initial training or fine-tuning phases, such as the Hebrew script of 'he-IL' and the Ethiopian script of 'am-ET,' further illustrating its robustness in generalizing across different scripts.

The performance of both models, combined with LINGUALCHEMY underscores the advantage of including a broader spectrum of languages within training groups for enhanced model generalization. However, the impact of this diversity is not uniform across all language families: While some consistently benefit from the expansion of training data, others do not, indicating that merely increasing the volume of data from the same family may not necessarily improve performance. This inconsistency indicates the potential limitations within the models' capacity to learn and generalize the linguistic features specific to certain language families. Consequently, our observation shows that the degree

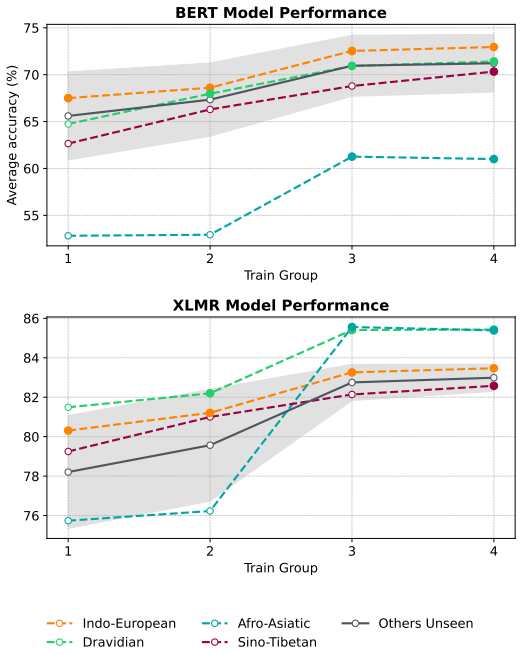

Figure 4: Model performance across language families. Dotted lines indicates language families used in training in some of the training stages (solid dots for active use– refer to Table 3), and solid grey lines for families unseen in all training stages, with variance shown in shading.

of generalization varies noticeably among different families, suggesting that while some may significantly profit from these models' capabilities, others may require more tailored strategies to gain similar performance improvement.

5.4 Seen Language Performance

One drawback of LINGUALCHEMY that we have noticed is the sacrifice in performance for seen languages. This result is related to our findings in unseen languages, where LINGUALCHEMY flattens out the performance across all languages, thus being extremely beneficial to poorly performing ones. Additionally, we note that the performance of seen languages is flattened to a similar level, which, most of the time, is worse than that of stan-

Performance	mBERT		XLM-R	
	Full FT	Ours	Full FT	Ours
Unseen Lgs.	45.48%	**64.68%**	78.97%	**80.43%**
Seen Lgs.	**84.52%**	67.45%	**86.45%**	81.05%
Average	64.25%	**66.01%**	**82.56%**	80.62%

Table 4: Average performance of LINGUALCHEMY on unseen vs seen languages

dard in-language fine-tuning. The compiled results can be seen in Table 4

Based on this finding, LINGUALCHEMY is suitable for enhancing the performance of extremely low-resource languages where standard cross-lingual zero-shot fine-tuning does not improve performance. In cases where training data is available, normal fine-tuning is a better choice. We are exploring why LINGUALCHEMY does not help with seen languages and how to boost the performance of seen languages as future work. Nevertheless, our method is still beneficial in under-resourced settings where multilingual models perform poorly.

6 Conclusion

We introduced LINGUALCHEMY, a novel approach that demonstrates strong performance across 27 unseen languages in a 60-class intent classification task. Our method hinges on the integration of linguistic knowledge through the URIEL vectors, enhancing the language model's ability to generalize across a diverse set of languages. We also proposed ALCHEMYSCALE and ALCHEMYTUNE, which employs a hyperparameter search for the URIEL scaling factor. This is achieved by two key strategies: (1) weight-averaging classification and URIEL loss, and (2) learning to balance the scale between classification and URIEL loss, thus ensuring a more adaptable and robust model performance.

Limitations

LINGUALCHEMY enhances performance across many unseen languages in intent classification, yet it faces limitations. Performance on seen languages is less than ideal, indicating room for improvement through methods like weight freezing. Also, better generalization appears to reduce accuracy in seen languages, pointing to a need for balanced approaches. Currently, the research is limited to intent classification, and expanding to other NLP tasks could reveal more about its versatility. Moreover, the choice of URIEL features—syntax, geography, language family—is theoretically sound, as discussed in Chapter 3, but empirical tests with different features might refine the model further. Overcoming these limitations could greatly improve the generalizability and effectiveness of multilingual NLP models.

References

Muhammad Farid Adilazuarda, Samuel Cahyawijaya, and Ayu Purwarianti. 2023. The obscure limitation of modular multilingual language models.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to african languages via multilingual adaptive fine-tuning.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Emily M. Bender. 2011. Linguistic issues in language technology on achieving and evaluating language-independence in nlp. *Linguistic Issues in Language Technology*, 6.

Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Leylia Khodra, Ayu Purwarianti, and Pascale Fung. 2021. Indonlg: Benchmark and resources for evaluating indonesian natural language generation.

Chris Collins. 2010. Syntactic structures of the world's languages (sswl). Colloquium presented at Yale University.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan,

John Ortega, Ricardo Ramos, Annette Rios, Ivan Meza-Ruiz, Gustavo A. Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Ngoc Thang Vu, and Katharina Kann. 2022. Americasnli: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. 2021. Compressing Large-Scale Transformer-Based Models: A Case Study on BERT. *Transactions of the Association for Computational Linguistics*, 9:1061–1080.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling.

Daniel Jurafsky and James H. Martin. 2019. Speech and language processing.

Melvyn Lewis. 2009. *Ethnologue: Languages of the World*, volume 9. SIL International.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14. Association for Computational Linguistics.

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.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

Saif M. Mohammad. 2019. The state of nlp literature: A diachronic analysis of the acl anthology.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021a. Adapterfusion: Non-destructive task composition for transfer learning.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021b. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling Language Variation and Universals: A Survey on Typological Linguistics for Natural Language Processing. *Computational Linguistics*, 45(3):559–601.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Vipul Rathore, Rajdeep Dhingra, Parag Singla, and Mausam. 2023. ZGUL: Zero-shot generalization to unseen languages using multi-source ensembling of language adapters. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6969–6987, Singapore. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang,

Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Jiacheng Xu, Siddhartha Jonnalagadda, and Greg Durrett. 2022. Massive-scale decoding for text generation using lattices. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4659–4676, Seattle, United States. Association for Computational Linguistics.

Lei Zhang, Jingru Fu, Shanshan Wang, David Zhang, Zhaoyang Dong, and C. L. Philip Chen. 2020. Guide subspace learning for unsupervised domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3374–3388.

A Languages in Dataset

The MASSIVE *Dataset*, also known as the *Multilingual Amazon SLU Resource Package* (SLUPR), offers a comprehensive collection of approximately one million annotated utterances for various natural language understanding tasks such as slot-filling, intent detection, and Virtual Assistant performance evaluation. It is an extensive dataset that includes 51 languages, 60 intents, 55 slot types, and spans 18 different domains. The dataset is further enriched with a substantial amount of English seed data, comprising 587k training utterances, 104k development utterances, and 152k test utterances.

Code	Name	Script	Genus	Code	Name	Script	Genus
ar-SA	Arabic	Arab	Semitic	is-IS	Icelandic	Latn	Germanic
bn-BD	Bengali	Beng	Indic	ka-GE	Georgian	Geor	Kartvelian
el-GR	Greek	Grek	Greek	km-KH	Khmer	Khmr	Khmer
en-US	English	Latn	Germanic	lv-LV	Latvian	Latn	Baltic
es-ES	Spanish	Latn	Romance	ml-IN	Malayalam	Mlym	Southern Dravidian
fa-IR	Persian	Arab	Iranian	nb-NO	Norwegian	Latn	Germanic
fr-FR	French	Latn	Romance	ro-RO	Romanian	Latn	Romance
he-IL	Hebrew	Hebr	Semitic	sl-SI	Slovenian	Latn	Slavic
hu-HU	Hungarian	Latn	Ugric	ur-PK	Urdu	Arab	Indic
hy-AM	Armenian	Armn	Armenian	zh-CN	Mandarin	Hans	Chinese
id-ID	Indonesian	Latn	Malayo-Sumbawan	zh-TW	Mandarin	Hant	Chinese

Table 5: Statistics and description of the dataset used (Xu et al., 2022). The dataset used is a subset of the MASSIVE dataset, selecting 25 different seen languages.

Code	Name	Script	Genus	Code	Name	Script	Genus
af-ZA	Afrikaans	Latn	Germanic	my-MM	Burmese	Mymr	Burmese-Lolo
am-ET	Amharic	Ethi	Semitic	nl-NL	Dutch	Latn	Germanic
az-AZ	Azerbaijani	Latn	Turkic	pl-PL	Polish	Latn	Slavic
cy-GB	Welsh	Latn	Celtic	pt-PT	Portuguese	Latn	Romance
da-DK	Danish	Latn	Germanic	ru-RU	Russian	Cyrl	Slavic
de-DE	German	Latn	Germanic	sq-AL	Albanian	Latn	Albanian
fi-FI	Finnish	Latn	Finnic	sv-SE	Swedish	Latn	Germanic
hi-IN	Hindi	Deva	Indic	sw-KE	Swahili	Latn	Bantoid
ja-JP	Japanese	Jpan	Japanese	ta-IN	Tamil	Taml	Southern Dravidian
kn-IN	Kannada	Knda	Southern Dravidian	te-IN	Telugu	Telu	South-Central Dravidian
ko-KR	Korean	Kore	Korean	th-TH	Thai	Thai	Kam-Tai
mn-MN	Mongolian	Cyrl	Mongolic	vi-VN	Vietnamese	Latn	Viet-Muong
ms-MY	Malay	Latn	Malayo-Sumbawan				

Table 6: Statistics and description of the dataset used (Xu et al., 2022). The dataset used is a subset of the MASSIVE dataset, selecting 27 different unseen languages.

B Algorithm

Formally, we define the language representation alignment in Algorithm 1, where F_U represents the features extracted from URIEL, S is the set of sentence representations, H_x and N_x are the hidden states and number of attention-masked tokens for a sentence x, respectively. The matrix W is used for the linear projection, and A holds the final aligned representations. Algorithm 1 outlines the process for aligning language representations we use in Figure 2. It leverages the URIEL database for linguistic features, processes sentences through a language model (Θ), and aligns these with mBERT representations (M). The algorithm iteratively updates transformation parameters (W and b) through a training loop to minimize the loss between the projected mBERT representations and the target sentence representations in set S, thus achieving aligned language representations (A).

Algorithm 1 Language Representation and Alignment Process

Require: Dataset D, URIEL database U, Language Model Θ, mBERT representations M
Ensure: Aligned Language Representations A

$F_U \leftarrow$ EXTRACTFEATURES(U)
$S \leftarrow \{\}$
for each sentence x in D **do**
  $H_x \leftarrow$ GETLASTHIDDENSTATES(x, Θ)
  $N_x \leftarrow$ COUNTATTENTIONMASKED(x)
  $R_x \leftarrow \frac{\text{SUM}(H_x)}{N_x}$
  $S \leftarrow S \cup \{R_x\}$
end for
$W, b \leftarrow$ INITIALIZEPARAMETERS()
for each training epoch **do**
  $P_U \leftarrow (W \times S) + b$
  $loss \leftarrow$ COMPUTELOSS(P_U, F_U)
  $W, b \leftarrow$ UPDATEPARAMETERSWITHCONSTRAINT($W, b, loss$)
end for
$A \leftarrow \{\}$
for each sentence representation s in S **do**
  $A_m \leftarrow (W \times s) + b$
  $A \leftarrow A \cup \{A_m\}$
end for

C Language Family Experiment

Tables 7 and 8 provide a comprehensive analysis of language family performance across different training groups. These tables compare the accuracy percentages of the Multilingual BERT and XLM-RoBERTa models, respectively. The results displayed in the tables elucidate the models' capabilities in generalizing from the training data to unseen languages. A clear trend that can be observed is the improvement in performance as the training groups progress from 1 to 4, which suggests that the models benefit from exposure to a wider variety of language families during training. The 'Average' row at the bottom of each table indicates the mean accuracy across all language families, providing an insight into the overall performance enhancement achieved by each model with incremental training diversity.

Language Family	**Train Group 1**	**Train Group 2**	**Train Group 3**	**Train Group 4**
Afro-Asiatic	52.82%	52.93%	61.26%	61.00%
Atlantic-Congo	65.71%	68.08%	70.62%	71.79%
Austroasiatic	64.77%	66.78%	69.72%	70.16%
Austronesian	66.88%	68.66%	72.06%	72.19%
Dravidian	64.74%	67.97%	70.93%	71.41%
Indo-European	67.50%	68.61%	72.53%	72.95%
Japonic	72.11%	71.98%	75.80%	75.67%
Kartvelian	68.91%	68.89%	72.46%	72.32%
Koreanic	64.80%	66.46%	70.04%	69.91%
Mongolic-Khitan	63.11%	66.44%	69.71%	69.59%
Sino-Tibetan	62.65%	66.29%	68.79%	70.33%
Tai-Kadai	63.52%	67.89%	70.23%	71.34%
Turkic	54.69%	56.91%	63.54%	64.05%
Uralic	71.49%	71.27%	75.33%	75.15%
Average	65.54%	67.07%	71.04%	71.43%

Table 7: Multilingual BERT Performance of Language Families Across Training Groups

Language Family	**Train Group 1**	**Train Group 2**	**Train Group 3**	**Train Group 4**
Afro-Asiatic	75.74%	76.23%	85.56%	85.39%
Atlantic-Congo	70.86%	72.38%	83.24%	82.73%
Austroasiatic	74.85%	76.04%	83.91%	83.59%
Austronesian	78.94%	79.83%	84.77%	84.69%
Dravidian	81.49%	82.20%	85.41%	85.43%
Indo-European	80.31%	81.21%	83.26%	83.47%
Japonic	80.21%	81.36%	82.67%	83.15%
Kartvelian	80.40%	81.53%	82.79%	83.27%
Koreanic	79.74%	80.91%	82.14%	82.61%
Mongolic-Khitan	79.54%	81.00%	82.20%	82.65%
Sino-Tibetan	79.25%	81.00%	82.14%	82.58%
Tai-Kadai	79.08%	80.81%	81.90%	82.35%
Turkic	79.20%	80.90%	81.96%	82.39%
Uralic	79.24%	80.91%	81.92%	82.47%
Average	79.45%	80.48%	83.44%	83.62%

Table 8: XLM-RoBERTa Performance of Language Families Across Training Groups