IT-UNIVERSITETET I KØBENHAVN

Department of Computer Science

Ph.D. Thesis

Max Müller-Eberstein

飯島マクシミリアン

Quantifying Linguistic Variation

Data-driven Navigation of Variety Space

This thesis has been submitted to the Ph.D. School of the IT University of Copenhagen on 22nd May, 2024.

The research for this doctoral thesis has received funding from the Independent Research Fund Denmark (DFF) Sapere Aude grant 9063-00077B and in parts by European Research Council (ERC) Consolidator grant DIALECT 101043235.

Committee

Advisor

prof. dr. Barbara Plank	Ludwig-Maximilians-Universität München IT-Universitetet i København

Co-Advisors

dr. Rob M. van der Goot	IT-Universitetet i København
prof. dr. Ivan Titov	University of Edinburgh Universiteit van Amsterdam

Members

prof. dr. Bonnie Webber	University of Edinburgh
prof. dr. Joakim Nivre	Uppsala University Research Institutes of Sweden
dr. Pinar Tözun	IT-Universitetet i København

Abstract

Language emerges naturally from human communication, and as such, linguistic variation across the many possible dimensions of expression is ubiquitous. Higher variation across specific dimensions leads to a decrease in mutual intelligibility, or, in the case of Natural Language Processing (NLP), to decreased model transferability. Linguistics delineates between dimensions such as typology, domain, register, etc., using qualitative definitions, however, these are difficult to apply quantitatively and to combine at scale. NLP on the other hand necessitates a quantization of language, and has thus enabled machines to learn data-driven, vectorized representations thereof, which measure language similarity remarkably well, but fall short of explaining exactly how two data points are related. By leveraging probing methods to segment the high-dimensional latent spaces of Language Models (LMs) into subspaces with linguistically interpretable similarity characteristics, we aim to bridge the divide between these two disciplines. Our results for cross-lingual syntax and cross-domain genre demonstrate that corresponding subspaces can be successfully recovered, and consequently used to predict which training data and models transfer well to unseen language varieties and domains. Combining dimensions from across this Variety Space, we further quantify task similarity in an interpretable way, and investigate how linguistic information emerges in LMs during their training. As NLP increasingly relies on general purpose information stored in LMs to solve myriads of downstream tasks, we argue that quantifying and understanding language and task variation is critical to ensure model robustness and trustworthiness. Towards this goal, our quantitative measures of linguistic variation provide a generally applicable framework grounded in traditional linguistics.

Resumé

Sprog er en naturlig del af menneskelig kommunikation, og de mange mulige dimensioner af udtryk, som kan variere på tværs, medfører sproglig variation. Større variationer i specifikke dimensioner kan føre til, at sproget er sværere at forstå eller, i forhold til sprogteknologi (en: *Natural Language Processing*; NLP), til lavere generaliserbarhed af modellerne. Lingvistik skelner mellem dimensioner som typologi, domæne, register osv. ved hjælp af kvalitative definitioner, men disse er svære at anvende kvantitativt og også svære at kombinere. NLP kvantificerer derimod sproget og har således gjort det muligt for maskiner at lære datadrevne, vektoriserede repræsentationer af sprog. Disse omfatter mange forskellige begreber af sproglig lighed, men forklarer ikke på en menneskelig forståelig måde, hvordan to datapunkter hænger sammen. Ved hjælp af probingmetoder til at opdele de højdimensionelle *latent spaces* i sprogmodeller (en: *Language Models*; LMs) i *subspaces*, der deler lingvistiske karakteristika, sigter vi mod at bygge bro mellem NLP og lingvistik. Vores resultater på tværs af sproglige typologier og genrer viser, at vi med success kan gendanne tilsvarende *subspaces*, der kan bruges til at forudsige, hvilke træningsdata og modeller, som generaliserer godt til nye sprog og domæner. På samme måde kan vi kvantificere ligheder mellem NLP-opgaver ved at kombinere variationsdimensioner og undersøge hvordan sproglig information bliver lært af LM'er under træningen. NLP er i stigende grad afhængig af generelle sproglige egenskaber gemt i LM'er for at løse opgaver, der kræver kombinationer af mange forskellige sproglige færdigheder. Derfor hævder vi, at kvantificering og forståelse af sprog- og opgavevariation er afgørende for at sikre modellens robusthed og troværdighed. Vores kvantitative mål for sproglig variation baserer sig dermed på kvalitative definitioner fra lingvistik. Derfor giver de en fleksibel ramme til at analysere relevante informationer i LM'er på en måde, der kan fortolkes af mennesker.

Acknowledgements

[Acknowledgments to be added after defense]

Declaration of Work

I, Max Müller-Eberstein, declare that this thesis—submitted in partial fulfillment of the requirements for the conferral of a Ph.D., from the IT University of Copenhagen—is solely my own work unless otherwise referenced or attributed. Neither the thesis nor its content have been submitted (or published) for qualifications at another academic institution.

– Max Müller-Eberstein

Table of Contents

Part I

Introduction

1 Motivation

Language emerges naturally from human communication, such that variation is inevitable to the point that—we would argue—everyone has their own unique language variety. Of course, completely disjunct languages would not serve their purpose, and thus, their variability is limited on some dimensions more than others: linguistic typology (phonology, lexicology, syntax), for instance, is relatively stable within a language community, i.e., people "speak the same language". On the other hand, even within one typology, the domain (e.g., genre), as well as many other factors, such as pragmatics (e.g., social context), may change language to such a degree that it becomes difficult to transfer between settings. These dimensions do not exist in isolation, but are closely tied to one another. Language thus exists in the high-dimensional manifold of *Variety Space* (Plank, 2016).

Understanding which variety dimensions constitute this space and how to navigate them is crucial, as larger distances correspond to decreases in mutual intelligibility. In Natural Language Processing (NLP), this manifests as decreased model transferability, i.e., how well a model trained on one language variety fits another, without additional tuning. The rigidity of computational models compared to humans makes it even more important to understand shifts in Variety Space a priori, as knowledge regarding the types of variation a model is robust against is essential for establishing its trustworthiness in downstream scenarios (Litschko et al., 2023). Acquiring this knowledge necessitates quantitative measures of linguistic variation, and as such, this work's objective is to provide a general framework for comparing textual language data and models thereof across Variety Space, and to evaluate this framework on a focused set of variety dimensions.

1.1 Navigating Variety Space

Humans intuitively navigate Variety Space on a daily basis. However, it is notoriously difficult to define (Section 2.1): Linguistics provides *qualitative* definitions of individual variety dimensions, which are helpful for delimiting properties that affect human communication in different ways. For example, order and inflections are closely tied to a word's function in a sentence, and show little variation within a language. Consolidating these patterns into a set of morphosyntactic rules then allows us to compare languages qualitatively. Although such definitions can be used to inform quantitative comparisons of language varieties across some dimensions (Baayen, 2008), they fail to scale across the entirety of Variety Space. This issue is exacerbated, the less formalizable a property becomes (e.g., typology versus pragmatics), and the more data points there are to compare. This makes it near impossible to accurately describe interactions between two or more variety dimensions of large-scale data on a continuous spectrum.

Natural Language Processing (NLP) not only necessitates the quantization of language, but has further excelled and thrived upon doing so at scale. Mirroring the hierarchy of increasing variability, early NLP focused on more consistent, typological properties, such as syntax in the form of expert-curated grammars, and later shifted towards data-driven methods for better coverage of less formalizable phenomena (Manning and Schütze, 2003). Representation Learning has been fundamental in this paradigm shift. Based on the assumption that the majority of relevant information is encoded by the probability of data co-occurring (Harris, 1954; Firth, 1957), statistical Language Models (LMs) learn quantized representations of language (i.e., vector spaces), which contain useful information for a myriad of downstream tasks (Rosenfeld, 2000). Usefulness, in this case, refers to the fact that data, which are similar with respect to real-world properties (e.g., syntax, semantics, register), have vectorized representations that are embedded close to each other in the LM's overall vector space. These embedding spaces form the foundation of contemporary NLP, and have proven to encode useful information for all of its sub-fields, without requiring expert feature engineering (Mikolov et al., 2013).

Downstream, task-specific models, make use of these embeddings by identifying latent information, which is highly correlated with the task's input-output pairs. This applies both to classification, where embedding features are used to identify decision boundaries, as well as to generation, where the output space is conditioned on the embeddings of the preceding input. As LMs are statistical in nature, this general approach is based on the assumption that data are independently and identically distributed, including those that the model will be applied to in the future. One of the largest ongoing issues in NLP, and statistical Machine Learning (ML) in general, has therefore been to ensure that models behave similarly on in-distribution (IND) and out-of-distribution data (OOD).

While model transferability is an important practical concern, the phenomenon itself already provides a valuable theoretical tool for quantifying linguistic variation: i.e., higher variation → lower transferability. Linking this idea back to Representation Learning, higher representational similarity between data implies lower variation, which should correlate with higher downstream transfer performance. LMs thus learn their own Variety Space, within which we are able to measure variation between language data quantitatively. These measures, however, lack a crucial feature compared to qualitative definitions of Variety Space: In contrast to comparisons of expert-curated features and rules, variation as measured by LMs is purely data-driven, and thus not immediately interpretable.

In order to bridge this divide, we build on model interpretability methods in the form of *probing* (Section 3.3). Contrary to the common use-case of probing, which aims to identify how much information relating to a certain linguistic property is encoded in an LM, we propose a new perspective, leveraging probes themselves as *variety subspaces*, within which representational similarity corresponds to linguistic similarity across specific variety dimensions.

In this work, we investigate the theoretical and practical implications of this hypothesis: In Part I, we build on existing qualitative definitions of variation (Chapter 2), and propose a general framework for identifying interpretable subspaces in LM representations (Chapter 3). In practice, we leverage these measures to compare language varieties, and to predict which model/data combinations have beneficial properties for transferability (Parts II to IV).

1.2 Focus Areas

Due to the high dimensionality of Variety Space, it would be impossible for a single study to cover each dimension, as well as their interactions, in detail. We therefore focus on three specific dimensions of variation with distinctive differences and overlaps, in order to demonstrate our general framework. In the following, we provide working definitions of each focus area, before surveying existing definitions in Chapter 2.

Typological Variation (Part II): This dimension includes the variety sub-dimensions most important to mutual intelligibility, such as phonology, lexicology and syntax (Background Section 2.1.1; Rijkhoff, 2007). Similarity in this dimension implies *cross-lingual transferability*. We examine typology from the perspective of *syntax*, as it is a relatively consistent feature within a language variety, thus forming an essential foundation for more complex, downstream language understanding (Comrie, 1981; Hawkins, 1983). Additionally, syntax benefits from established datasets (Nivre et al., 2020) and standards for formalization (de Marneffe et al., 2014, 2021).

Domain Variation (Part III): Variation's effects on downstream performance are frequently evaluated in the context of this dimension. Despite *cross-domain transferability* being studied extensively, the notion of what constitutes a domain has actually not been rigorously formalized (Background Section 2.1.2). Within the practical context of NLP, we broadly consider it as any non-typological property, which necessitates significant modeling changes. Out of these properties, we chose to focus on *genre*. After typology, it is one of the top-level dimensions considered during dataset creation (e.g., Aston and Burnard, 1998; Nivre et al., 2020; Sharoff, 2021; Kuzman et al., 2022), and across which models trained on the same typological language variety are expected to transfer reasonably well.

Task Variation (Part IV): Tasks in NLP rely on different mixtures of linguistic information and skills to map natural language inputs to a task-specific output space (Schlangen, 2021; Weber et al., 2021). As such, they are also susceptible to variation along the dimensions described above. With the advent of large LMs prompted with complex user instructions, NLP models are increasingly expected to transfer well to a wide variety of open-domain tasks in a zero-shot manner. To ensure human-model trust in these scenarios, understanding how well task-relevant variety dimensions are represented in a model, as well as how an unseen task may relate to other tasks, becomes of critical importance (Litschko et al., 2023). By taking a broader view, and characterizing tasks as variations over output space, we place tasks on a continuous spectrum, which allows for quantifying task similarity in a linguistically grounded way (Background Section 2.2.2).

1.3 Research Questions

To enable a deeper understanding of Variety Space, we contribute findings towards research questions centered around our three aforementioned focus areas of typology, domain, and task variation. For each of them, we first survey human-centric, *qualitative* definitions (Chapter 2), before linking these back to *quantitative*, data-driven measures (Chapter 3). This dual approach is illustrated in Figure 1.1, and ensures that the interpretability methods proposed in this work (Parts II to IV) are grounded linguistically.

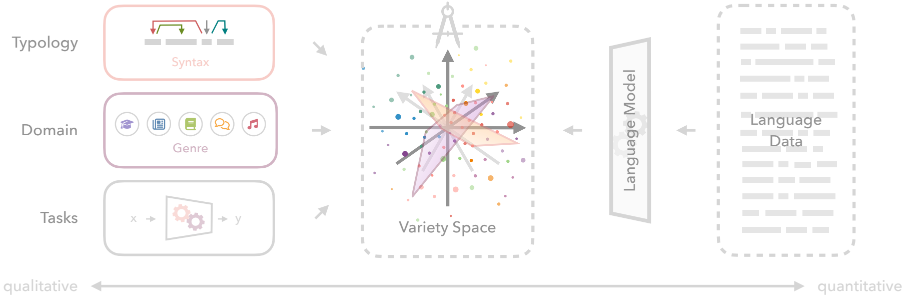


Figure 1.1: **Overview of Research Objectives.** We propose a framework for measuring linguistic variation in an interpretable way by bridging qualitative and quantitative notions of typology, domains, and tasks.

1.3.1 Typological Variation

RQ1 How is syntactic information from different typologies represented in data-driven latent spaces?

We further subdivide this larger question into:

RQ1.1 Which qualitative definitions exist for typological variation?

Interpretability requires human definitions of the property of interest. In Background Section 2.1.1, we therefore survey existing definitions of linguistic typology to extract a working definition for our subsequent studies. Within typology, we focus on syntax and provide an overview of how this property has been modeled in NLP.

RQ1.2 Does quantitative LM latent space contain sufficient typological information to extract fully directed and labeled dependency trees?

In Chapter 4, we build on the typological formalism of syntactic dependency trees, and investigate whether probing methods can robustly extract these structures from LM latent spaces. In contrast to prior work, our proposed dependency probe (DEPPROBE) is able to learn syntactic subspaces within the LM, that contain fully-directed and labeled dependency trees.

RQ1.3 How well does syntactic probing predict the cross-lingual transferability of a full parser?

Leveraging the syntactic subspaces extracted by DEPPROBE, we analyze their representational overlaps across languages (Chapter 4). The amount of overlap allows us to predict the robustness of parsers trained on one language when applied to another—effectively ranking which language data are best suited for cross-lingual transfer.

RQ1.4 How well does syntactic probing predict which LM is best suited for dependency parsing in a specific language?

Chapter 5 follows a related, but orthogonal, setting, in which we investigate how well the previous probing-to-rank approach predicts the suitability of LMs as initializations for training language-specific parsers. In contrast to selecting a language to transfer from, model selection is much less well defined and currently relies on practitioner intuition. We show how measuring syntactic information in pre-trained LMs using DepProbe allows us to rank their suitability in a more evidence-rooted way.

1.3.2 Domain Variation

RQ2 How does domain information manifest in data-driven latent spaces across languages?

We further subdivide this larger question into:

RQ2.1 Which qualitative definitions exist for domain variation?

In Background Section 2.1.2, we survey what constitutes a domain in linguistics and NLP. Compared to typology, this variety dimension is even less strictly formalized. Nonetheless, we are able to extract a working definition based on the property of genre, which we apply in our subsequent studies.

RQ2.2 To what extent can humans qualitatively identify domain from text alone, and how well does this align with machines?

Chapter 6 focuses on domain variation via the properties of genre and topic. Our study is the first of its kind to qualitatively investigate how these domain properties are perceived intuitively by human annotators. Both for humans and machines, our analysis provides a first indication for the necessity of a more continuous spectrum when measuring domain variation.

RQ2.3 Can cross-lingual genre information be amplified in LM latent spaces using weak supervision?

Combining the dimensions of domain and typology in Chapters 7 and 8, we survey how genre manifests in the highly cross-lingual context of Universal Dependencies. On the qualitative side, genres are only labeled at the treebank level, making more granular analyses difficult, while on the quantitative side, the raw representational similarity of individual sentences is insufficient to explicitly isolate variation along the genre dimension. As such, we propose weakly-supervised learning to amplify latent genre information within the overall LM embedding space using treebank-level genre metadata. By leveraging a shared, multilingual latent space, our proposed methods are able to bootstrap how genres are distributed across the dataset via cross-lingual representational similarity.

RQ2.4 Can amplified genre guide our selection of cross-lingual training data from a significantly larger, more diverse pool?

In Chapter 8, we leverage the previously amplified cross-lingual genre information to select training data for an unseen target language with a known genre. Assuming we do not have access to in-language data, we gather proxy training data in different languages, while controlling for genre. This improves the robustness of zero-shot, cross-lingual transfer performance for dependency parsing.

1.3.3 Task Variation

RQ3 Can data-driven measures of linguistic variation be leveraged to quantify task similarity in an interpretable way?

We further subdivide this larger question into:

RQ3.1 What constitutes a task in NLP?

While RQs 1 and 2 correspond to input variation, we argue that tasks in NLP can be characterized as variations over output space (Background Section 2.2.2). Consequently, a model's transferability to new tasks is tied to how well it represents task-relevant dimensions of Variety Space, with the overlap of relevant variety dimensions further quantifying task similarity.

RQ3.2 When does task-specific linguistic information emerge during LM training?

Towards RQs 1 and 2, Parts II and III investigate the presence of information related to typology and domain in LM latent spaces. In Chapter 9, we analyze how and when this information emerges from self-supervised LM training objectives by comparing respective information-theoretic probes from across LM training time. Our study reveals critical learning phases, as well as previously unseen interactions between different task-relevant variety dimensions.

RQ3.3 Which linguistic information is shared across tasks, and how do their subspaces interact across LM training time?

Treating the resulting probes from Chapter 9 as linguistically-motivated representational subspaces allows us to measure shifts in their overlaps across LM training. We leverage this formulation as a method for quantifying task variation on a continuous, yet interpretable, spectrum. This helps us identify which linguistic information is shared across tasks, and during which training phase.

RQ3.4 How can the same task be characterized consistently across different languages?

While the probed subspaces from the previous approaches are difficult to compare across models and languages due to their specificity with respect to the base LM representations, Chapter 10 seeks signatures of linguistic tasks, which remain more generally consistent. Towards this purpose, we focus on how task-relevant information is encoded across a sentence, i.e., the consistency of that information over time. Intuitively, for instance, syntactic information (e.g., parts-of-speech) should be correlated more at shorter distances, while the sentiment of a sentence should be encoded consistently even across longer distances. We propose Spectral Probing as a method to characterize this rate of change of task-relevant information across a sequence as frequency profiles in the signal processing sense, and apply it to seven common NLP tasks across six typologically diverse languages.

1.4 List of Publications

The chapters in this thesis aim to provide answers towards these research questions, and are based on the following works, published over the course of the Ph.D. project:

Part II Typological Variation

- **Chapter 4**: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022a. Probing for labeled dependency trees. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7711–7726, Dublin, Ireland. Association for Computational Linguistics.
- **Chapter 5**: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022b. Sort by structure: Language model ranking as dependency probing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1307, Seattle, United States. Association for Computational Linguistics.

Part III Domain Variation

- **Chapter 6**: Maria Barrett, Max Müller-Eberstein, Elisa Bassignana, Amalie Brogaard Pauli, Mike Zhang, and Rob van der Goot. 2024. Can humans identify domains? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, Torino, Italy. European Language Resources Association. (All authors contributed equally.)
- **Chapter 7**: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021b. How universal is genre in Universal Dependencies? In *Proceedings of the 20th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2021)*, pages 69–85, Sofia, Bulgaria. Association for Computational Linguistics.
- **Chapter 8**: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021a. Genre as weak supervision for cross-lingual dependency parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4786–4802, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Part IV Task Variation

- **Chapter 9**: Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208, Singapore. Association for Computational Linguistics.
- **Chapter 10**: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022c. Spectral probing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7730–7741, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Additional Publications Collaborations during the Ph.D. have lead to additional publications, tangential to this work:

- Rob van der Goot, Max Müller-Eberstein, and Barbara Plank. 2022. Frustratingly easy performance improvements for low-resource setups: A tale on BERT and segment embeddings. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1418–1427, Marseille, France. European Language Resources Association.
- Dennis Ulmer, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank. 2022. Experimental standards for deep learning in natural language processing research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2673–2692, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Tanja Samardžić, Ximena Gutierrez-Vasques, Rob van der Goot, Max Müller-Eberstein, Olga Pelloni, and Barbara Plank. 2022. On language spaces, scales and cross-lingual transfer of UD parsers. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 266–281, Abu Dhabi,

United Arab Emirates (Hybrid). Association for Computational Linguistics.

- Elisa Bassignana, Max Müller-Eberstein, Mike Zhang, and Barbara Plank. 2022. Evidence > intuition: Transferability estimation for encoder selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4218–4227, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- Kia Kirstein Hansen, Maria Barrett, Max Müller-Eberstein, Cathrine Damgaard, Trine Eriksen, and Rob van der Goot. 2023. DanTok: Domain beats language for Danish social media POS tagging. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 271–279, Tórshavn, Faroe Islands. University of Tartu Library.

- Robert Litschko, Max Müller-Eberstein, Rob van der Goot, Leon Weber-Genzel, and Barbara Plank. 2023. Establishing trustworthiness: Rethinking tasks and model evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Singapore. Association for Computational Linguistics.

- Rob van der Goot, Zoey Liu, and Max Müller-Eberstein. 2024. Enough is enough! a case study on the effect of data size for evaluation using universal dependencies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, Torino, Italy. European Language Resources Association.

2 Defining Linguistic Variation

Linguistic variation is notoriously difficult to define: While definitions for some variety dimensions, such as typology, enjoy a broader (yet not uncontested) consensus in linguistics and NLP, there are essentially infinitely many axes, which are impossible to capture manually (Biber, 1988). Working definitions of each dimension are nonetheless essential to establish common ground for interpreting their quantitative manifestations. By viewing this problem through the joint lens of traditional linguistics and data-driven NLP, this question further offers opportunities to compare how similar human and machine-driven notions of variation are. In the following, we therefore first survey qualitative definitions of our focus areas of typology (Section 2.1.1) and domain (Section 2.1.2), before exploring NLP-specific notions of input variation (Section 2.2.1) and task variation over the output space (Section 2.2.2), as well as their implications for model robustness and trustworthiness (Section 2.2.3).

2.1 Qualitative Definitions of Variation

A decrease in mutual intelligibility signals a shift in Variety Space. Linguistics literature aims to qualitatively define this differentiation between language varieties by categorizing the features which differ, or are shared, between them. Building on our working definitions from Section 1.2, we first survey the categories of typology and domain, as well as how prior work in NLP aims to operationalize these dimensions.

2.1.1 Typology

As variation across the typological dimension has the largest effects on cross-lingual transferability, it is used to draw lines between language families, dialects etc., and is considered relatively well-defined in NLP, e.g., via language codes set by the International Organization for Standardization (ISO).[1] Nonetheless, its exact definition is far from uncontested, as, for instance, many languages do not have standardized ISO codes, or are grouped together despite larger differences (Gillis-Webber and Tittel, 2020). Such typological differences manifest across multiple layers, including phonological, lexical, and syntactic differences, as well as influences via geographic proximity (see Rijkhoff, 2007 for a brief overview).

A seminal resource for NLP practitioners exploring these typological properties is the URIEL knowledge base of typological features and its associated vectorized form, lang2vec (Littell et al., 2017). It consolidates manual annotations of syntactic features (Dryer, 1992; Lewis et al., 2015), phonological features (Moran et al., 2014), broader phylogenic properties (Hammarström, 2015), together with geographic information compiled from across these sources. These qualitative annotations are extensive, covering approximately 8,000 ISO 639-3 codes, however they are neither complete, nor equally information dense across languages—especially for under-resourced varieties. As such, lang2vec fills in missing entries by interpolating between the 10 most similar languages, building on the correlation between typological features (Daumé III and Campbell, 2007; Takamura et al., 2016).

From across the aforementioned typological dimensions, morphology and syntax have enjoyed the widest use for grouping language varieties into families (Greenberg, 1966; Comrie, 1981; Hawkins, 1983; Dryer, 1992). For example, despite differences in phonology, lexicology and orthography within languages of the same family, morphosyntax remains relatively consistent (e.g., subject-verb-object versus subject-object-verb order). Building on this consistency, early NLP focused on automatically deriving meaning from text by first parsing syntactic structures, based on the assumption that certain syntactic rules hold within a language. Consequently, these parsers were built on context-free grammars (Chomsky, 1956), which were manually curated by linguistic experts. The importance of syntax to establish basic meaning also applied cross-lingually, with some machine translation approaches relying on synchronous grammars, which map syntactic

[1] ISO-639 Language Codes (https://www.iso.org/iso-639-language-code) and its subsets 1–5, covering up to 8,440 variants (accessed 5th March, 2024).

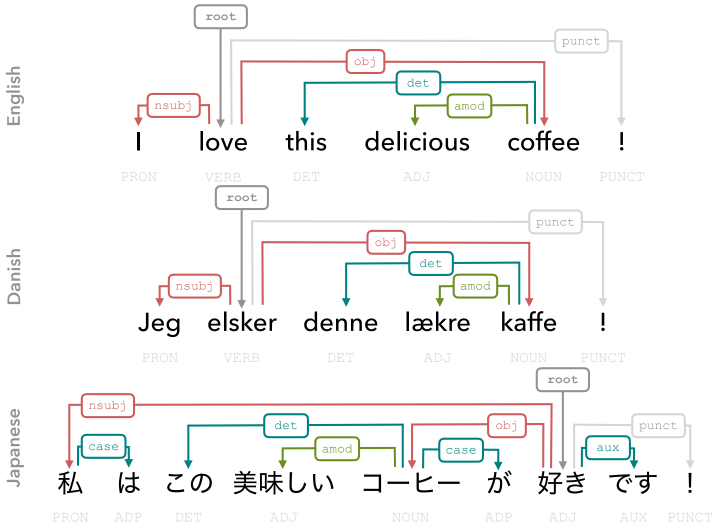

English
root
punct
obj
det
amod
nsubj
I love this delicious coffee !
PRON VERB DET ADJ NOUN PUNCT
Danish
root
punct
obj
det
amod
nsubj
Jeg elsker denne lække kaffe !
PRON VERB DET ADJ NOUN PUNCT
Japanese
root
nsubj
det
obj
case
amod
case
punct
aux
私 は この 美味しい コーヒー が 好き です ！
PRON ADP DET ADJ NOUN ADP ADJ AUX PUNCT

Figure 2.1: **Universal Dependencies Annotations** (tokenization, parts-of-speech, typed dependency relations) for an example sentence across languages. Dependency edges are colored according to the official taxonomy (de Marneffe et al., 2014, 2021).

structures from one language to another (Chiang, 2007). Despite the relative rigidity of syntactic rules compared to other variety dimensions, these grammars never captured the full scope of even a well-studied language such as English, and as such, the role of statistics derived from large amounts of raw data continued increasing over time (Manning and Schütze, 2003). Indeed, contemporary approaches to syntactic parsing rely almost purely on the latent information learned by self-supervised Language Models, and merely train a parsing head on top of the LM in a final, task-specific fine-tuning step (e.g., Dozat and Manning, 2017).

Regardless of how syntax is inferred, NLP systems require some explicit or implicit model of it to perform downstream tasks, and to generalize to new cases beyond their training data. This applies both to in-language generalization, as well as cross-lingual transfer. Evaluating syntactic processing across languages is difficult, however, there have been formalisms proposed to examine this phenomenon, at the forefront of which stands the Universal Dependencies project (UD;

Nivre et al., 2020). It is based on a morphosyntactic annotation scheme, which aims for universality across typologies (de Marneffe et al., 2014, 2021). Figure 2.1 demonstrates the basic annotation scheme for a sentence with an equivalent meaning across English, Danish and Japanese. For any included sentence, UD is guaranteed to include annotation layers for tokenization, parts-of-speech, and syntactic dependencies, plus additional information supplied by the independent contributors. The dependency relations surface context-sensitive information in the form of a sentence's overall dependency tree, where each word-level unit is connected to exactly one parent node in a directed acyclic graph terminating at the sentence's root. Each connection is labeled by its syntactic function, such that, e.g., the nominal subject of a sentence is connected via a `nsubj` relation to the root predicate. The provided example shows how this scheme can be consistently applied to a typologically varied set of languages, and highlights the syntactic similarities of English and Danish (i.e., identical trees), versus their differences to Japanese (e.g., different word order, lack of a verb), despite all sentences' shared semantic content.

In order to ensure broad applicability and low-friction implementation, the annotation scheme is neither the most granular (e.g., universal part-of-speech tags by Petrov et al., 2012, instead of language-specific sets), nor comprehensive (e.g., tokenization lacks consensus in languages such as Japanese; Omura et al., 2023). However, it has yielded one of the most diverse cross-lingual annotation efforts to date, and continues to grow.[2] This makes UD an invaluable resource for our goal of examining typological variation, specifically by analyzing the dimension of cross-lingual syntactic variation.

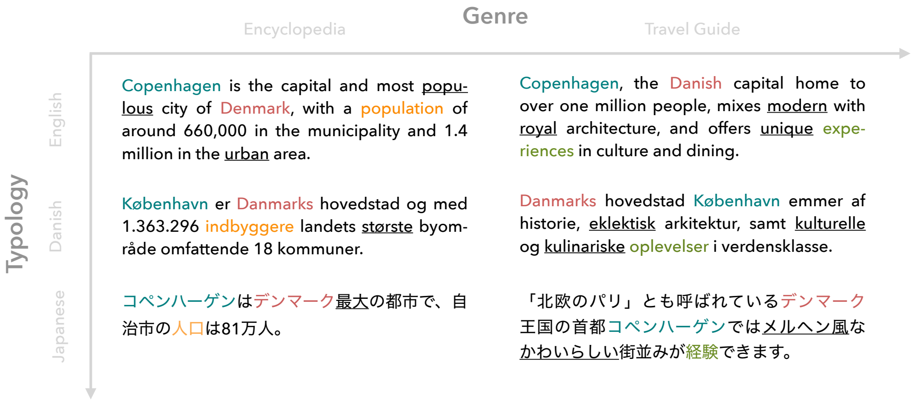
2.1.2 Domain

In contrast to typology, domain has been widely studied, but hardly formalized. Out-of-domain generalization is studied in all sub-disciplines of Machine Learning, and in NLP, there are certain non-typological, linguistic properties, which are commonly subsumed under this term. Initially referred to as *sublanguages* (Kittredge, 1982), these proper-

[2] 283 contributions in 161 languages as of version 2.14 in May, 2024.

ties manifest as shifts across various linguistic dimensions (Grishman and Kittredge, 1986). Biber (1995) further refines this definition into *registers*, which correspond to “any [language] variety associated with particular situational contexts or purposes”. In practice, they may refer to style and genre, as well as to a text’s source/type, but as they are frequently used interchangeably (Lee, 2001), we consolidate them under the term *genre*. As a working definition, we build on the fact that, across genres, the communicative purpose shifts, while, within a genre, there are certain shared linguistic features, which make it suitable for that particular purpose (Karlgren and Cutting, 1994; Kessler et al., 1997; Lee and Myaeng, 2002; Webber, 2009). Because the same communicative purpose can be expressed in any language, genre is orthogonal to typology, making it an ideal variational dimension to study in combination with the latter. Indeed, genre is often used as the next level of categorization after language, when creating large-scale corpora, both monolingually (e.g., Aston and Burnard, 1998; Kuzman et al., 2022), as well as multilingually (e.g., Nivre et al., 2020; Sharoff, 2021). It further differs decidedly from another variational dimension relevant to domain-shift, namely content in the form of *topic*, in that it defines a text’s communicative purpose while the latter conveys semantic content (Petrenz and Webber, 2011; van der Wees et al., 2015). It also differs from social communicative context (Hovy, 2015; Flek, 2020; Nguyen et al., 2021) as genre’s defining features inherently stem from its purpose and not from the characteristics of an external broadcaster or receipient.

Although our working definition of genre remains broad, it nonetheless manifests itself in a testable hypothesis, in that abstract genre differences should be reflected by the suitability of its associated linguistic features (e.g., more/less complex syntax). Some of these features of genre have further been shown to remain consistent independently of typology, via the NLP task of cross-lingual genre classification (Sharoff, 2007; Petrenz and Webber, 2011, 2012). Specifically, while surface-level lexical features are indicative of topic, they are tied too strongly to the language itself to reflect genre in another language. Meanwhile, syntactic parts-of-speech remain relatively consistent for genres across languages, but are less indicative of topic. Our examples in Figure 2.2 highlight these aforementioned properties: The topic of



	Encyclopedia	Travel Guide
English	Copenhagen is the capital and most populous city of Denmark, with a population of around 660,000 in the municipality and 1.4 million in the urban area.	Copenhagen, the Danish capital home to over one million people, mixes modern with royal architecture, and offers unique experiences in culture and dining.
Danish	København er Danmarks hovedstad og med 1.363.296 indbyggere landets største byområde omfattende 18 kommuner.	Danmarks hovedstad København emmer af historie, eklektisk arkitektur, samt kulturelle og kulinariske oplevelser i verdensklasse.
Japanese	コペンハーゲンはデンマーク最大の都市で、自治市の人口は81万人。	「北欧のパリ」とも呼ばれているデンマーク王国の首都コペンハーゲンではメルヘン風なかわいらしい街並みが経験できます。

Figure 2.2: **Examples of Typology/Genre Variation**, with a consistent topic, i.e., the city of Copenhagen. Select concepts are color-matched, and adjectives are underlined.

"the city of Copenhagen" remains consistent throughout, but takes on different surface forms depending on the language (e.g., Copenhagen, København, コペンハーゲン), and co-occurs with different context depending on the genre (e.g., population numbers versus touristic experiences). Meanwhile, focusing on syntactic information, we observe an increase in descriptive adjectives in the travel guides compared to the more neutrally valent encyclopedia entries. This shift occurs consistently despite the relatively high intuitive similarity of these genres (i.e., both communicate factual information), as well as the different cultural contexts across typologies (i.e., travel guides in different languages highlight different aspects of the destination). This first indication of syntax being a stable cross-lingual feature for genre will guide our further investigations into quantifying this variational dimensions across more languages.

2.2 Variation from an NLP Perspective

Analogously to human mutual intelligibility decreasing with increased variational distance, NLP methods similarly struggle to transfer across disparate language varieties. While our work leverages this fact as a measure of variation itself, a broad range of work has tackled the

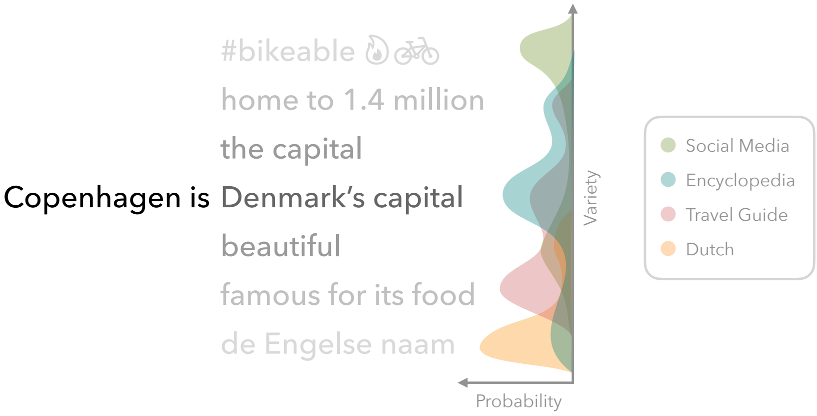


Figure 2.3: **Language Variation in NLP** corresponds to the divergence of word co-occurrence distributions across dimensions such as typology and genre (e.g., with respect to English encyclopedic text).

downstream implications of variation by focusing on increasing model robustness across distributional shifts in the input data (Section 2.2.1). Since all NLP tasks fundamentally rely on linguistic information, but vary with respect to how it is mapped to the output space, we further argue that measuring variation is essential to understanding cross-task transferability (Section 2.2.2). With the aforementioned trend of relying on statistical features of raw data, instead of manual feature engineering, it is becoming increasingly important to understand typological, domain and task variability on a continuous spectrum. In Section 2.2.3, we therefore outline how quantifying these variety dimensions links back to not only model robustness, but also their fundamental trustworthiness.

2.2.1 Input Variation as Distributional Divergence

NLP aims to model language regardless of its variety, and has therefore widely studied generalization beyond the input training distribution—both in terms of typology and domain. Contemporary LMs acquire almost all of their latent linguistic knowledge by learning probability distributions over word co-occurrences in large, unlabeled datasets. These models are subsequently applied to downstream target data based on the hypothesis that there are useful correlations between the training and target distributions. This approach of *transfer learning* has become the prevalent paradigm in the field: Initially, by pre-training LMs on as much unlabeled data as possible in a self-supervised manner before fine-tuning them on the target task using traditional supervised learning (Peters et al., 2018; Howard and Ruder, 2018), and more recently, by relying solely on the LM's self-supervised training, and using few-shot, in-context learning for immediate inference without supervised fine-tuning (Brown et al., 2020).

The transfer learning paradigm is predominant both for high-resource languages, such as English, where LMs trained on multi-billion token corpora are later prompted/fine-tuned on a specific task with fewer annotated data (e.g., Devlin et al., 2019; Brown et al., 2020), as well as for under-resourced languages, where a multilingual LM is trained on raw data from as many languages as possible, before being applied in a zero-shot manner to the target language variety (e.g., Conneau et al., 2020). In both cases, these LMs rely on statistical correlations from a diverse pool of training data to generalize to a broad set of unseen tasks, or even languages.

Due to their heavy reliance on transferable correlations, distributional divergence is the largest factor for LM performance. Figure 2.3 illustrates this issue, as the most likely continuation of "Copenhagen is [...]" strongly depends on the language variety being modeled. An LM, trained on encyclopedic text may therefore exhibit useful correlations to travel guides for some contexts, but not for all, and may further tend towards irrelevant or detrimental correlations for larger typological (e.g., Dutch) or genre shifts (e.g., social media).

The negative effects of distributional shift on model performance have been widely studied in NLP (Sekine, 1997; Gildea, 2001; Plank, 2011; Nagarajan et al., 2021; Ramesh Kashyap et al., 2021; White and Cotterell, 2021), in order to improve model robustness across Variety Space. While our work shares this overarching goal, we also leverage distributional divergence on its own as a basis for building measures of linguistic variation. Additionally, prior efforts have primarily targeted cross-lingual or cross-domain transferability separately. However, we argue that capturing a more holistic picture across multiple variety dimensions is crucial to ensure true model robustness. For example, a model trained on multilingual Wikipedia data, which is tested on an unseen language's Wikipedia would likely benefit from transferable features, such as the syntactic similarity to some training languages, as well as genre consistency in the form of cross-lingually consistent article editing guidelines. Applying the same model to social media data from a typological isolate however, will likely yield fewer useful correlations. In the aforementioned example, it is easy to identify potential reasons for variational shift, however most cases are more difficult to intuit. Understanding why certain transfers of information work well, and others do not, thus requires us to operate in the entirety of Variety Space, while also measuring shifts across individually separated variety dimensions (e.g., typology and genre).

2.2.2 Tasks as Variation over Output Space

While transfer learning has primarily targeted cross-lingual and cross-domain settings, practitioners have increasingly begun to focus on transferability at the level of tasks, as well. Based on the idea that similar tasks require similar knowledge and skills, models have either been trained consecutively on similar tasks (Wang et al., 2019a; Gururangan et al., 2020; Weller et al., 2022), or simultaneously via multi-task learning (Aribandi et al., 2022; Padmakumar et al., 2022; van der Goot, 2023). Larger LMs trained on more data have further foregone task-specific fine-tuning all together, and instead rely on few-shot, in-context learning—essentially enabling transferability to an unbounded set of tasks (Brown et al., 2020).

Mirroring language, variation across tasks lacks a precise definition, and is primarily guided by practitioner intuition (Bassignana et al., 2022). Some approaches have attempted quantifying task similarity via the representational similarity of their datasets (Poth et al., 2021), or by comparing the gradient updates from a general to a task-specific model (Achille et al., 2019; Vu et al., 2020; Ilharco et al., 2023). At a more abstract level however, all NLP tasks essentially rely on a mixture of the same basic linguistic information—from lower-level syntactic to higher-level semantic features—by mapping them to task-specific outputs. As illustrated in Figure 2.4, manual feature engineering makes this mapping more explicit (e.g., bag-of-words, PoS and sentiment

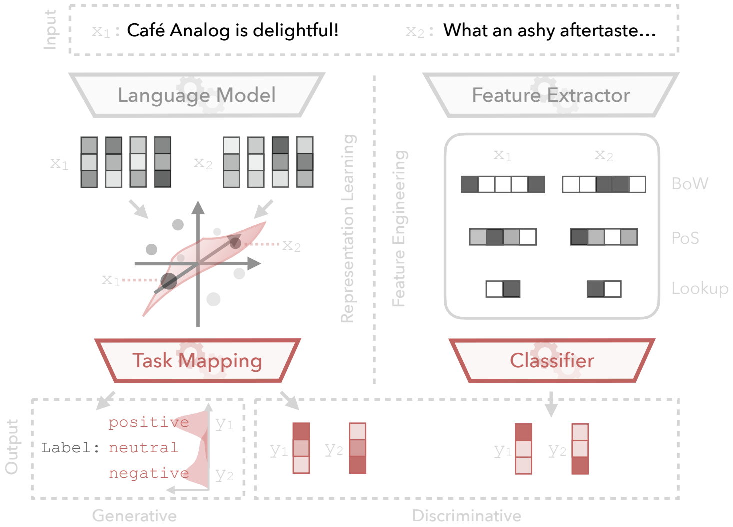


Figure 2.4: **Task Paradigms in NLP** for a sentiment analysis example: Linguistic information is extracted via manual feature engineering, or via LM representations. This information is then mapped to the task's outputs space in a discriminative or generative manner.

lookup dictionaries), while contemporary methods are purely dependent on latent linguistic information in LMs. In this example, certain features may indicate sentiment information (e.g., adjectives), while others help differentiate positive and negative valence (e.g., sentiment lookup/representational similarity). Regardless of the source, this linguistic information is mapped to the task-specific output space. This applies both to discriminative approaches, which map directly into label space, as well as to generative approaches, where the mapping induces a distributional shift in the output token probabilities.

As tasks fundamentally rely on mapping linguistic information from different variety dimensions to their output space, they implicitly form their own variation manifold, which is closely tied to Varitey Space—essentially forming subspaces of task-relevant linguistic information. We hypothesize that by not only measuring the degree of linguistic variation along individual dimensions, but by making variety subspaces comparable, it is possible to quantify task similarity in a linguistically interpretable manner.

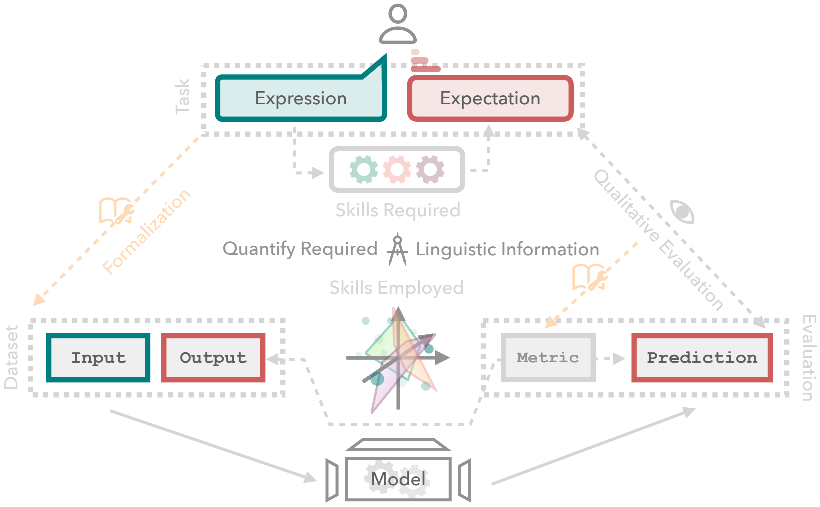


Figure 2.5: **Establishing Trust in NLP** (adapted from Litschko et al., 2023). With formalization breaking down across the NLP cycle, from tasks → datasets → evaluation, quantifying linguistic information can help identify which required skills are actually employed by models.

2.2.3 Robustness and Trustworthiness

Even without active measures to mitigate distributional divergence, LMs provide remarkably transferable initializations across languages, domains and tasks (Peters et al., 2018; Howard and Ruder, 2018; Devlin et al., 2019; Conneau et al., 2020; Brown et al., 2020). Nonetheless, fine-tuning or even re-training LMs on target-like data has repeatedly been shown to be crucial for the best possible downstream performance (Dai et al., 2020; Gururangan et al., 2020), regardless of the recent increases in scale (Ling et al., 2023).

Understanding how robustly a model transfers to new settings, as well as which skills it employs in the process, are key to establishing its trustworthiness. Indeed, Hays (1979) defines trust as "knowledge of origin as well as from knowledge of functional capacity". We concretize this definition for contemporary, data-driven NLP in Litschko et al. (2023) to include desiderata for statistical LMs. To establish model trustworthiness, NLP relies on the overall cycle shown in Figure 2.5, where tasks are formalized as machine-readable datasets, and models are subsequently evaluated using metrics, that themselves are formalized versions of qualitative evaluation criteria. This traditional paradigm can provide knowledge of functional capacity, and even knowledge of origin, if employed in a targeted manner (e.g., linguistic benchmarks as in Wang et al., 2019b; Ribeiro et al., 2020; Warstadt et al., 2020). However, similarly to how contemporary models rely less on manual feature engineering (as in Figure 2.4), tasks have also started to become less rigorously formalized. Instead of natural language expressions mapping to a discrete output label space, tasks are now more often phrased as instructions, for which the LM generates an open response (e.g., "write a polite email to my supervisor, asking for time off tomorrow"). Whether or not the task's expectation has been fulfilled (i.e., knowledge of functional capacity) is no longer measurable by discrete benchmarks, but requires a more holistic evaluation. Effectively, models are now expected to not only behave robustly against input variation, but also output variation.

In order to establish a model's trustworthiness in this context, we argue that quantitative measures of different variety dimensions can provide knowledge of both origin and functional capacity. Using our definition of tasks as variations over output space (see Section 2.2.2), any NLP task can be expressed as a mixture of specific linguistic skills, i.e., the skills required to solve it. Disentangling the actual skills, which a model employs to generate a prediction, is confounded by many factors (Schlangen, 2021). Nonetheless, measuring how much task-relevant linguistic information is present in the model can provide an a priori estimate of its functional capacity as well as the origin thereof. This approach allows us to answer questions regarding a model's trustworthiness, e.g.: Does the model encode enough syntactic information, such that it could generate a well-formed sentence in the target language? Does it differentiate between genres, such as emails versus academic papers? Does it model different levels of politeness well enough to address my supervisor?

An Approach to Quantifying Variation 3

Building on our definitions of typological, domain and task variation from Chapter 2, we next investigate methods for quantifying these properties while maintaining interpretability. Given that linguistic variation within the context of contemporary NLP corresponds to distributional divergence (Section 2.2), we start from the highest level of abstraction and review model transferability as measured by standard performance metrics (Section 3.1). Turning to variation across individual data points, we survey prior methods, which leverage the representational similarity of LM embeddings (Section 3.2). Finally, we introduce our own framework for quantifying linguistic variation by probing for interpretable representational subspaces (Section 3.3).

3.1 Transferability as a Proxy

Even if not explicitly labeled as measures of linguistic variation, NLP provides multiple implicit candidates: At the most general level, an LM will have worse performance when evaluated on a language variety it has not been designed/trained for, due to increased distributional divergence. This performance-Δ therefore corresponds to a crude, yet common way to quantify the similarity of two datasets. Compared to qualitative estimates of variation, it already enables numerical comparisons, and can further be applied to a wide range of linguistic dimensions, including cross-lingual, cross-domain, and even cross-task transfer (depending on the model architecture and task formulation).

Classification For classification tasks, performance is typically measured using standard metrics such as the ratio of correct over incorrect predictions (i.e., accuracy), or the slightly more balanced F1-score:

$$F_1 = \frac{2\text{TP}}{(\text{TP} + \text{FP}) + (\text{TP} + \text{FN})} \quad , \qquad 3.1$$

which corresponds to the harmonic mean over the ratio of correct classifications (true positives; TP) with respect to overassignments of a class (false positives; FP) and missed assignments of a class (false negatives; FN)—i.e., precision and recall, respectively. For tasks with more than one class, two different methods for aggregating the F1-score exist: *Micro-F1* counts the total number of TP, FP, FN across all classes before inserting them into Equation 3.1, while *macro-F1* computes each class-wise F1-score before computing the arithmetic mean over those F1s. Which averaging method to use depends on whether class or instance-level accuracy is more important. For example, in a highly imbalanced three-label classification task with class ratios 1/1/98, a majority classifier would yield a micro-F1 of 98%, while its macro-F1 score would be ~33%, since it labels two thirds of the classes incorrectly. In our subsequent experiments, we explicitly label each type of F1 to ensure the appropriate interpretation of the results.

Dependency Parsing When evaluating a model's ability to parse syntax, different performance metrics apply. For dependency parsing in particular, the de-facto standard is given by labeled and unlabeled attachment scores (LAS/UAS). They are closely related to the F1-score, as each node in a dependency tree, despite being part of a graph structure, only has one definitive parent edge + its associated relation, i.e., a single label. With respect to the total number of edges (corresponding to the number of lexical units), LAS therefore defines a TP as any node that is connected to the correct parent via the correct relation, while for UAS, the correct connection alone is sufficient. Within this formulation, there are multiple additional factors, such as the granularity of dependency relations, as well as whether to use macro/micro-F1. Following prior work, we make use of the widely-used official evaluation script of the CoNLL 2017 shared task, which uses top-level relations and micro-F1 averaging (Zeman et al., 2017). Together with the standardized annotation formalism of Universal Dependencies, LAS/UAS allow us to quantify high-level syntactic similarity across a broad range of language varieties via the proxy of transferability.

3.2 Representational Similarity

Measuring variation via transfer performance allows for very general comparisons, as it can be applied even across model architectures. However, it is relatively imprecise and consolidates all variety dimensions in a single number using post-hoc evaluation. Since contemporary NLP models universally rely on latent, task-related information from pre-trained embedding spaces, their representations of the input data can yield an estimate of variation a priori. Concretely, the geometric vector similarity of different data points already encodes some notion of similarity out-of-the-box. To measure the distance between two d-dimensional vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, a commonly employed measure is cosine similarity:

$$\text{sim}(\boldsymbol{a}, \boldsymbol{b}) = \cos(\theta) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{a}\| \|\boldsymbol{b}\|} = \frac{\sum_{i=1}^{d} a_i b_i}{\sqrt{\sum_{i=1}^{d} a_i^2} \sqrt{\sum_{i=1}^{d} b_i^2}} \quad . \qquad 3.2$$

Compared to the similarly common Euclidean distance $\|\boldsymbol{a} - \boldsymbol{b}\|$, cosine similarity is invariant to the magnitude of the vectors, and solely measures the cumulative angle between the vectors across each dimension. While measuring data similarity via their LM representations does not require task-specific model training and transfer evaluations, the vectors encode many notions of similarity simultaneously, such that interpreting the resulting number as well as how it relates to the downstream robustness of a model is difficult.

This issue extends across all ML disciplines, with efforts in Computer Vision to, for instance, disentangle task and domain-specific features using dataset metadata (Peng et al., 2020; Jomaa et al., 2021). So far, these methods have assumed access to domain annotations with clear distinctions, i.e., photographs versus drawings. However, as seen in Chapter 2, Variety Space for language is much more difficult to delineate, resulting in far less comprehensive, and noisier domain annotations for most NLP datasets (see also Chapter 6).

Typology In addition to the semi-qualitative language vectors from lang2vec (Littell et al., 2017), parallel work has examined learning typological representations via regression on word order features (Baylor et al., 2024), or as a side-effect of auxiliary tasks, such as language modeling (Tsvetkov et al., 2016; Malaviya et al., 2017; Östling and Tiedemann, 2017) and translation (Ha et al., 2016; Johnson et al., 2017). The resulting language representations have enabled comparisons on a continuous spectrum, and have further been found to cluster into phylogenic hierarchies matching linguistics literature (Östling and Tiedemann, 2017). Our work does not aim for the level of coverage obtained by these approaches, but instead specifically focuses on measuring the level of syntactic similarity in self-supervised representations, in absence of discrete language-identifiers.

Domain Given its more ambiguous nature, domain lacks comparable approaches. Practitioners thus rely on higher-level measures of domain similarity, based on correlated features, such as PoS tags, or distributional semantics. Ramesh Kashyap et al. (2021) provide an extensive survey of such metrics and broadly divide them into information theoretic (e.g. KL-divergence, Wasserstein distance), geometric (e.g. cosine similarity) and higher-order measures (e.g. Proxy-A-Distance). These measures are applied to distributions over lexical items, PoS tags and also over continuous word representations from LMs. Ramponi and Plank (2020) further survey how such measures are applied in practice: They broadly differentiate between model-centric approaches, which focus on increasing the robustness of model training, data-centric approaches, which focus on training-data selection, as well as hybrids of the two. For these purposes, continuous, vectorized representations of the data are especially useful, since they do not require manual annotation. On the modeling side, these representations have been successfully applied to regularizing model training using data with divergent embeddings (Xu and Lapata, 2019; Xu et al., 2021). Similarly, they can be used to detect out-of-domain data during inference to help calibrate a model's trustworthiness (Tan et al., 2019; Pokharel and Agrawal, 2023), as well as to predict transfer performance (Pogrebnyakov and Shaghaghian, 2021). If the target domain is known, data-driven vector similarity has also been shown to aid training data selection and improve robustness on the unseen target-like data (Ruder and Plank, 2017; Aharoni and Goldberg, 2020; van der Goot et al., 2021a; Müller-Eberstein et al., 2021a).

Overall, across a wide range of NLP tasks and datasets, manual features are still competitive for quantifying variation, but data-driven representations are becoming more common and increasingly effective (Ramponi and Plank, 2020; Ramesh Kashyap et al., 2021). The largest open issue with raw representational similarity is their lack of interpretability, and hence lack of control over what kind of similarity is being measured. This issue is especially pronounced for heterogeneous datasets, which mix multiple sources of variation (Pogrebnyakov and Shaghaghian, 2021). Monolingual embedding similarity, for instance, implicitly controls for typology, while selecting for domain-like features. Multilingual embeddings already make it impossible to clearly distinguish between typological and domain similarity, as it is unclear what each vector dimension represents. Any more specific differentiations, such as genres, topics, registers, etc., require even more granular control of the source data, which is typically unavailable, and may also be confounded across variety dimensions. Nonetheless, LM latent spaces likely encode information related to these specific dimensions, and as such, we require methods to recover them from the overall embedding space mixture.

3.3 Variety Subspaces via Probing

Data-driven representational similarity provides an information-dense mechanism for detecting language variability. For the purposes of interpretability, however, it is too imprecise. We therefore propose probing as a general methodology for identifying interpretable subspaces in Variety Space, within which we can perform quantitative comparisons along qualitatively defined linguistic dimensions.

What Is A Probe? Probing itself was initially developed to better understand whether task-relevant information is encoded in the latent representations of pre-trained ML models (Alain and Bengio, 2017). Depending on the task at hand, a wide array of probe formulations exist, which share the general properties of being architecturally simpler than the host model, and being trained from scratch, while the underlying host model is kept fixed. In NLP, the input to a probe typically consists of token-level embeddings from a pre-trained LM, which are mapped to target labels for the property of interest (e.g., PoS tags). What level of simplicity constitutes a probe has been a topic of debate, with opinions diverging regarding, e.g., the inclusion of non-linearities in the probe model (Conneau et al., 2018a; Liu et al., 2019a). Regardless of the probe's exact architecture, the relative amount of retrieved information has been shown to be similar (Qian et al., 2016; Belinkov et al., 2017), with the larger issue being the differentiation of actual linguistic information versus spurious correlations (Zhang and Bowman, 2018; Hewitt and Liang, 2019; Voita and Titov, 2020).

Probes as Subspaces In contrast to probing LMs for their own sake, we propose a meta-framework within which the LM and probe jointly form a tool for measuring linguistic variation in an interpretable way. Via self-supervised Representation Learning, the LM first constructs a high-dimensional latent space within which a data-driven notion of Variety Space is encoded. The overall variation between two sets of data can be quantified by representational divergence in this space. To identify variation across specific variety dimensions, we propose probing the general embedding space for information that is correlated with these dimensions—i.e., typology via syntax and domain via genre. Note that, in addition to using probes to measure the amount of relevant linguistic information, we are also interested in using the probes themselves as subspaces to perform relative comparisons in. Intuitively, each probe maps information from all dimensions of the LM latent space into a manifold of lower rank, within which it is easier to separate different classes of the linguistic property being probed for. As such, similarity within this subspace allows us to to perform comparisons of data with respect to specific, interpretable variety dimensions. Going one step further, we also examine the probes themselves as characterizations of task-specific subspaces. This allows us to quantify task similarity, since the amount of overlap between subspaces corresponds to how similar the types of linguistic information are, that are required to map task inputs to their respective outputs.

Towards these goals, we focus less on one individual probe formalism, task, or variational dimension, and rather examine a broader

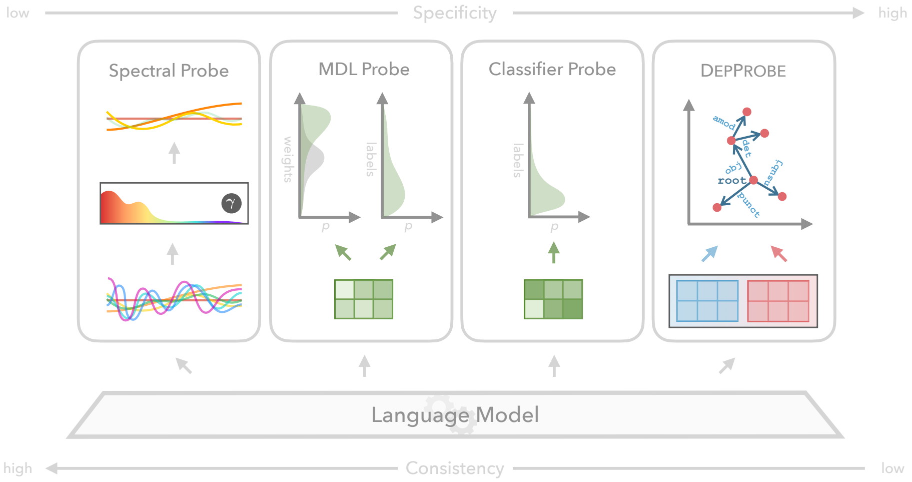


Figure 3.1: **Overview of Probes** in this work, sorted by task specificity and cross-probe consistency. All probes take LM latent representations as their input, but operate on information frequencies (Spectral Probe), minimum description lengths (MDL Probe), standard classification (Classifier Probe), or dependency information (DEPPROBE).

range of complementary methods. Figure 3.1 illustrates our categorization of these formalisms under the umbrellas of high-specificity probes (Section 3.3.1), which aim to extract as much task-correlated information as possible, as well as high-consistency probes (Section 3.3.2), which are more focused on keeping the probes themselves comparable across LMs and tasks. The main commonality across our proposed probing methods is their architectural simplicity with respect to the LMs they analyze, as well as the ability to compare the resulting probes and their subspaces with each other.

3.3.1 Probes with High Specificity

For their respective tasks, high-specificity probes aim to extract as much correlated information from an LM as possible. This implies that, despite not being designed for performance, these probes would achieve higher relative accuracy for distinguishing between different classes of a task, when compared to an alternative formulation.

Classifier Probe In its simplest form, this type of probe can be formulated as a linear classification over the output space of the task in question (Alain and Bengio, 2017). Given a labeled sentence s, consisting of the words $\{w_1,\dots,w_N\}$ and the corresponding labels $\{l_1,\dots,l_N\}$ (with c possible classes), the probe $\theta \in \mathbb{R}^{d\times c}$ is trained to map the corresponding d-dimensional LM latent vectors $\{\boldsymbol{h}_1,\dots,\boldsymbol{h}_N\}$ to logits, which maximize the probability of the correct label, by minimizing the cross-entropy loss:

$$\begin{aligned}\mathscr{L}_{\mathrm{XE}} &= -\frac{1}{N}\sum_{i=1}^{N}\log p_\theta(l_i|w_i)\\ &= -\frac{1}{N}\sum_{i=1}^{N}\theta^T\boldsymbol{h}_i \quad .\end{aligned} \tag{3.3}$$

In this way, the probe characterizes a linear subspace within which data with the same class are mapped closer together, and data with different labels are located further apart. Simultaneously, the subdimension with the highest magnitude corresponds to the inferred label for a given word, such that we can evaluate performance with respect to ground truth labels using the standard F1-score. While the use of non-linear classifiers is not uncommon, we opt specifically for a linear transformation, as it directly translates the LM's latent dimensions into the task's output subspace. This further enables well-defined comparisons of entire probes in linear space—something that is not possible using even a small Multi-layer Perceptron (Section 3.3.3).

Dependency Probe While most NLP tasks can be reduced to classification (i.e., a mapping of latent states to output logits), some types of linguistic information require alternative formulations. The task of structured prediction, which includes extracting syntactic dependencies between words in a sentence, falls into this category. In Part II, we therefore introduce dependency probing (DEPPROBE), which builds on structural probing (Hewitt and Manning, 2019) to map LM latent representations directly onto dependency trees. Correspondingly, the loss function is specific to this task, and includes components reflecting tree structure and relation types, with the full formulation detailed in Chapter 4. In the context of subspaces, the first loss term aims to learn a subspace θ_B, in which vector distances d_B correspond to tree distances d_P, while θ_R corresponds to a subspace, where words in similar syntactic relations are grouped together:

$$\begin{aligned}\mathscr{L}_{\text{DP}} &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|d_P(w_i, w_j) - d_B(\boldsymbol{h}_i, \boldsymbol{h}_j)\right| - \frac{1}{N}\sum_{i=1}^{N}\log p_{\theta_R}(l_i|w_i) \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|d_P(w_i, w_j) - \|\theta_B^T\boldsymbol{h}_i - \theta_B^T\boldsymbol{h}_j\|\right| - \frac{1}{N}\sum_{i=1}^{N}\theta_R^T\boldsymbol{h}_i \quad .\end{aligned} \tag{3.4}$$

This formulation allows us to extract fully labeled and directed dependency trees, equivalently to a full parser, allowing us to measure UAS and LAS. The transfer performance of these probes across languages can subsequently be used as a proxy measure of linguistic variation (as motivated in Section 3.1). Furthermore, analogously to linear classifier probes, the linearity of DepProbe enables us to compare how similarly syntactic information is represented across languages by comparing their respective θ_B and θ_R.

3.3.2 Probes with High Consistency

An issue using data-driven representations is that LMs trained on different data will learn to encode Variety Space differently. Due to the highly chaotic nature of training non-linear models, this issue applies even to models, which have been trained on the same data, but starting from different random initializations (Sellam et al., 2022; Müller-Eberstein et al., 2023). As such, measurements of representational and probe similarity are consistent within the same LM, but not across them. By leveraging probing methodologies, which prioritize cross-probe consistency over specificity and accuracy, we therefore aim to enable comparisons of variety subspaces independently of specific LM setups.

Information-theoretic Probing As even random representations can be mapped to labels with high accuracy (e.g., by applying the most common PoS-tag to a random word embedding consistently), Voita and Titov (2020) originally proposed information-theoretic probing as a method to measure representational efficiency directly. Different from evaluating classification accuracy via cross-entropy alone, the approach adds a KL-divergence term which measures how much the probe itself must deviate from a canonical form in order to achieve the current level of performance. This total entropy is consolidated in the minimum description length (MDL), i.e., the minimum number of bits required to transmit the data plus the probe:

$$\begin{aligned}\mathscr{L}_{\text{MDL}} &= -\,\mathbb{E}_{\theta\sim\beta}\left[\sum_{i=1}^{N}\log p_\theta(l_i|w_i)\right] + \text{KL}(\beta||\gamma)\\ &= -\,\mathbb{E}_{\theta\sim\beta}\left[\sum_{i=1}^{N}\theta^T\boldsymbol{h}_i\right] + \text{KL}(\beta||\gamma) \quad .\end{aligned} \tag{3.5}$$

While the core cross-entropy term remains equivalent to the standard classifier probe, the information-theoretic probe θ follows a Bayesian formulation, and is sampled from a distribution over probes parameterized by β. This means that cross-entropy is estimated using the current expectation over probes, and represents the accuracy with which $\theta \sim \beta$ models the data. In addition, the KL-divergence term computes the amount β needs to deviate from a canonical prior distribution γ, in order to achieve the current level of cross-entropy. Intuitively, LM representations which already encode relevant information out-of-the-box require a less complex mapping to the task's labels than uncorrelated, random representations.

In Chapter 9, we apply information-theoretic probing to comparing LM representations across tasks, as well as pre-training time. For this purpose, we leverage another implicit property of the MDL formulation: First, although MDL can be applied to any probe architecture, we retain a linear classification approach, in order to have θ characterize a linear subspace, for which well-defined comparisons in Euclidean space exist. Second, we build on MDL's additional KL-regularization objective, which ensures that all probes aim to follow the same canonical prior. By further employing a sparsity-inducing prior for γ (Figueiredo, 2001; Louizos et al., 2017), we encourage linear probes which aim for high accuracy, while transforming as few original LM dimensions as possible, using small scaling factors. This makes the resulting probes more consistent, and more geometrically comparable—even across different LM checkpoints and tasks.

Spectral Probing A commonality across the aforementioned probing approaches is their rigidity with respect to the underlying LM representations. Even with regularization such as MDL, it remains difficult to compare probes across languages or even model initializations, due to their dependence on the exact numerical values of the underlying LM embeddings, i.e., *where* certain information is encoded.

With Spectral Probing (Chapter 10), we focus on a more abstract, yet highly consistent, signal of linguistic information: namely, time. Intuitively, different tasks rely on linguistic information across different contextual scales, e.g., words, phrases, sentences. Whether for English or Japanese, the topic is less likely to change within a sentence than the syntactic functions at the scale of sub-phrases. In terms of contextualized embeddings, these differences across time can be interpreted as information encoded at different frequencies, i.e., embedding values which are more correlated at the phrase level for syntax, while values relating to topic should be correlated consistently across the entire sentence. Based on the finding that PoS-tagging and topic classification can be decomposed into different frequencies for English (Tamkin et al., 2020), we introduce a continuous spectral probing method for extracting full spectrograms automatically.

Probing for frequency spectra follows a standard cross-entropy objective, however, the sequence of contextualized input embeddings $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N\}$ is first decomposed into its composite frequencies using an invertible transformation $f(\cdot)$. The Spectral Probe corresponds to a frequency filter $\boldsymbol{\gamma} \in \mathbb{R}^N$, which learns to scale each composite frequency according to its importance to the probed property. The scaled frequencies are then recomposed into a filtered sequence of embeddings $\{\boldsymbol{h}'_1, \ldots, \boldsymbol{h}'_N\}$ by applying the inverse function $f^{-1}(\cdot)$:

$$\begin{aligned}\mathscr{L}_{\text{SP}} &= -\frac{1}{N}\sum_{i=1}^{N} \log p_{\boldsymbol{\gamma},\theta}(l_i | w_i) \\ &= -\frac{1}{N}\sum_{i=1}^{N} \theta^T \boldsymbol{h}'_i \\ &= -\frac{1}{N}\sum_{i=1}^{N} \theta^T f^{-1}\left(\boldsymbol{\gamma} f(\boldsymbol{h}_i)\right) \quad .\end{aligned} \tag{3.6}$$

In this formulation, the classifier θ is, once again, model specific. However, the scale of each frequency in $\boldsymbol{\gamma}$ is independent of the exact values in $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N\}$, and only depends on the importance of longer versus shorter-scale consistencies. The resulting spectrograms form *frequency subspaces*, which not only link directly to the linguistically interpretable dimension of time, but also allow for cross-task comparisons, which are highly stable across languages (Chapter 10).

3.3.3 Subspace Comparisons

All of the aforementioned probing methods define subspaces in the overall LM embedding space, within which similarity corresponds to an interpretable linguistic property, i.e., geometric subspaces for linear probes and frequency subspaces for the Spectral Probe. By comparing probe performance, representations, and subspaces, we can thus operationalize similarity between data, models and tasks in an interpretable and computationally efficient manner:

Probe Performance The transferability of a model trained on one language variety to another provides the most practical measure of similarity, but does not scale well across many source-target combinations due to its high data and compute requirements. As such, probe transferability provides a parameter-efficient approximation that does not require full LM tuning (Chapter 4). Similarly, probes can also be applied to prototyping which LM may be best suited for a specific target language by probing for the amount of language-specific information a priori (Chapter 5). Using high-consistency probes, we can further rank how likely an LM will perform well on a variety of downstream tasks (Bassignana et al., 2022). These approaches for predicting model performance prior to full fine-tuning already provide a more evidence-based solution compared to the typical method of relying solely on practitioner intuition regarding which base model to deploy.

Subspace Representations In contrast to performance, which approximates variation at the level of datasets, linguistic similarity at the granularity of individual data points is typically measured using vector similarity in LM latent space. As these comparisons do not specify how exactly two data are similar, we propose using probes to first extract specific linguistic subspaces, within which vector similarity corresponds to interpretable properties. These subspaces can be extracted using any of the aforementioned probing methods, or by tuning an LM to amplify a specific dimension of variation (Chapter 8). Representational similarity within these subspaces can still be measured using standard geometric measures (e.g., cosine similarity), and corresponds to similarity with respect to the property being probed for. Intuitively, subspaces scale up relevant, and scale down irrelevant dimensions from the original embedding space. Our approach of leveraging probes as subspaces thus extends their utility beyond measuring the presence of specific linguistic information, to enabling quantitative, yet interpretable, comparisons of individual data points.

Subspace Similarity Compared to performance or embedding comparisons, we propose using probes themselves to compare language varieties and tasks more holistically. This approach once again builds on the fact that probes characterize a subspace within which information correlated with the linguistic property or task in question is particularly salient. By further employing probes, which have well-defined comparison metrics (i.e., in linear or frequency space), we are able to directly compare how much they overlap with respect to the types of information they use from the original embedding space (Chapters 4, 9 and 10). Notably, this differs from comparing data points via their vector similarity, as these individual comparisons cannot be representative of the language or task in their entirety. In addition, compared to transfer performance, subspace similarity is much more efficient and interpretable, as it allows us to disentangle and target similarity with respect to specific linguistic properties.

For comparisons of linear subspaces, we turn to principal subspace angles (SSAs; Knyazev and Argentati, 2002), which allow us to measure the distance between two linear transformations (i.e., linear probes in our case) $\theta_A \in \mathbb{R}^{d \times p}$ and $\theta_B \in \mathbb{R}^{d \times q}$. It is based on the magnitude of the transformation mapping one matrix to another, and intuitively corresponds to the amount of 'energy' required for this process. Using the orthonormal bases $Q_A = \text{orth}(\theta_A)$ and $Q_B = \text{orth}(\theta_B)$ of each matrix to compute the transformation magnitudes $M = Q_A^T Q_B$ further ensures linear invariance. This is an important characteristic of this measure, as it ensures robustness against cases in which one subspace simply corresponds to a re-scaling and/or rotation of the other. The final angle is obtained by converting M's singular values $U\Sigma V^T = \text{SVD}(M)$ into values between 0° and 90° (i.e., similar/dissimilar):

$$\text{SSA}(\theta_A, \theta_B) = \arccos(\text{diag}(\Sigma)) \quad . \tag{3.7}$$

In the following, we employ SSAs to compare representational subspaces across languages (Chapter 4), as well as across LM checkpoints and tasks (Chapter 9).

For comparisons of frequency subspaces (Chapter 10), we define a filter overlap metric, that measures the degree to which the same frequencies are up or down-weighted by each Spectral Probe. Given m composite frequencies, it is computed via the percentage-normalized L1-distance between the probes $\boldsymbol{\gamma}_A \in \mathbb{R}^m$ and $\boldsymbol{\gamma}_B \in \mathbb{R}^m$:

$$\text{overlap}(\boldsymbol{\gamma}_A, \boldsymbol{\gamma}_B) = 1 - \frac{\sum_{i=1}^{m} \left|\gamma_{A,i} - \gamma_{B,i}\right|}{\max(\boldsymbol{\gamma}_A, \boldsymbol{\gamma}_B)} \quad . \tag{3.8}$$

This yields a measure between 0 and 1, which corresponds to the prioritization of completely different versus completely matching + equally weighted frequencies, respectively.

Each of the aforementioned approaches for comparing either performance, representations, or subspaces allows us to quantify linguistic variation with different degrees of granularity, and towards different purposes: For in-practice applicability, estimating downstream performance is most appropriate, while for comparisons of individual data points, representational similarity is indispensable. Finally, our newly proposed approach of training probes to extract fully comparable variety subspaces bridges the former granularities to enable holistic and interpretable comparisons within Variety Space—the applicability of which we will demonstrate for typological, domain and task variation in the following chapters.

Part II

Typological Variation

Probing for Labeled Dependency Trees 4

The work presented in this chapter is based on: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022a. Probing for labeled dependency trees. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7711–7726, Dublin, Ireland. Association for Computational Linguistics.

Abstract

Probing has become an important tool for analyzing representations in Natural Language Processing (NLP). For graphical NLP tasks such as dependency parsing, linear probes are currently limited to extracting undirected or unlabeled parse trees which do not capture the full task. This work introduces DEPPROBE, a linear probe which can extract *labeled* and *directed* dependency parse trees from embeddings while using fewer parameters and compute than prior methods. Leveraging its full task coverage and lightweight parametrization, we investigate its predictive power for selecting the best transfer language for training a full biaffine attention parser. Across 13 languages, our proposed method identifies the best source treebank 94% of the time, outperforming competitive baselines and prior work. Finally, we analyze the informativeness of task-specific subspaces in contextual embeddings as well as which benefits a full parser's non-linear parametrization provides.

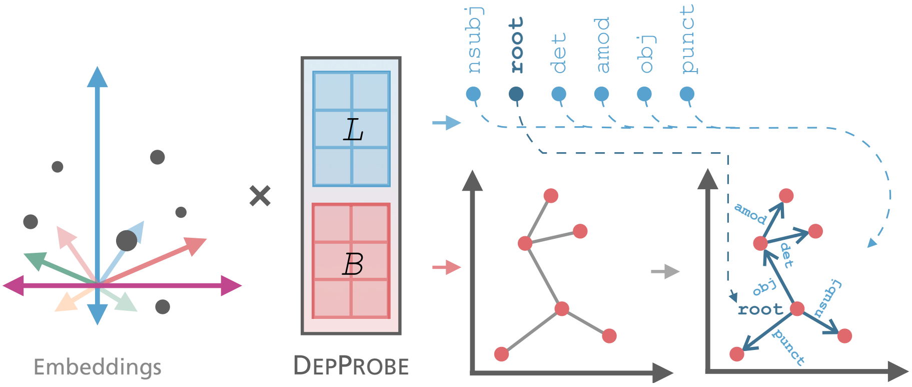


Figure 4.1: **DEPPROBE** extracts tree structure using transformation B, labels using L and infers directionality using `root`, based on contextualized embeddings.

4.1 Introduction

Pre-trained, contextualized embeddings have been found to encapsulate information relevant to various syntactic and semantic tasks out-of-the-box (Tenney et al., 2019a; Hewitt and Manning, 2019). Quantifying this latent information has become the task of *probes* — models which take frozen embeddings as input and are parametrized as lightly as possible (e.g. linear transformations). Recent proposals for edge probing (Tenney et al., 2019a) and structural probing (Hewitt and Manning, 2019) have enabled analyses beyond classification tasks, including graphical tasks such as dependency parsing. They are able to extract dependency graphs from embeddings, however these are either undirected (Hewitt and Manning, 2019; Maudslay et al., 2020) or unlabeled (Kulmizev et al., 2020), thereby capturing only a subset of the full task.

In this work, we investigate whether this gap can be filled and ask: *Can we construct a lightweight probe which can produce fully directed and labeled dependency trees?* Using these trees, we further aim to study the less examined problem of transferability estimation for graphical tasks, extending recent work targeting classification and regression tasks (Nguyen et al., 2020; You et al., 2021). Specifically: *How well do our probe's predictions correlate with the transfer performance of a full parser across a diverse set of languages?*

To answer these questions, we contribute DepProbe (Figure 4.1), the first linear probe to extract directed and labeled dependency trees while using fewer parameters than prior work and three orders of magnitude fewer trainable parameters than a full parser (Section 4.3). As this allows us to measure labeled attachment scores (LAS), we investigate the degree to which our probe is predictive of cross-lingual transfer performance of a full parser across 13 typologically diverse languages, finding that our approach chooses the best transfer language 94% of the time, outperforming competitive baselines and prior work (Section 4.4). Finally, we perform an in-depth analysis of which latent information is most relevant for dependency parsing as well as which edges and relations benefit most from the expressivity of the full parser (Section 4.5).[1]

[1] Code available at https://personads.me/x/acl-2022-code.

4.2 Related Work

Given the ubiquitous use of contextualized embeddings (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021), practitioners have turned to various methods for analyzing their linguistic features (Rogers et al., 2020). Hewitt and Manning (2019) examine these intrinsic properties in greater detail for English dependency parsing using a *structural probe*, finding that *undirected* dependency graphs are recoverable from BERT by learning a linear transformation on its embeddings (Section 4.3.1).

Extending the structural probe of Hewitt and Manning (2019) to 12 languages, Chi et al. (2020) extract *undirected* dependency graphs from mBERT (Devlin et al., 2019), further showing that head-to-child difference vectors in the learned subspace cluster into relations from the Universal Dependencies taxonomy (de Marneffe et al., 2014).

Building on both the structural and tree depth probes (Hewitt and Manning, 2019), Kulmizev et al. (2020) extract *directed* dependency graphs from mBERT for 13 languages (Section 4.3.2). Further variations to structural probing include regularization of the linear transformation (Limisiewicz and Mareček, 2021) as well as alternative objective functions (Maudslay et al., 2020).

None of the proposed linear probing approaches so far are able to produce full dependency parse trees (i.e. directed and labeled), however the closer a probe approximates the full task, the better it quantifies relevant information (Maudslay et al., 2020). It would for example be desirable to estimate LAS for parsing a target treebank with a model trained on a different source without having to train a resource-intensive parser (e.g. Dozat and Manning, 2017) on each source candidate. Although performance prediction methods for such scenarios exist, they typically do not cover graph prediction (Nguyen et al., 2020; You et al., 2021).

In order to bridge the gap between full parsers and unlabeled probes, in addition to the gap between full fine-tuning and lightweight performance prediction, this work proposes a linear probe which can extract *labeled* and *directed* dependency parse trees while using less compute than prior methods (Section 4.3). We use our probe's LAS to evaluate its predictive power for full parser performance and leverage its linear nature to investigate how dependencies are represented in subspaces of contextual embeddings (Section 4.5).

4.3 Probing for Dependencies

In order to construct a directed and labeled dependency parse tree for a sentence s consisting of the words $\{w_0, \ldots, w_N\}$, we require information on the presence or absence of edges between words, the directionality of these edges $(\overrightarrow{w_i, w_j})$, and the relationships $\{r_0, \ldots, r_N\}$ which they represent. Using the contextualized embeddings $\{\boldsymbol{h}_0, \ldots, \boldsymbol{h}_N\}$ with $\boldsymbol{h}_i \in \mathbb{R}^e$, prior probing work has focused on the first step of identifying edges (Section 4.3.1) and later directionality (Section 4.3.2). In this work, we propose a probe which completes the final relational step (Section 4.3.3) and simultaneously provides a more efficient method for identifying directionality (Section 4.3.4).

4.3.1 Undirected Probing

The structural probe introduced by Hewitt and Manning (2019) recovers the first piece of information (i.e. the undirected graph) remarkably well. Here, the probe is a linear transformation $B \in \mathbb{R}^{e \times b}$ with $b < e$ which maps contextual embeddings into a subspace in which the distance measure

$$d_B(\boldsymbol{h}_i, \boldsymbol{h}_j) = \sqrt{(B\boldsymbol{h}_i - B\boldsymbol{h}_j)^T (B\boldsymbol{h}_i - B\boldsymbol{h}_j)} \tag{4.1}$$

between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ is optimized towards the distance between two words in the dependency graph $d_P(w_i, w_j)$, i.e. the number of edges between the words. For each sentence, the loss is defined as the mean absolute difference across all word pairs:

$$\mathscr{L}_B(s) = \frac{1}{N^2} \sum_{i=0}^{N} \sum_{j=0}^{N} \left| d_P(w_i, w_j) - d_B(\boldsymbol{h}_i, \boldsymbol{h}_j) \right|. \tag{4.2}$$

In order to extract an undirected dependency graph, one computes the distances for a sentence's word pairs using d_B and extracts the minimum spanning tree (Jarník, 1930; Prim, 1957; MST).

4.3.2 Directed Probing

Apart from the structural probe B, Hewitt and Manning (2019) also probe for tree depth. Using another matrix $C \in \mathbb{R}^{e \times c}$, a subspace is learned in which the squared L_2 norm of a transformed embedding $\|C\boldsymbol{h}_i\|_2^2$ corresponds to a word's depth in the tree, i.e. the number of edges from the root.

Kulmizev et al. (2020) combine the structural and tree depth probe to extract directed graphs. This directed probe (DirProbe) constructs a score matrix $M \in \mathbb{R}^{N \times N}$ for which each entry corresponds to a word pair's negative structural distance $-d_B(\boldsymbol{h}_i, \boldsymbol{h}_j)$. The shallowest node in the depth subspace C is set as root. Entries in M which correspond to an edge between w_i and w_j for which the word depths follow $\|C\boldsymbol{h}_i\|_2^2 > \|C\boldsymbol{h}_j\|_2^2$ are set to $-\infty$. A word's depth in subspace C therefore corresponds to edge directionality. The directed graph is built from M using Chu-Liu-Edmonds decoding (Chu and Liu, 1965; Edmonds, 1967).

DirProbe extracts directed dependency parse trees, however it would require additional complexity to label each edge with a relation (e.g. using an additional probe). In the following, we propose a probe which can extract both directionality and relations while using fewer parameters and no dynamic programming-based graph-decoding algorithm.

4.3.3 Relational Probing

The incoming edge of each word w_i is governed by a single relation. As such the task of dependency relation classification with l relations can be simplified to a labeling task using a linear transformation $L \in \mathbb{R}^{e \times l}$ for which the probability of a word's relation r_i being of class l_k is given by:

$$p(r_i = l_k | w_i) = \text{softmax}(L\boldsymbol{h}_i)_k \qquad 4.3$$

and optimization uses standard cross-entropy loss given the gold label r_i^* for each word w_i:

Algorithm 1: DepProbe Inference

```
1  input Distance matrix $D_B \in \mathbb{R}^{N \times N}$, $p(l_k|w_i)$ of relation label $l_k$
     given $w_i$
2  $w_r \leftarrow \underset{w_i}{\text{argmax}}\ p(\text{root}|w_i)$
3  $\mathcal{T}_w \leftarrow \{w_r\}, \mathcal{T}_e \leftarrow \{\}$
4  while $|\mathcal{T}_w| < N$ do
5      $w_i, w_j \leftarrow \underset{w_i, w_j}{\text{argmin}}\ D_B(w_i \in \mathcal{T}_w, w_j)$
6      $r_j \leftarrow \underset{l_k}{\text{argmax}}\ p(l_k|w_j)$ with $l_k \neq \text{root}$
7      $\mathcal{T}_w \leftarrow \mathcal{T}_w \cup \{w_j\}$
8      $\mathcal{T}_e \leftarrow \mathcal{T}_e \cup \{(\overrightarrow{w_i, w_j}, r_j)\}$
9  end
10 return $\mathcal{T}_e$
```

$$\mathcal{L}_L(s) = -\frac{1}{N}\sum_{i=0}^{N} \ln p(r_i^*|w_i) \ . \tag{4.4}$$

Should dependency relations be encoded in contextualized embeddings, each dimension of the subspace L will correspond to the prevalence of information relevant to each relation, quantifiable using relation classification accuracy (RelAcc).

4.3.4 Constructing Dependency Parse Trees

Combining structural probing (Section 4.3.1) and dependency relation probing (Section 4.3.3), we propose a new probe for extracting fully directed and labeled dependency trees (DepProbe). It combines undirected graphs and relational information in a computationally efficient manner, adding labels while requiring *less* parameters than prior unlabeled or multi-layer-perceptron-based approaches.

As outlined in Algorithm 1 and illustrated in Figure 4.1, DepProbe uses the distance matrix D_B derived from the structural probe B in conjunction with the relation probabilities of the relational probe L (line 1). The graph is first rooted using the word w_r for which $p(\text{root}|w_r)$

is highest (line 2). Iterating over the remaining words until all w_j are covered in $\mathscr{T}_w$, an edge is drawn to each word w_j from its head w_i based on the minimum distance in D_B. The relation r_j for an edge $(\overrightarrow{w_i, w_j}, r_j)$ is determined by taking the relation label l_k which maximizes $p(r_j = l_k | w_j)$ with $l_k \neq \text{root}$ (line 6). The edge is then added to the set of labeled tree edges $\mathscr{T}_e$. With edge directionality being inferred as simply pointing away from the root, this procedure produces a dependency graph that is both directed and labeled without the need for additional complexity, running in $\mathcal{O}(n^2)$ while dynamic programming-based decoding such as DIRPROBE have runtimes of up to $\mathcal{O}(n^3)$ (Stanojević and Cohen, 2021).

Constructing dependency trees from untuned embeddings requires the matrices B and L, totaling $e \cdot b + e \cdot l$ trainable parameters. Optimization can be performed using gradient descent on the sum of losses $\mathscr{L}_B + \mathscr{L}_L$. With $l = 37$ relations in UD, this constitutes a substantially reduced training effort compared to prior probing approaches (with subspace dimensionalities b and c typically set to 128) and multiple magnitudes fewer fine-tuned parameters than for a full biaffine attention parser.

4.4 Experiments

4.4.1 Setup

Parsers In our experiments, we use the deep biaffine attention parser (BAP) by Dozat and Manning (2017) as implemented in van der Goot et al. (2021b) as an upper bound for MLM-based parsing performance. As it is closest to our work, we further reimplement DIRPROBE (Kulmizev et al., 2020) with $b = 128$ and $c = 128$. Note that this approach produces directed, but unlabeled dependency graphs. Finally, we compare both methods to our directed and labeled probing approach, DEPPROBE with $b = 128$ and $l = 37$.

All methods use mBERT (Devlin et al., 2019) as their encoder ($e = 768$). For BAP, training the model includes fine-tuning the encoder's parameters, while for both probes they remain fixed and only the linear transformations are adjusted. This results in 183M tuned parameters for BAP, 197k for DIRPROBE and 127k for DEPPROBE. Hyperparameters are set to the values reported by the authors,[2] while for DEPPROBE we perform an initial tuning step in Section 4.4.2.

Target Treebanks As targets, we use the set of 13 treebanks proposed by Kulmizev et al. (2019), using versions from Universal Dependencies v2.8 (Zeman et al., 2021). They are diverse with respect to language family, morphological complexity and script (Appendix 4.7.1). This set further includes EN-EWT (Silveira et al., 2014) which has been used in prior probing work for hyperparameter tuning, allowing us to tune DEPPROBE on the same data.

Metrics We report labeled attachment scores (LAS) wherever possible (BAP, DEPPROBE) and unlabeled attachment scores (UAS) for all methods. For DEPPROBE's hyperparameters, we evaluate undirected, unlabeled attachment scores (UUAS) as well as relation classification accuracy (RelAcc). One notable difference to prior work is that we include punctuation both during training and evaluation — contrary to prior probing work which excludes all punctuation (Hewitt and Manning, 2019; Kulmizev et al., 2020; Maudslay et al., 2020) — since we are interested in the full parsing task.

Training Each method is trained on each target treebank's training split and is evaluated on the test split. For cross-lingual transfer, models trained on one language are evaluated on the test splits of all other languages without any further tuning. For DEPPROBE tuning (Section 4.4.2) we use the development split of EN-EWT.

BAP uses the training schedule implemented in van der Goot et al. (2021b) while DIRPROBE and DEPPROBE use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-3} which is reduced by a factor of 10 each time the loss plateaus (see also Hewitt and Manning, 2019).

Both probing methods are implemented using PyTorch (Paszke et al., 2019) and use mBERT as implemented in the Transformers library (Wolf et al., 2020). Each model is trained with three random initializations of which we report the mean.

[2] For better comparability, we use the best single layer reported by Kulmizev et al. (2020) instead of the weighted sum over all layers.

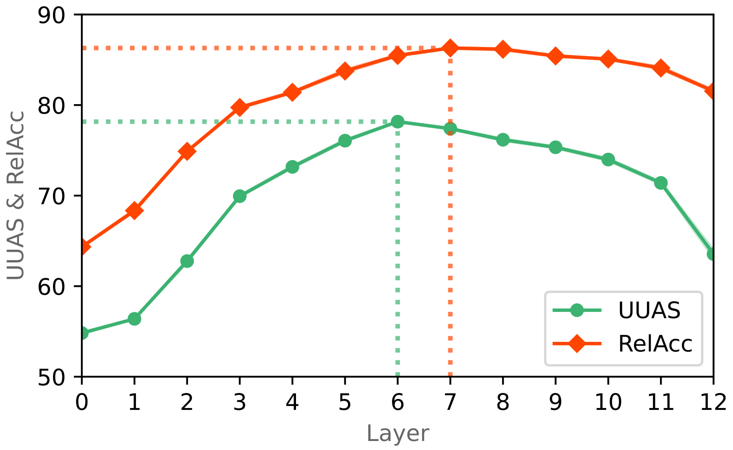


Figure 4.2: **Layer-wise Performance on EWT (Dev)** for DEPPROBE as measured by UUAS for the structural probe B and RelAcc for the relational probe L.

4.4.2 DEPPROBE Tuning

As prior work has repeatedly found that MLM layers encode different linguistic information, the layers which are most relevant for a probe's task are typically first identified (Tenney et al., 2019a; Hewitt and Manning, 2019). Following this paradigm, we train DEPPROBE on embeddings from each layer of mBERT. Layer 0 is equivalent to the first, non-contextualized embeddings while layer 12 is the output of the last attention heads. The probe is trained on EN-EWT and evaluated on its development split using UUAS for the structural transformation B (akin to Hewitt and Manning, 2019) as well as RelAcc for the relational transformation L.

Figure 4.2 shows that structure is most prevalent around layer 6 at 78 UUAS, corroborating the 6–8 range identified by prior work (Tenney et al., 2019a; Hewitt and Manning, 2019; Chi et al., 2020). Dependency relations are easiest to retrieve at around layer 7 with an accuracy of 86%. The standard deviation across initializations is around 0.1 in both cases. Based on these tuning results, we use layer 6 for structural probing and layer 7 for relational probing in the following experiments.

4.4.3 Parsing Performance

Figure 4.3 lists UAS for all methods and LAS for BAP and DepProbe both on target-language test data (=L) and zero-shot transfer targets (¬L). Table 4.3e further shows the mean results for each setting.

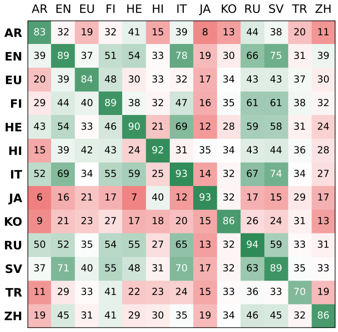
Unsurprisingly, the full parametrization of BAP performs best, with in-language scores of 88 LAS and 91 UAS. For zero-shot transfer, these scores drop to 35 LAS and 52 UAS, with some language pairs seeing differences of up to 85 points: e.g. JA → JA (93 LAS) versus AR → JA (8 LAS) in Figure 4.3a. This again confirms the importance of selecting appropriate source data for any given target.

Both probes, with their limited parametrization, fall short of the full parser's performance, but still reach up to 73 LAS and 79 UAS. DirProbe has a mean in-language UAS which is 3 points higher than for DepProbe, attributable to the more complex decoder. Due to DirProbe's output structures being unlabeled, we cannot compare LAS.

DepProbe reaches a competitive 67 UAS despite its much simpler decoding procedure and appears to be more stable for zero-shot transfer as it outperforms DirProbe by around 2 UAS while maintaining a lower standard deviation. Most importantly, it produces directed and *labeled* parses such that we can fully compare it to BAP. Considering that DepProbe has more than three orders of magnitude fewer tunable parameters, a mean in-language LAS of 60 is considerable and highlights the large degree of latent dependency information in untuned, contextual embeddings. For zero-shot transfer, the performance gap to BAP narrows to 13 LAS and 14 UAS.

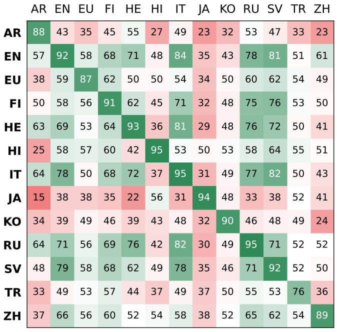
4.4.4 Transfer Prediction

Given that DepProbe provides a highly parameter-efficient method for producing directed, labeled parse trees, we next investigate whether its performance patterns are indicative of the full parser's performance and could aid in selecting an appropriate source treebank for a given target without having to train the 183 million parameters of BAP.



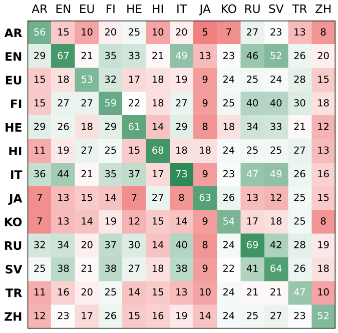
	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	83	32	19	32	41	15	39	8	13	44	38	20	11
EN	39	89	37	51	54	33	78	19	30	66	75	31	39
EU	20	39	84	48	30	33	32	17	34	43	43	37	30
FI	29	44	40	89	38	32	47	16	35	61	61	38	32
HE	43	54	33	46	90	21	69	12	28	59	58	31	24
HI	15	39	42	43	24	92	31	35	34	43	44	36	28
IT	52	69	34	55	59	25	93	14	32	67	74	34	27
JA	6	16	21	17	7	40	12	93	32	17	15	29	17
KO	9	21	23	27	17	18	20	15	86	26	24	31	13
RU	50	52	35	54	55	27	65	13	32	94	59	33	31
SV	37	71	40	55	48	31	70	17	32	63	89	35	33
TR	11	29	33	41	22	23	24	15	33	36	33	70	19
ZH	19	45	31	41	29	30	35	19	34	46	45	32	86

(a) BAP (LAS)



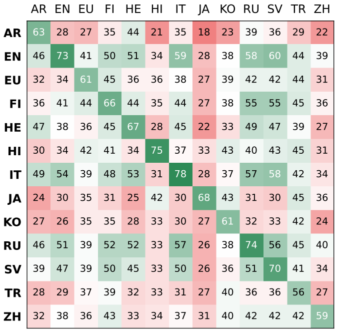
	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	88	43	35	45	55	27	49	23	32	53	47	33	23
EN	57	92	58	68	71	48	84	35	43	78	81	51	61
EU	38	59	87	62	50	50	54	34	50	60	62	54	49
FI	50	58	56	91	62	45	71	32	48	75	76	53	50
HE	63	69	53	64	93	36	81	29	48	76	72	50	41
HI	25	58	57	60	42	95	53	50	53	58	64	55	51
IT	64	78	50	68	72	37	95	31	49	77	82	50	43
JA	15	38	38	35	22	56	31	94	48	33	38	52	41
KO	34	39	49	46	39	43	48	32	90	46	48	49	24
RU	64	71	56	69	76	42	82	30	49	95	71	52	52
SV	48	79	58	68	62	49	78	35	46	71	92	52	50
TR	33	49	53	57	44	37	49	37	50	55	53	76	36
ZH	37	66	56	60	52	54	58	38	52	65	62	54	89

(b) BAP (UAS)



	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	56	15	10	20	25	10	20	5	7	27	23	13	8
EN	29	67	21	35	33	21	49	13	23	46	52	26	20
EU	15	18	53	32	17	18	19	9	24	25	24	28	15
FI	15	27	27	59	22	18	27	9	25	40	40	30	18
HE	29	26	18	29	61	14	29	8	18	34	33	21	12
HI	11	19	27	25	15	68	18	18	24	25	25	27	13
IT	36	44	21	35	37	17	73	9	23	47	49	26	16
JA	7	13	15	14	7	27	8	63	26	13	12	25	15
KO	7	13	14	19	12	15	14	9	54	17	18	25	8
RU	32	34	20	37	30	14	40	8	24	69	42	28	19
SV	25	38	21	38	27	18	38	9	22	41	64	26	18
TR	11	16	20	25	14	15	13	10	24	21	21	47	10
ZH	12	23	17	26	15	16	19	14	24	25	27	23	52

(c) DepProbe (LAS)

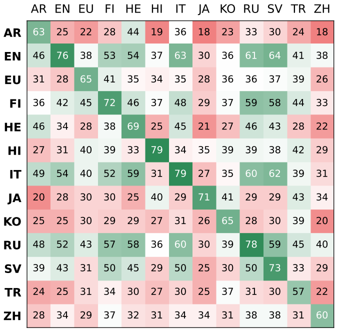


	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	63	28	27	35	44	21	35	18	23	39	36	29	22
EN	46	73	41	50	51	34	59	28	38	58	60	44	39
EU	32	34	61	45	36	36	38	27	39	42	42	44	31
FI	36	41	44	66	44	35	44	27	38	55	55	45	36
HE	47	38	36	45	67	28	45	22	33	49	47	39	27
HI	30	34	42	41	34	75	37	33	43	40	43	45	31
IT	49	54	39	48	53	31	78	28	37	57	58	42	34
JA	24	30	35	31	25	42	30	68	43	31	30	45	36
KO	27	26	35	35	28	33	30	27	61	32	33	42	24
RU	46	51	39	52	52	33	57	26	38	74	56	45	40
SV	39	47	39	50	45	33	50	26	36	51	70	41	34
TR	28	29	37	39	32	33	31	27	40	36	36	56	27
ZH	32	38	36	43	33	34	37	31	40	42	42	42	59

(d) DepProbe (UAS)

Model	BAP	Dep	Dir
LAS=L	88 ±6.4	60 ±7.8	—
LAS¬L	35 ±15.7	22 ±9.9	—
UAS=L	91 ±5.0	67 ±6.7	70 ±7.8
UAS¬L	52 ±14.5	38 ±8.8	36 ±10.4

(e) Mean in-language (=L) and transfer (¬L) UAS/LAS (± stddev).



	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	63	25	22	28	44	19	36	18	23	33	30	24	18
EN	46	76	38	53	54	37	63	30	36	61	64	41	38
EU	31	28	65	41	35	34	35	28	36	36	37	39	26
FI	36	42	45	72	46	37	48	29	37	59	58	44	33
HE	46	34	28	38	69	25	45	21	27	46	43	28	22
HI	27	31	40	39	33	79	34	35	39	39	38	42	29
IT	49	54	40	52	59	31	79	27	35	60	62	39	31
JA	20	28	30	30	25	40	29	71	41	29	29	43	34
KO	25	25	30	29	29	27	31	26	65	28	30	39	20
RU	48	52	43	57	58	36	60	30	39	78	59	45	40
SV	39	43	31	50	45	29	50	25	30	50	73	33	29
TR	24	25	31	34	30	27	30	25	37	31	30	57	22
ZH	28	34	29	37	32	31	34	34	31	38	38	31	60

(f) DirProbe (UAS)

Figure 4.3: **In-language and Cross-lingual Transfer Performance** for 13 target treebanks (**train** → test) in UAS for BAP (fully tuned parser), DepProbe, DirProbe and LAS for BAP, DepProbe (DirProbe is unlabeled).

MODEL	LAS		UAS	
	ρ	τ_w	ρ	τ_w
L2V	.86	.72	.80	.70
DIRPROBE	—	—	.91	.81
DEPPROBE	**.97**	**.88**	.94	.85

Table 4.1: **Transfer Correlation with BAP.** Pearson ρ and weighted Kendall's τ_w for BAP's LAS and UAS with respect to DIRPROBE's UAS, DEPPROBE's UAS and LAS as well as lang2vec cosine similarity (L2V).

Setup Comparing UAS and LAS of BAP with respective scores of DEPPROBE and DIRPROBE, we compute the Pearson correlation coefficient ρ and the weighted Kendall's τ_w (Vigna, 2015). The latter can be interpreted as corresponding to a correlation in $[-1, 1]$, and that given a probe ranking one source treebank over another, the probability of this higher rank corresponding to higher performance in the full parser is $\frac{\tau_w+1}{2}$. All reported correlations are significant at $p < 0.001$. Similarly, differences between correlation coefficients are also significant at $p < 0.001$ as measured using a standard Z-test. In addition to the probes, we also compare against a method commonly employed by practitioners by using the cosine similarity of typological features from the URIEL database as represented in lang2vec (Littell et al., 2017; L2V) between our 13 targets (details in Appendix 4.7.1).

Results Table 4.1 shows that the L2V baseline correlates with final parser performance, but that actual dependency parses yield significantly higher correlation and predictive power. For UAS, we find that despite having similar attachment scores, DEPPROBE performance correlates higher with BAP than that of DIRPROBE, both with respect to predicting the ability to parse any particular language as well as ranking the best source to transfer from. Using the labeled parse trees of DEPPROBE results in almost perfect correlation with BAP's LAS at $\rho = .97$ as well as a τ_w of .88, highlighting the importance of modeling the full task and including dependency relation information. Using Kendall's τ_w with respect to LAS, we can estimate that selecting the

MODEL	LAS		UAS	
	ρ	τ_w	ρ	τ_w
SSA-STRUCT	.68	.42	.60	.43
SSA-DEPTH	.62	.34	.53	.35
SSA-REL	**.73**	**.55**	.65	.53

Table 4.2: **SSA Correlation with BAP.** Pearson ρ and weighted Kendall's τ_w for BAP's LAS and UAS with respect to subspace angles between structural (STRUCT), depth (DEPTH) and relation probes (REL).

highest performing source treebank from DEPPROBE to train the full parser will be the best choice 94% of the time for any treebank pair.

4.5 Analysis

4.5.1 Tree Depth versus Relations

Why does DEPPROBE predict transfer performance more accurately than DIRPROBE despite its simpler architecture? As each probe consists only of two matrices optimized to extract tree structural, depth or relational information, we can directly compare the similarity of all task-relevant parameters across languages against the full BAP's cross-lingual performance.

In order to measure the similarity of probe matrices from different languages, we use mean subspace angles (Knyazev and Argentati, 2002; SSA), similarly to prior probing work (Chi et al., 2020). Intuitively, SSA quantifies the energy required to transform one matrix to another by converting the singular values of the transformation into angles between 0° and 90°. SSAs are computed for the structural probe (SSA-STRUCT) which is equivalent in both methods, DIRPROBE's depth probe (SSA-DEPTH) and DEPPROBE's relational probe (SSA-REL). We use Pearson ρ and the weighted Kendall's τ_w to measure the correlation between cross-lingual probe SSAs and BAP performance. This allows us to investigate which type of information is most important for final parsing performance.

From Table 4.2, we can observe that SSAs between probes of dif-

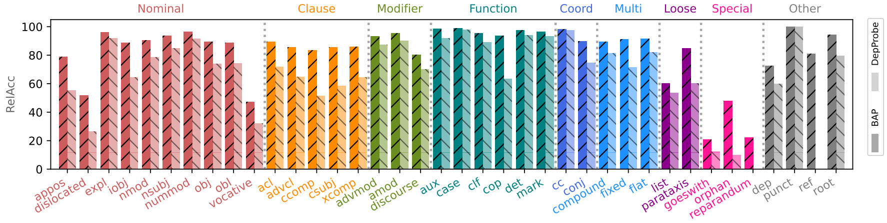


Figure 4.4: **Relation Accuracy of BAP and DepProbe** compared for all 13 in-language targets, grouped according to the Universal Dependencies taxonomy (de Marneffe et al., 2014).

ferent languages correlate less with transfer performance than UAS or LAS (Table 4.1), underlining the importance of extracting full parses. Among the different types of dependency information, we observe that SSAs between the *relational* probes used by DepProbe correlate highest with final performance at .73 for LAS and .65 for UAS. Structural probing correlates significantly both with BAP's LAS and UAS at .68 and .60 respectively, but to a lesser degree. Probes for tree depth have the lowest correlation at .62 for LAS and .53 for UAS. Despite tree depth being a distinctive syntactic feature for language pairs such as the agglutinative Turkish and the more function word-based English, depth is either not as relevant for BAP or may be represented less consistently in embeddings across languages, leading to lower correlation between SSAs and final performance.

4.5.2 Full Parser versus Probe

In the following analysis we investigate performance differences between the full BAP and DepProbe across all 13 targets in order to identify finer-grained limitations of the linear approach and also which kinds of dependencies benefit from full parameter tuning and non-linear decoding.

Edge Length Figure 4.5 shows offsets between gold and predicted head positions. The majority of heads are predicted correctly with a ratio of 92.1% for BAP and 69.7% for DepProbe. Both methods are less accurate in predicting long-distance edges with length 150–250,

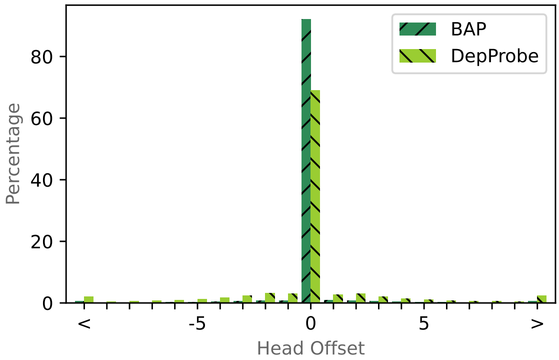


Figure 4.5: **Ratio of Offsets between Gold and Predicted Heads** for BAP and DepProbe (i.e., 0 is correct) across all 13 targets.

resulting in offsets of ca. 100 (aggregated into < and > in Figure 4.5). Most likely, this is due to these edges' overall sparsity in the data (only 6.7% of edges cover a distance of more than 10 tokens) as well as their higher overall subjective difficulty. Nonetheless, BAP is able to capture such dependencies more accurately as shown by its lower error rates for long edges compared to those of DepProbe.

In addition to very distant head nodes, BAP also seems to recover more of the nuanced edges in the $[-5,5]$ interval. This range is particularly impactful for downstream performance as the edges in our target treebanks have a median length of 2 (mean length 3.62 with $\sigma = 5.70$). The structural probing loss (Equation 4.2) and the simple linear parametrization of the probe are able to capture a large number of these edges as evidenced by overall low error rates, but lack the necessary expressivity in order to accurately capture all cases.

Relations Looking at RelAcc for each category in the UD taxonomy (de Marneffe et al., 2014) in Figure 4.4 allows us to identify where higher parametrization and more complex decoding are required for high parsing performance. While we again observe that performance on all relations is higher for BAP than for DepProbe, a large subset of the relations is characterized by comparable or equivalent performance. These include simple punctuation (`punct`), but also the majority of function word relations such as `aux`, `case`, `clf`, `det` and `mark` as well as coordination (e.g. `cc`, `conj`). We attribute the high performance of DepProbe on these relations to the fact that the words used to express them typically stem from closed classes and consequently similar embeddings: e.g., determiners "the/a/an" (EN), case markers "di/da" (IT).

Interestingly, some relations expressed through open class words are also captured by the linear probe. These include the modifiers `advmod`, `amod` and `discourse` as well as some nominal relations such as `expl`, `nmod`, `nsubj` and `nummod`. As prior work has identified PoS information in untuned embeddings (Tenney et al., 2019a), the modifiers are likely benefiting from the same embedding features. The fact that DepProbe nonetheless identifies syntax-specific relations such as `nsubj`, and to a lesser degree `obj` and `obl`, indicates the presence of context-dependent syntactic information in addition to PoS.

The larger the set of possible words for a relation, the more difficult it is to capture with the probe. The functional `cop` (copula) relation provides an informative example: In English (and related languages), it is almost exclusively assigned to the verb "be" resulting in 85% RelAcc, while in non-European languages such as Japanese it can be ascribed to a larger set which often overlaps with other relations (e.g. `aux`) resulting in 65% RelAcc. BAP adapts to each language by tuning all parameters while DepProbe, using fixed embeddings, reaches competitive scores on European languages, but performs worse in non-European settings (details in Appendix 4.7.2).

Besides capturing larger variation in surface forms, BAP also appears to benefit from higher expressivity when labeling clausal relations such as `ccomp`, `csubj`. These relations are often characterized not only by surface form variation, but also by PoS variation of head/child words and overlap with other relation types (e.g. clausal subjects stem from verbs or adjectives), making them difficult to distinguish in untuned embeddings. Simultaneously, they often span longer edges compared to determiners or other function words.

Another relation of particular importance is `root` as it determines the direction of all edges predicted by DepProbe. An analysis of the 14% RelAcc difference to BAP reveals that both methods most frequently confuse `root` with relations that fit the word's PoS, e.g. `NOUN` roots with `nsubj` or `nmod`. For the majority PoS `VERB` (70% of all `root`), we further observe that DEPPROBE predicts twice as many `xcomp` and `parataxis` confusions compared to BAP, likely attributable to their `root`-similar function in subclauses. Since their distinction hinges on context, the full parser, which also tunes the contextual encoder, is better equipped to differentiate between them.

The last category in which BAP outperforms DEPPROBE includes rare, treebank-specific relations such as `reparandum` (reference from a corrected word to an erroneous one). Again, the larger number of tunable parameters in addition to the non-linear decoding procedure of the full parser enable it to capture more edge cases while DEPPROBE's linear approach can only approximate a local optimum for any relations which are represented non-linearly.

Efficiency When using a probe for performance prediction, it is important to consider its computational efficiency over the full parser's fine-tuning procedure. In terms of tunable parameters, DEPPROBE has 36% fewer parameters than DIRPROBE and three orders of magnitude fewer parameters than BAP. In practice, this translates to training times in the order of minutes instead of hours.

Despite its simple $\mathcal{O}(n^2)$ decoding procedure compared to dynamic programming-based graph-decoding algorithms ($\mathcal{O}(n^3)$), DEPPROBE is able to extract full dependency trees which correlate highly with downstream performance while maintaining high efficiency (Section 4.4.4).

4.6 Conclusion

With DEPPROBE, we have introduced a novel probing procedure to extract fully labeled and directed dependency trees from untuned, contextualized embeddings. Compared to prior approaches which extract structures lacking labels, edge directionality or both, our method retains a simple linear parametrization which is in fact more lightweight and does not require complex decoders (Section 4.3).

To the best of our knowledge, this is the first linear probe which can be used to estimate LAS from untuned embeddings. Using this property, we evaluated the predictive power of DepProbe on cross-lingual parsing with respect to the transfer performance of a fully fine-tuned biaffine attention parser. Across the considered 169 language pairs, DepProbe is surprisingly effective: Its LAS correlates significantly ($p < 0.001$) and most highly compared with unlabeled probes or competitive language feature baselines, choosing the best source treebank in 94% of all cases (Section 4.4).

Leveraging the linearity of the probe to analyze structural and relational subspaces in mBERT embeddings, we find that dependency *relation* information is particularly important for parsing performance and cross-lingual transferability, compared to both tree depth and structure. DepProbe, which models structure and relations, is able to recover many functional and syntactic relations with competitive accuracy to the full BAP (Section 4.5).

Finally, the substantially higher efficiency of DepProbe with respect to time and compute make it suitable for accurate parsing performance prediction. As contemporary performance prediction methods lack formulations for graphical tasks and handcrafted features such as lang2vec are not available in all transfer settings (e.g. document domains, MLM encoder choice), we see linear approaches such as DepProbe as a valuable alternative.

4.7 Appendix

4.7.1 Experimental Setup

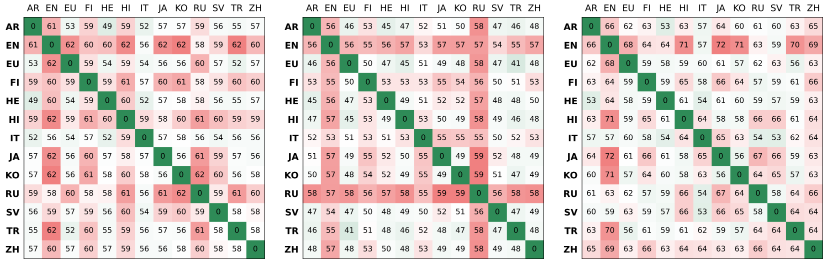
Target Treebanks Table 4.3 lists the 13 target treebanks based on the set by Kulmizev et al. (2019): AR-PADT (Hajič et al., 2009), EN-EWT (Silveira et al., 2014), EU-BDT (Aranzabe et al., 2015), FI-TDT (Pyysalo et al., 2015), HE-HTB (McDonald et al., 2013), HI-HDTB (Palmer et al., 2009), IT-ISDT (Bosco et al., 2014), JA-GSD (Asahara et al., 2018), KO-GSD (Chun et al., 2018), RU-SynTagRus (Droganova et al., 2018), SV-Talbanken (McDonald et al., 2013), TR-IMST (Sulubacak et al., 2016), ZH-GSD (Shen et al., 2016a). In our experiments, we use these treebanks as provided in Universal Dependencies version 2.8 (Zeman et al., 2021). Each method (BAP, DepProbe, DirProbe) is trained on each target's respective training split and evaluated on each test split both in

TARGET	LANG	FAMILY	SIZE
AR-PADT	Arabic	Afro-Asiatic	7.6k
EN-EWT	English	Indo-European	16.6k
EU-BDT	Basque	Basque	9.0k
FI-TDT	Finnish	Uralic	15.1k
HE-HTB	Hebrew	Afro-Asiatic	6.2k
HI-HDTB	Hindi	Indo-European	16.6k
IT-ISDT	Italian	Indo-European	14.1k
JA-GSD	Japanese	Japanese	8.1k
KO-GSD	Korean	Korean	6.3k
RU-SynTagRus	Russian	Indo-European	61.9k
SV-Talbanken	Swedish	Indo-European	6.0k
TR-IMST	Turkish	Turkic	5.6k
ZH-GSD	Chinese	Sino-Tibetan	5.0k

Table 4.3: **Target Treebanks** based on Kulmizev et al. (2019) with language family (FAMILY) and total number of sentences (SIZE).

the in-language and cross-lingual setting without further fine-tuning. For the layer-hyperparameter of DEPPROBE, we use the development split of EN-EWT as in prior probing work (Hewitt and Manning, 2019).

Implementation BAP (Dozat and Manning, 2017) uses the implementation in the MaChAmp toolkit v0.2 (van der Goot et al., 2021b) with the default training schedule and hyperparameters. DIRPROBE (Kulmizev et al., 2020) is reimplemented based on the authors' algorithm description and uses their reported hyperparameters. Both it and DEPPROBE (this work) are implemented in PyTorch v1.9.0 (Paszke et al., 2019) and use mBERT (`bert-base-multilingual-cased`) from the Transformers library v4.8.2 (Wolf et al., 2020). Following prior probing work, each token which is split by mBERT into multiple subwords is mean-pooled (Hewitt and Manning, 2019). For lang2vec (Littell et al., 2017), we use its `syntax_knn`, `phonology_knn` and `inventory_knn` features from v1.1.2. For our analyses (Section 4.5), we use numpy v1.21.0 (Harris et al., 2020), SciPy v1.7.0 (Virtanen et al., 2020) and Matplotlib v3.4.3 (Hunter, 2007).



(a) SSA-Struct

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	0	61	53	59	49	59	52	57	57	59	56	55	57
EN	61	0	62	60	60	62	56	62	62	58	59	62	60
EU	53	62	0	59	54	59	54	56	56	60	57	52	57
FI	59	60	59	0	59	61	57	60	61	58	59	60	60
HE	49	60	54	59	0	60	52	57	58	58	56	55	57
HI	59	62	59	61	60	0	59	58	60	61	60	59	59
IT	52	56	54	57	52	59	0	57	58	56	54	56	56
JA	57	62	56	60	57	58	57	0	56	61	59	57	56
KO	57	62	56	61	58	60	58	56	0	62	60	56	58
RU	59	58	60	58	58	61	56	61	62	0	59	61	60
SV	56	59	57	59	56	60	54	59	60	59	0	58	58
TR	55	62	52	60	55	59	56	57	56	61	58	0	58
ZH	57	60	57	60	57	59	56	56	58	60	58	58	0

(b) SSA-Depth

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	0	56	46	53	45	47	52	51	50	58	47	46	48
EN	56	0	56	55	56	57	53	57	57	57	54	55	57
EU	46	56	0	50	47	45	51	49	48	58	47	41	48
FI	53	55	50	0	53	53	53	55	54	56	50	51	53
HE	45	56	47	53	0	49	51	52	52	57	48	48	50
HI	47	57	45	53	49	0	53	50	49	58	49	46	48
IT	52	53	51	53	51	53	0	55	55	55	50	52	53
JA	51	57	49	55	52	50	55	0	49	59	52	48	49
KO	50	57	48	54	52	49	55	49	0	59	51	47	49
RU	58	57	58	56	57	58	55	59	59	0	56	58	58
SV	47	54	47	50	48	49	50	52	51	56	0	47	49
TR	46	55	41	51	48	46	52	48	47	58	47	0	48
ZH	48	57	48	53	50	48	53	49	49	58	49	48	0

(c) SSA-Rel

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	0	66	62	63	53	63	57	64	60	61	60	63	65
EN	66	0	68	64	64	71	57	72	71	63	59	70	69
EU	62	68	0	59	58	59	60	61	57	62	63	56	63
FI	63	64	59	0	59	65	58	66	64	57	59	61	66
HE	53	64	58	59	0	61	54	61	60	59	57	59	63
HI	63	71	59	65	61	0	64	58	58	66	66	61	64
IT	57	57	60	58	54	64	0	65	63	54	53	62	64
JA	64	72	61	66	61	58	65	0	56	67	66	59	63
KO	60	71	57	64	60	58	63	56	0	64	65	57	63
RU	61	63	62	57	59	66	54	67	64	0	58	64	66
SV	60	59	63	59	57	66	53	66	65	58	0	64	64
TR	63	70	56	61	59	61	62	59	57	64	64	0	64
ZH	65	69	63	66	63	64	64	63	63	66	64	64	0

Figure 4.6: **SSA of Probe Transformations** in degrees across 13 target treebanks for the structural (SSA-Struct), depth (SSA-Depth) and relational probes (SSA-Rel).

Training Details Each model is trained on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. Mean training time for BAP is ca. 2 h (± 30 min). DirProbe requires around 20 min (± 5 min). DepProbe can be trained the fastest in around 15 min (± 5 min) with the embedding forward operation consuming most of the time. The models use batches of size 64 and both probes have an early stopping patience of 3 (max. 30 epochs) on each target's dev data. All models are initialized thrice using the random seeds 41, 42 and 43.

Reproducibility In order to ensure reproducibility for future work, we release the code for our methods and reimplementations in addition to token-level predictions (e.g. for significance testing) at https://personads.me/x/acl-2022-code.

4.7.2 Additional Results

Subspace Angles (SSA) are used in Section 4.5.1 in order to identify which types of dependency information are most relevant to final parsing performance. Figure 4.6 lists all cross-lingual SSAs for the structural (Figure 4.6a), depth (Figure 4.6b) and relational probes (Figure 4.6c). SSA values are converted from radians to degrees ∈ [0, 90] for improved readability. Correlation in Table 4.2 is calculated based on negative SSA (Chi et al., 2020).

Relation Accuracy (RelAcc) is used in Section 4.5.2 to analyze dependency relations which benefit from the full parametrization of BAP compared to the linear DepProbe. Figures 4.7–4.19 show RelAcc per language in addition to the aggregated scores in Figure 4.4. As noted in Section 4.5.2, some relations such as `cop` differ substantially across languages with respect to their realization (e.g. surface form variation). Furthermore, the set of relations represented in each target treebank may differ, especially for specializied categories.

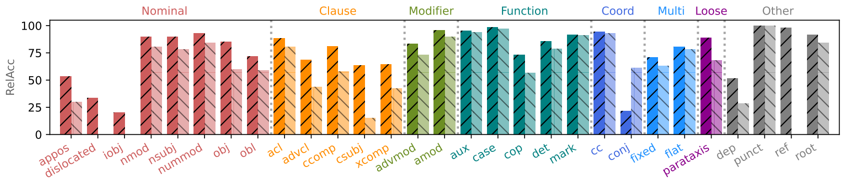


Figure 4.7: **RelAcc of BAP and DepProbe on AR-PADT (Test)** grouped according to UD taxonomy.

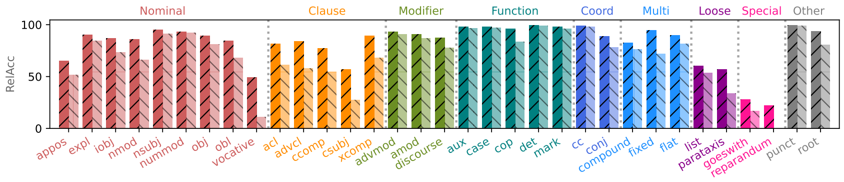


Figure 4.8: **RelAcc of BAP and DepProbe on EN-EWT (Test)** grouped according to UD taxonomy.

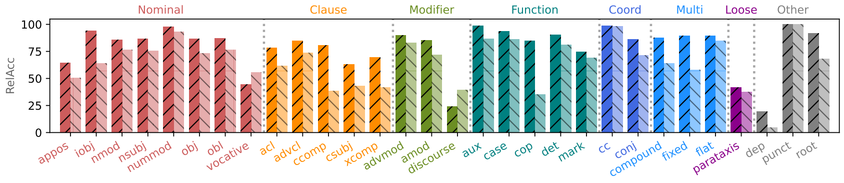


Figure 4.9: **RelAcc of BAP and DepProbe on EU-BDT (Test)** grouped according to UD taxonomy.

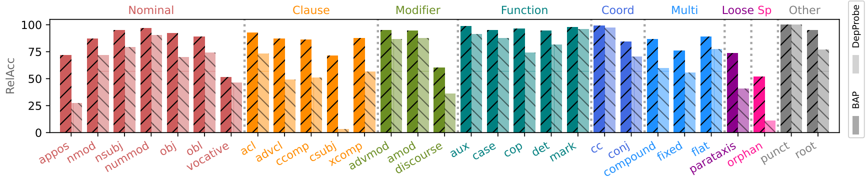


Figure 4.10: **RelAcc of BAP and DepProbe on FI-TDT (Test)** grouped according to UD taxonomy.

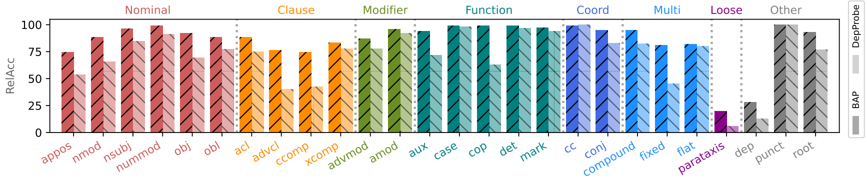


Figure 4.11: **RelAcc of BAP and DepProbe on HE-HTB (Test)** grouped according to UD taxonomy.

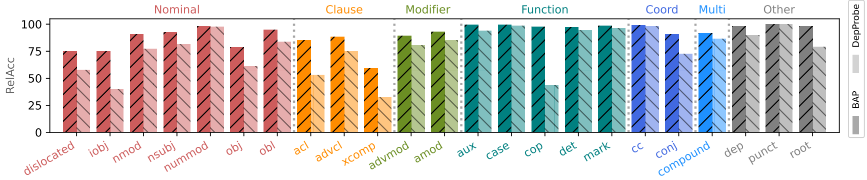


Figure 4.12: **RelAcc of BAP and DepProbe on HI-HDTB (Test)** grouped according to UD taxonomy.

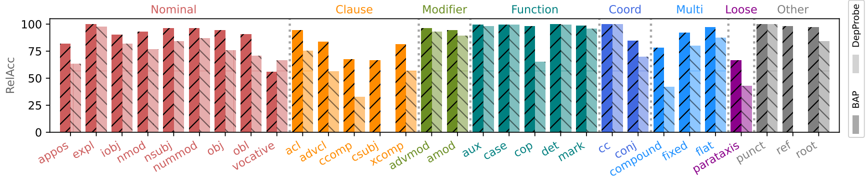


Figure 4.13: **RelAcc of BAP and DepProbe on IT-ISDT (Test)** grouped according to UD taxonomy.

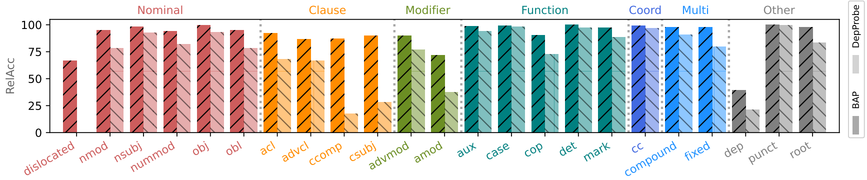


Figure 4.14: **RelAcc of BAP and DEPPROBE on JA-GSD (Test)** grouped according to UD taxonomy.

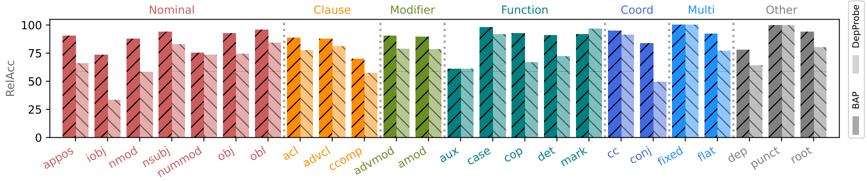


Figure 4.15: **RelAcc of BAP and DEPPROBE on KO-GSD (Test)** grouped according to UD taxonomy.

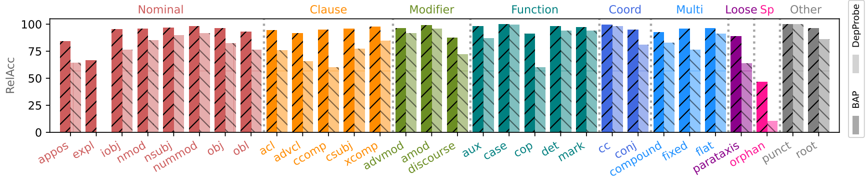


Figure 4.16: **RelAcc of BAP and DEPPROBE on RU-SynTagRus (Test)** grouped according to UD taxonomy.

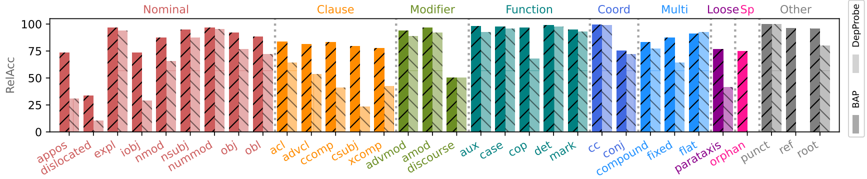


Figure 4.17: **RelAcc of BAP and DEPPROBE on SV-Talbanken (Test)** grouped according to UD taxonomy.

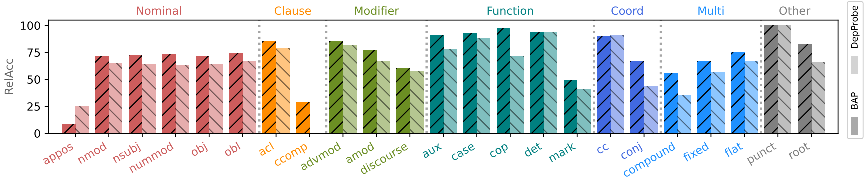


Figure 4.18: **RelAcc of BAP and DepProbe on TR-IMST (Test)** grouped according to UD taxonomy.

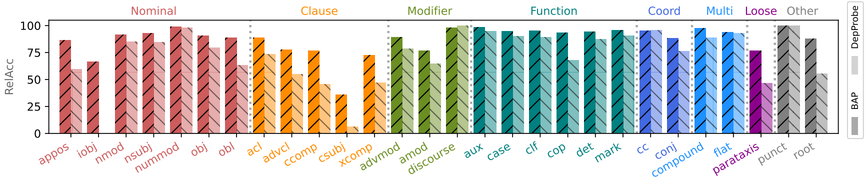


Figure 4.19: **RelAcc of BAP and DepProbe on ZH-GSD (Test)** grouped according to UD taxonomy.

Sort by Structure: LM Ranking as Dependency Probing 5

The work presented in this chapter is based on: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022b. Sort by structure: Language model ranking as dependency probing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1307, Seattle, United States. Association for Computational Linguistics.

Abstract

Making an informed choice of pre-trained language model (LM) is critical for performance, yet environmentally costly, and as such widely underexplored. The field of Computer Vision has begun to tackle encoder ranking, with promising forays into Natural Language Processing, however they lack coverage of linguistic tasks such as structured prediction. We propose *probing to rank LMs*, specifically for parsing dependencies in a given language, by measuring the degree to which labeled trees are recoverable from an LM's contextualized embeddings. Across 46 typologically and architecturally diverse LM-language pairs, our probing approach predicts the best LM choice 79% of the time using orders of magnitude less compute than training a full parser. Within this study, we identify and analyze one recently proposed decoupled LM—RemBERT—and find it strikingly contains less inherent dependency information, but often yields the best parser after full fine-tuning. Without this outlier our approach identifies the best LM in 89% of cases.

5.1 Introduction

With the advent of massively pre-trained language models (LMs) in Natural Language Processing (NLP), it has become crucial for practitioners to choose the best LM encoder for their given task early on, regardless of the rest of their proposed model architecture. The greatest variation of LMs lies in the language or domain-specificity of the unlabelled data used during pre-training (with architectures often staying identical).

Typically, better expressivity is expected from language/domain-specific LMs (Gururangan et al., 2020; Dai et al., 2020) while open-domain settings necessitate high-capacity models with access to as much pre-training data as possible. This tradeoff is difficult to navigate, and given that multiple specialized LMs (or none at all) are available, practitioners often resort to an ad-hoc choice. In absence of immediate performance indicators, the most accurate choice could be made by training the full model using each LM candidate, however this is often infeasible and wasteful Strubell et al. (2019).

Recently, the field of Computer Vision (CV) has attempted to tackle this problem by quantifying useful information in pre-trained image encoders as measured directly on labeled target data without fine-tuning (Nguyen et al., 2020; You et al., 2021). While first forays for applying these methods to NLP are promising, some linguistic tasks differ substantially: Structured prediction, such as parsing syntactic dependencies, is a fundamental NLP task not covered by prior encoder ranking methods due to its graphical output. Simultaneously, performance prediction in NLP has so far been studied as a function of dataset and model characteristics (Xia et al., 2020; Ye et al., 2021) and has yet to examine how to rank large pools of pre-trained LMs.

Given the closely related field of probing, in which lightweight models quantify task-specific information in pre-trained LMs, we recast its objective in the context of performance prediction and ask: *How predictive is lightweight probing at choosing the best performing LM for dependency parsing?* To answer this question, we contribute:

- An efficient encoder ranking method for structured prediction using dependency probing (Müller-Eberstein et al., 2022a; DepProbe) to quantify latent syntax (Section 5.2).
- Experiments across 46 typologically and architecturally diverse LM + target language combinations (Section 5.3).[1]
- An in-depth analysis of the surprisingly low inherent dependency information in RemBERT (Chung et al., 2021) compared to its high fine-tuned performance (Section 5.4).

5.2 Methodology

Probing pre-trained LMs is highly related to encoder ranking in CV where the ease of recoverability of class-differentiating information is key (Nguyen et al., 2020; You et al., 2021). This approach is more immediate than existing NLP performance prediction methods which rely on featurized representations of source and target data without actively ranking encoders (Xia et al., 2020; Ye et al., 2021). As most

[1] Code at https://personads.me/x/naacl-2022-code.

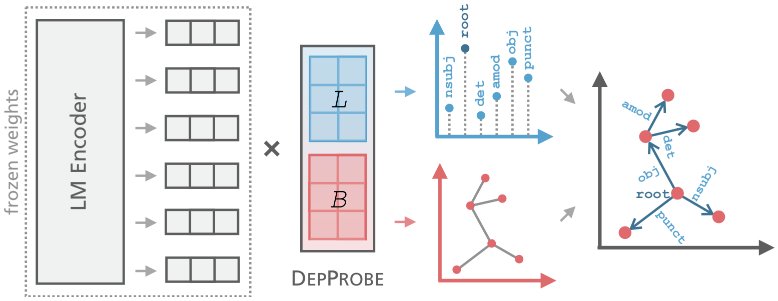


Figure 5.1: **Visualization of DepProbe.** Relational and structural subspaces L and B are combined to extract labeled, directed trees from embeddings.

experiments in NLP are conducted using a limited set of LMs—often a single model—without strong prior motivations, *we see LM ranking as a critical task on its own.*

While probes for LMs come in many forms, they are generally characterized as lightweight, minimal architectures intended to solve a particular task (Maudslay et al., 2020). While non-linear models such as small multi-layer perceptrons are often used (Tenney et al., 2019b), there have been criticisms given that their performance highly depends on the complexity of their architecture (Hewitt and Liang, 2019; Voita and Titov, 2020). As such, we rely on linear probes alone, which have the benefit of being extremely lightweight, closely resembling existing performance prediction methods (You et al., 2021), and allow for statements about linear subspaces contained in LM latent spaces.

DepProbe (Müller-Eberstein et al., 2022a; visualized in Figure 5.1) is a linear formulation for extracting fully labeled dependency trees based on the structural probe by Hewitt and Manning (2019). Given contextualized embeddings of dimensionality d, a linear transformation $B \in \mathbb{R}^{b \times d}$ with $b \ll d$ (typically $b = 128$) maps them into a subspace in which the Euclidean distance between embeddings corresponds to the number of edges between the respective words in the gold dependency graph.

In our formulation, we supplement a linear transformation $L \in \mathbb{R}^{l \times d}$ (with l = number of dependency relations) which maps each embedding to a subspace in which the magnitude of each dimension corresponds to the likelihood of a word and its head being governed by a certain relation.

By computing the minimum spanning tree in B and then finding the word with the highest root likelihood in L, we can determine the directionality of all edges as pointing away from the root. All remaining edges are labeled according to the most likely non-root class in L, resulting in a fully directed and labeled dependency tree.

Note that this approach differs substantially from prior approaches which yield undirected and/or unlabeled trees (Hewitt and Manning, 2019; Kulmizev et al., 2020) or use pre-computed edges and non-linear classifiers (Tenney et al., 2019b). DEPPROBE efficiently computes the full target metric (i.e. labeled attachment scores) instead of approximate alternatives (e.g. undirected, unlabeled attachment scores or tree depth correlation).

5.3 Experiments

Setup We investigate the ability of DEPPROBE to select the best performing LM for dependency parsing across nine linguistically diverse treebanks from Universal Dependencies (Zeman et al., 2021; UD) which were previously chosen by Smith et al. (2018b) to reflect diverse writing systems and morphological complexity (see Appendix 5.6.1).

For each target language, we employ three multilingual LMs—mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), RemBERT (Chung et al., 2021)—as well as 1–3 language-specific LMs retrieved by popularity from HuggingFace's Model Hub (Wolf et al., 2020), resulting in a total of 46 LM-target pair setups (see Appendix 5.6.3).

For each combination, we train a DEPPROBE to compute labeled attachment scores (LAS), hypothesizing that LMs from which trees are most accurately recoverable also perform better in a fully tuned parser. To evaluate the true downstream performance of a fully-tuned model, we further train a deep biaffine attention parser (BAP; Dozat and Manning, 2017) on each LM-target combination. Compared to full fine-tuning, DEPPROBE only optimizes the matrices B and L, resulting in the extraction of labeled trees with as few as 190k instead of

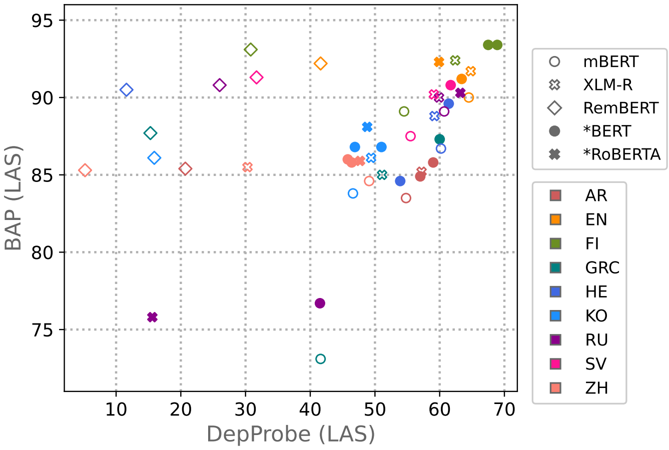


Figure 5.2: **LAS of DEPPROBE in relation to full BAP** across nine language targets (dev) using language-specific and multilingual LM encoders of different architecture types (exact scores in Appendix 5.6.3).

583M trainable parameters for the largest RemBERT model (details in Appendix 5.6.2).

We measure the predictive power of probing for fully fine-tuned model performance using the Pearson correlation coefficient ρ as well as the weighted Kendall's τ_w (Vigna, 2015). The latter metric corresponds to a correlation coefficient in $[-1, 1]$ and simultaneously defines the probability of choosing the better LM given a pair as $\frac{\tau_w+1}{2}$, allowing us to quantify the overall quality of a ranking.

Results Comparing the LAS of DEPPROBE's lightweight predictions against full BAP fine-tuning in Figure 5.2, we see a clear correlation as the probe correctly predicts the difficulty of parsing languages relative to each other and also ranks models within languages closely according to their final performance. With a τ_w of .58 between scores ($p < 0.001$), this works out to DEPPROBE selecting the better performing final model given any two models 79% of the time. Additionally, LAS is slightly more predictive of final performance than unlabeled, undirected attachment scores (UUAS) with $\tau_w = .57$ to which prior probing approaches are restricted (see Appendix 5.6.3).

Given a modest ρ of .32 ($p < 0.05$), we surprisingly also observe a single strong outlier to this pattern, namely the multilingual RemBERT (Chung et al., 2021) decoupled LM architecture. While DEPPROBE consistently ranks it low as it cannot extract dependency parse trees as accurately as from the BERT and RoBERTa-based architectures, RemBERT actually performs best on four out of the nine targets when fully fine-tuned in BAP. Excluding monolingual LMs, it further outperforms the other multilingual LMs in seven out of nine cases. As it is a more recent and distinctive architecture with many differences to the most commonly-used contemporary LMs, we analyze potential reasons for this discrepancy in Section 5.4.

Excluding RemBERT as an outlier, we find substantially higher correlation among all other models: $\rho = .78$ and $\tau_w = .78$ ($p < 0.001$). This means that among these models, fully fine-tuning the LM for which DEPPROBE extracts the highest scores, yields the better final performance 89% of the time.

In practice, learning DEPPROBE's linear transformations while keeping the LM frozen is multiple orders of magnitude more efficient than fully training a complex parser plus the LM's parameters. As such, linear probing offers a viable method for selecting the best encoder in absence of qualitative heuristics or intuitions. This predictive performance is furthermore achievable in minutes compared to hours and at a far lower energy budget (see Appendices 5.6.2 and 5.6.3).

5.4 Probing Decoupled LMs

Considering DEPPROBE's high predictive performance across LMs with varying architecture types, languages/domains and pre-training procedures, we next investigate its limitations: Specifically, which differences in RemBERT (Chung et al., 2021) lead to it being measured as an outlier with seemingly low amounts of latent dependency information despite reaching some of the highest scores after full fine-tuning. The architecture has 32 layers and embeddings with $d = 1152$, compared to most models' 12 layers and $d = 768$. It accommodates these size and depth increases within a manageable parameter envelope by using smaller input embeddings with $d_{\text{in}} = 256$. While choosing different d for the input and output embeddings is not possible in

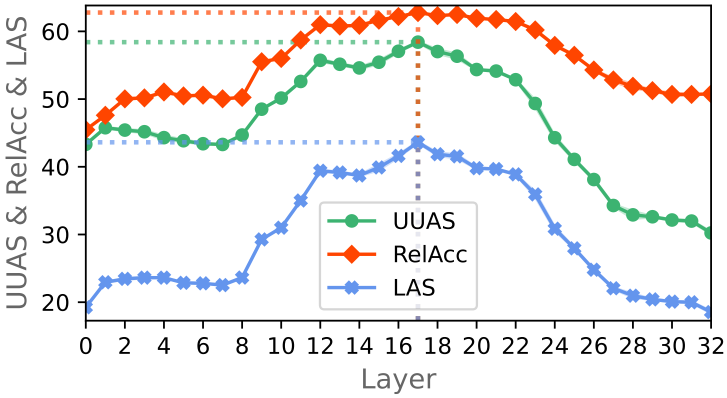


Figure 5.3: **Dependency Information per RemBERT Layer** via DEPPROBE's structural, relational and parsing accuracy (UUAS, RelAcc, LAS) on EN-EWT (dev).

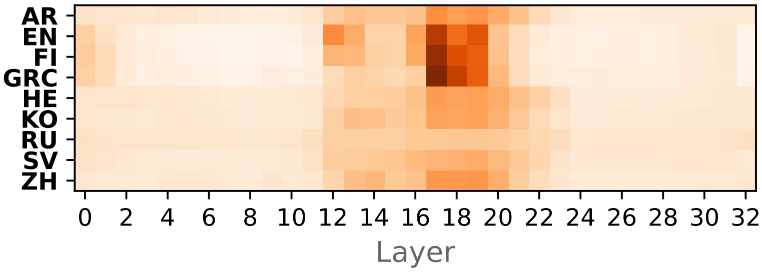


Figure 5.4: **Per-language α of RemBERT Layers** for DEPPROBE across all layer weights (dark > light).

most prior models due to both embedding matrices being coupled, RemBERT decouples them, leading to a larger parameter budget and less overfitting on the masked language modeling pre-training task (Chung et al., 2021).

Layer-wise Probing Prior probing studies have found dependency information to be concentrated around the middle layers of an LM (Hewitt and Manning, 2019; Tenney et al., 2019b; Fayyaz et al., 2021). Using EN-EWT (Silveira et al., 2014), we evaluate whether this holds for RemBERT's new architecture. Figure 5.3 confirms that both dependency structural and relational information are most prominent

MODEL	AR	EN	FI	GRC	HE	KO	RU	SV	ZH
mBERT	**65**	**74**	65	46	**69**	**58**	**68**	**65**	**58**
	±.08	±.09	±.35	±.14	±.23	±.18	±.31	±.12	±.17
XLM-R	60	70	**66**	53	60	49	57	51	51
	±.14	±.08	±.18	±.19	±.20	±.08	±.34	±.24	±.53
RemBERT	58	56	52	**54**	52	46	49	43	39
	±.12	±.22	±.15	±.18	±.05	±.14	±.04	±.08	±.24

Table 5.1: **LAS of BAP Trained on Frozen LMs.** A biaffine attention parsing head is trained on top of frozen mBERT, XLM-R and RemBERT for each of the nine target languages (± standard deviation).

around layer 17 of 32 as indicated by UUAS and relation classification accuracy (RelAcc) respectively. Combining the structural and relational information in DEPPROBE similarly leads to a peak of the LAS at the same layer while decreasing with further distance from the center.

Across all target languages, we next investigate whether probing a sum over the embeddings of all layers weighted by $\boldsymbol{\alpha} \in \mathbb{R}^{32}$ can boost extraction performance in RemBERT. The heavier weighting of middle layers by $\boldsymbol{\alpha}$, visible in Figure 5.4, reaffirms a concentration of dependency information in the center. Contrasting probing work on prior models (Tenney et al., 2019b; Kulmizev et al., 2020), using all layers does not increase the retrievable dependencies, with LAS differences ±1 point. This further confirms that there is not a lack of dependency information in any specific layer, but that there is less within the encoder as a whole.

Frozen Parsing Our probing results show that linear subspaces in RemBERT contain less dependency information than prior LMs. However, DEPPROBE's parametrization is kept intentionally simple and may therefore not be capturing non-linearly represented information that is useful during later fine-tuning. To evaluate this hypothesis, we train a full biaffine attention parsing head, but keep the underlying LM encoder frozen. This allows us to quantify the performance gains which come from inherent dependency information versus later task-specific fine-tuning.

Table 5.1 confirms our findings from DEPPROBE and shows that despite RemBERT outperforming mBERT and XLM-R when fully fine-tuned, it has substantially lower LAS across almost all languages when no full model fine-tuning is applied. This leads us to conclude that there indeed is less inherent dependency information in the newer model and that most performance gains must be occurring during task-specific full fine-tuning.

Given that DEPPROBE extracts dependency structures reliably from LM architectures with different depths and embedding dimensionalities (e.g. RoBERTa$_{\text{large}}$ with 24 layers and $d = 1024$ versus RuBERT$_{\text{tiny}}$ with 3 layers and $d = 312$) as well as varying tokenization, optimization and pre-training data, the key difference in RemBERT appears to be embedding decoupling. The probe's linear formulation is not the limiting factor as the non-linear, biaffine attention head also produces less accurate parses when the LM's weights are frozen. Our analyses thus suggest that RemBERT's decoupled architecture contains less dependency information out-of-the-box, but follows prior patterns such as consolidating dependency information towards its middle layers and serving as strong initialization for parser training.

Lastly, RemBERT's larger number of tunable parameters compared to all other LM candidates may provide it further capacity, especially after full fine-tuning. As our probing methods are deliberately applied to the frozen representations of the encoder, it becomes especially important to consider the degree to which these embeddings may change after updating large parts of the model. Taking these limitations into account, the high correlations with respect to encoder ranking nonetheless enable a much more informed selection of LMs from a larger pool than was previously possible.

5.5 Conclusion

To guide practitioners in their choice of LM encoder for the structured prediction task of dependency parsing, we leveraged a lightweight, linear DEPPROBE to quantify the latent syntactic information via the *labeled* attachment score. Evaluating 46 pairs of multilingual/language-specific LMs and nine typologically diverse target treebanks, we found

DEPPROBE to not only be efficient in its predictions, with orders of magnitude fewer trainable parameters, but to also be accurate 79–89% of the time in predicting which LM will outperform another when used in a fully tuned parser. This allows for a substantially faster iteration over potential LM candidates, saving hours worth of compute in practice (Section 5.3).

Our experiments further revealed surprising insights on the newly proposed RemBERT architecture: While particularly effective for multilingual dependency parsing when fully fine-tuned, it contains substantially less latent dependency information relative to prior widely-used models such as mBERT and XLM-R. Among its architectural differences, we identified embedding decoupling to be the most likely contributor, while added model capacity during fine-tuning may also improve final performance. Our analyses showed that despite containing less dependency information overall, RemBERT follows prior findings such as structure and syntactic relations being consolidated towards the middle layers. Given these consistencies, performance differences between decoupled LMs may be predictable using probes, but in absence of similar multilingual LMs using decoupled embeddings this effect remains to be studied (Section 5.4).

Overall, the high efficiency and predictive power of ranking LM encoders via linear probing as well as the ease with which they can be analyzed—even when they encounter their limitations—offers immediate benefits to practitioners who have so far had to rely on their own intuitions when making a selection. This opens up avenues for future research by extending these methods to more tasks and LM architectures in order to enable better informed modeling decisions.

5.6 Appendix

5.6.1 Treebanks

Table 5.2 lists the nine target treebanks based on the set by Smith et al. (2018b): AR-PADT (Hajič et al., 2009), EN-EWT (Silveira et al., 2014), FI-TDT (Pyysalo et al., 2015), GRC-PROIEL (Eckhoff et al., 2018), HE-HTB (McDonald et al., 2013), KO-GSD (Chun et al., 2018), RU-GSD (McDonald et al., 2013), SV-Talbanken (McDonald et al., 2013), ZH-GSD (Shen

TARGET	LANG	FAMILY	SIZE
AR-PADT	Arabic	Afro-Asiatic	7.6k
EN-EWT	English	Indo-European	16.6k
FI-TDT	Finnish	Uralic	15.1k
GRC-PROIEL	Ancient Greek	Indo-European	17.1k
HE-HTB	Hebrew	Afro-Asiatic	6.2k
KO-GSD	Korean	Korean	6.3k
RU-GSD	Russian	Indo-European	5k
SV-Talbanken	Swedish	Indo-European	6.0k
ZH-GSD	Chinese	Sino-Tibetan	5.0k

Table 5.2: **Target Treebanks** based on Smith et al. (2018b) with language family (FAMILY) and total number of sentences (SIZE).

et al., 2016a). We use these treebanks as provided in Universal Dependencies v2.9 (Zeman et al., 2021). DEPPROBE and BAP are trained on each target's respective training split and are evaluated on the development split as this work aims to analyze general performance patterns instead of state-of-the-art performance.

5.6.2 Experiment Setup

DEPPROBE is implemented in PyTorch v1.9.0 (Paszke et al., 2019) and uses language models from the Transformers library v4.13.0 and the associated Model Hub (Wolf et al., 2020). Following the structural probe by Hewitt and Manning (2019), each token which is split by the LM encoder into multiple subwords is mean-pooled. Similarly, we follow the original hyperparameter settings and set the structural subspace dimensionality to $b = 128$ and use embeddings from the middle layer of each LM (Hewitt and Manning, 2019; Tenney et al., 2019b; Fayyaz et al., 2021). The structural loss is computed based on the absolute difference of the Euclidean distance between transformed word embeddings and the number of edges separating the words in the gold tree (see Hewitt and Manning, 2019 for details). The relational loss is computed using cross entropy between the logits and gold head-child relation. Optimization uses AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-3} which is reduced by a factor of

10 each time the loss plateaus. Early stopping is applied after three epochs without improvement and a maximum of 30 total epochs. With the only trainable parameters being the matrices B and L, the model's footprint ranges between 51k and 190k parameters.

BAP For the biaffine attention parser (Dozat and Manning, 2017) we use the implementation in the MaChAmp framework v0.3 (van der Goot et al., 2021b) with the default training schedule and hyperparameters. The number of trainable parameters depends on the LM encoder's size and ranges between 14M and 583M.

Analyses For our analyses in Sections 5.3 and 5.4 we further make use of numpy v1.21.0 (Harris et al., 2020), SciPy v1.7.0 (Virtanen et al., 2020) and Matplotlib v3.4.3 (Hunter, 2007).

Training Details Models are trained on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. BAP requires around 1 h (± 30 min). DepProbe can be trained in around 15 min (± 5 min) with the embedding forward operation being most computationally expensive. The models use batches of size 32 and are initialized using the random seeds 692, 710 and 932.

Reproducibility In order to ensure reproducibility and comparability with future work, we release our code and token-level predictions at https://personads.me/x/naacl-2022-code.

5.6.3 Detailed Results

Tables 5.3–5.11 list exact LAS and standard deviations for each experiment in Section 5.3's Figure 5.2 in addition to the HuggingFace Model Hub IDs of the LMs used in each of the 46 setups as well as their number of layers, embedding dimensionality d and total number of parameters. In addition, Figure 5.5 shows UUAS for all setups, equivalent to only probing structurally (Hewitt and Manning, 2019) for unlabeled, undirected dependency trees.

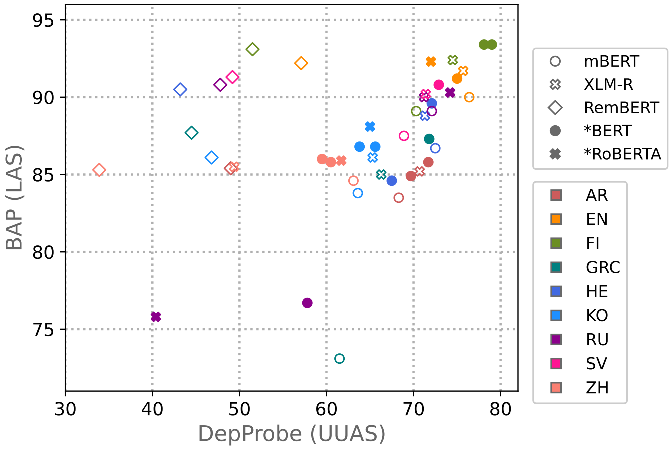


Figure 5.5: **UUAS of DEPPROBE in relation to BAP** across nine language targets (dev) using language-specific and multilingual LM encoders of different architecture types.

MODELS	SOURCE	LAYERS	EMB *d*	PARAMS	BAP	DEPPROBE
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	83.5±0.2	54.8±0.6
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	85.2±0.1	57.2±0.1
`google/rembert`	Chung et al. (2021)	32	1152	576M	85.4±0.2	20.7±0.1
`aubmindlab/bert-base-arabertv02`	Antoun et al. (2020)	12	768	135M	85.8±0.1	59.0±0.1
`asafaya/bert-base-arabic`	Safaya et al. (2020)	12	768	111M	84.9±0.1	57.0±0.2

Table 5.3: **LAS on AR-PADT (Dev)** using BAP and DEPPROBE with different LMs (± standard deviation).

MODELS	SOURCE	LAYERS	EMB *d*	PARAMS	BAP	DEPPROBE
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	90.0±0.1	64.5±0.3
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	91.7±0.2	64.8±0.1
`google/rembert`	Chung et al. (2021)	32	1152	576M	92.2±0.0	41.6±0.3
`bert-base-uncased`	Devlin et al. (2019)	12	768	109M	91.2±0.1	63.4±0.3
`roberta-large`	Liu et al. (2019b)	24	1024	355M	92.3±0.2	59.9±0.2

Table 5.4: **LAS on EN-EWT (Dev)** using BAP and DEPPROBE with different LMs (± standard deviation).

Models	Source	Layers	Emb d	Params	BAP	DepProbe
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	89.1±0.2	54.5±0.4
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	92.4±0.1	62.4±0.2
`google/rembert`	Chung et al. (2021)	32	1152	576M	93.1±0.1	30.8±0.1
`TurkuNLP/bert-base-finnish-uncased-v1`	Virtanen et al. (2019)	12	768	125M	93.4±0.1	68.9±0.3
`TurkuNLP/bert-base-finnish-cased-v1`	Virtanen et al. (2019)	12	768	125M	93.4±0.1	67.5±0.4

Table 5.5: **LAS on FI-TDT (Dev)** using BAP and DepProbe with different LMs (± standard deviation).

Models	Source	Layers	Emb d	Params	BAP	DepProbe
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	73.1±0.1	41.6±0.5
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	85.0±0.2	51.1±0.2
`google/rembert`	Chung et al. (2021)	32	1152	576M	87.7±0.1	15.3±0.1
`pranaydeeps/Ancient-Greek-BERT`	Singh et al. (2021)	12	768	113M	87.3±0.1	60.0±0.0
`nlpaueb/bert-base-greek-uncased-v1`	Koutsikakis et al. (2020)	12	768	113M	84.6±0.3	53.9±0.1

Table 5.6: **LAS on GRC-PROIEL (Dev)** using BAP and DepProbe with different LMs (± standard deviation).

Models	Source	Layers	Emb d	Params	BAP	DepProbe
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	86.7±0.2	60.2±0.6
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	88.8±0.1	59.2±0.3
`google/rembert`	Chung et al. (2021)	32	1152	576M	90.5±0.1	11.6±0.4
`onlplab/alephbert-base`	Seker et al. (2021)	12	768	126M	89.6±0.1	61.4±0.2

Table 5.7: **LAS on HE-HTB (Dev)** using BAP and DepProbe with different LMs (± standard deviation).

Models	Source	Layers	Emb d	Params	BAP	DepProbe
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	83.8±0.2	46.6±0.2
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	86.1±0.1	49.4±0.3
`google/rembert`	Chung et al. (2021)	32	1152	576M	86.1±0.2	15.9±0.3
`klue/bert-base`	Park et al. (2021)	12	768	111M	86.8±0.0	51.0±0.1
`klue/roberta-large`	Park et al. (2021)	24	1024	337M	88.1±0.3	48.8±0.5
`kykim/bert-kor-base`	Kim (2020)	12	768	118M	86.8±0.1	46.9±0.4

Table 5.8: **LAS on KO-GSD (Dev)** using BAP and DepProbe with different LMs (± standard deviation).

Models	Source	Layers	Emb d	Params	BAP	DepProbe
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	89.1±0.1	60.7±0.1
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	90.0±0.2	59.9±1.1
`google/rembert`	Chung et al. (2021)	32	1152	576M	90.8±0.0	26.0±0.2
`cointegrated/rubert-tiny`	Dale (2021)	3	312	11M	76.7±0.1	41.5±0.6
`sberbank-ai/ruRoberta-large`	Sber Devices (2021)	24	1024	355M	90.3±0.3	63.2±0.4
`blinoff/roberta-base-russian-v0`	Blinov (2021)	12	768	124M	75.8±0.0	15.6±0.2

Table 5.9: **LAS on RU-GSD (Dev)** using BAP and DepProbe with different LMs (± standard deviation).

Models	Source	Layers	Emb d	Params	BAP	DepProbe
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	87.5±0.1	55.5±0.2
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	90.2±0.1	59.1±0.2
`google/rembert`	Chung et al. (2021)	32	1152	576M	91.3±0.3	31.7±0.3
`KB/bert-base-swedish-cased`	Malmsten et al. (2020)	12	768	125M	90.8±0.1	61.7±0.2

Table 5.10: **LAS on SV-Talbanken (Dev)** using BAP and DepProbe with different LMs (± standard deviation).

Models	Source	Layers	Emb d	Params	BAP	DepProbe
`bert-base-multilingual-cased`	Devlin et al. (2019)	12	768	178M	84.6±0.4	49.1±0.4
`xlm-roberta-base`	Conneau et al. (2020)	12	768	278M	85.5±0.3	30.3±0.1
`google/rembert`	Chung et al. (2021)	32	1152	576M	85.3±0.2	5.2±0.1
`bert-base-chinese`	Devlin et al. (2019)	12	768	102M	85.8±0.1	46.4±0.1
`hfl/chinese-bert-wwm-ext`	Cui et al. (2021)	12	768	102M	86.0±0.3	45.8±0.3
`hfl/chinese-roberta-wwm-ext`	Cui et al. (2021)	12	768	102M	85.9±0.3	47.7±0.4

Table 5.11: **LAS on ZH-GSD (Dev)** using BAP and DepProbe with different LMs (± standard deviation).

Part III

Domain Variation

6 Can Humans Identify Domains?

The work presented in this chapter is based on the publication: Maria Barrett, Max Müller-Eberstein, Elisa Bassignana, Amalie Brogaard Pauli, Mike Zhang, and Rob van der Goot. 2024. Can humans identify domains? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, Torino, Italy. European Language Resources Association.

Abstract

Textual *domain* is a crucial property within the Natural Language Processing (NLP) community due to its effects on downstream model performance. The concept itself is, however, loosely defined and, in practice, refers to any non-typological property, such as genre, topic, medium or style of a document. We investigate the core notion of domains via human proficiency in identifying related intrinsic textual properties, specifically the concepts of genre (communicative purpose) and topic (subject matter). We publish our annotations in **TGeGUM**: A collection of 9.1k sentences from the GUM dataset (Zeldes, 2017) with single sentence and larger context (i.e., prose) annotations for one of 11 genres (source type), and its topic/subtopic as per the Dewey Decimal library classification system (Dewey, 1952), consisting of 10/100 hierarchical topics of increased granularity. Each instance is annotated by three annotators, for a total of 32.7k annotations, allowing us to examine the level of human disagreement and the relative difficulty of each annotation task. With a Fleiss' kappa of at most 0.53 on the sentence level and 0.66 at the prose level, it is evident that despite the ubiquity of domains in NLP, there is little human consensus on how to define them. By training classifiers to perform the same task, we find that this uncertainty also extends to NLP models.

6.1 Introduction

The concept of "domain" is ubiquitous in Natural Language Processing (NLP), as differences between "sublanguages" have strong effects on model transferability (Kittredge and Grisham, 1986). This issue of domain divergence has prompted comprehensive surveys on how to best adapt language models (LMs) trained on one or more source domains to more specific targets (Ramponi and Plank, 2020; Ramesh Kashyap et al., 2021; Saunders, 2022), and remains an open issue, even with LMs of increasing size (Ling et al., 2023; Singhal et al., 2023; Wu et al., 2023). Despite its importance, what constitutes a domain remains loosely defined, typically referring to any non-typological property that degrades model transferability. In practice, textual properties with the largest domain effects relate to a document's genre/medium/style (McClosky,

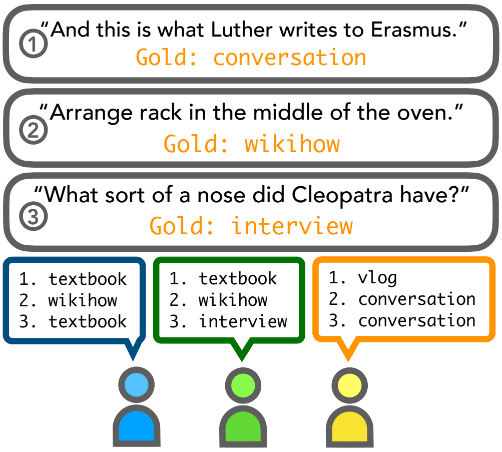


Figure 6.1: Graphical illustration of our triple-annotation setup with gold genre labels.

2010; Plank, 2011; Müller-Eberstein et al., 2021b), topic (Lee, 2001; Karouzos et al., 2021), or mixtures thereof (Aharoni and Goldberg, 2020). More broadly, domains can be viewed as a high-dimensional space with variation across the aforementioned properties, plus factors such as author personality, age, or gender (Plank, 2011, 2016).

We attempt to gain a better understanding of the foundational concept of domain, by taking a step back from modeling this phenomenon, and instead investigating whether humans themselves can distinguish between different instantiations of domain-related properties of textual data. In linguistics literature, these properties are separated into register, style and genre (Biber, 1988; Biber and Conrad, 2009, 2019), of which we choose to focus on *genre*, as it distinguishes itself from register and style by remaining consistent across complete texts. In addition, we examine the orthogonal factor of *topic*, i.e., the subject matter of a text, which can be expressed independently of genre (Kessler et al., 1997; Lee and Myaeng, 2002; Stein and Zu Eissen, 2006; Webber, 2009). We operationalize these two factors analogously to van der Wees et al. (2015) as genre stemming from different source types with distinct communicative styles, and topic being the principal subject matter of a given text.

More formally, our main research question is: *To what extent can* *humans detect genres and topics from text alone, and how does this align with machines?* We investigate the human proficiency in detecting these intrinsic properties by turning our attention to the Georgetown University Multilayer Corpus (GUM; Zeldes, 2017),[1] a large-scale multi-layer corpus consisting of texts from 11 different source types (henceforth *genre*). These act as gold annotations against which we compare the manual genre labels provided by 12 human annotators for the entirety of the corpus (Figure 6.1). In addition, the annotators supply a new annotation layer regarding the texts' subject matter (henceforth *topic*). As no gold labels are available for topic, they are annotated according to Dewey Decimal Classification (DDC; Dewey, 1952), a library classification system that allows new books to be added to a collection based on the subject matter. The DDC consists of 10 topics, 100 fine-grained topics, and 1,000 even finer-grained topics, of which we investigate the former two in detail and provide a preliminary study on the latter.

To understand the importance of context, we have annotators label genre and topic at both the sentence and prose level (defined as sequences of five sentences), and compare annotator agreement. Due to the subjective uncertainty associated with these types of characteristics, we gather three annotations per instance, measure their agreement, and release them in their unaggregated form as multi-annotations for future research.

Finally, we investigate the ability of machines to identify the same characteristics by training multiple ablations of genre and topic classifiers. Concretely, these experiments examine the difficulty of discerning each property, whether metadata or human notions of genre are more easily recoverable, as well as which level of context is most appropriate for the different ways in which the genre and topic label distributions can be represented.

Overall, this work is the first to explore the discernability of domain by both humans *and* machines. In Section 6.5, we further discuss the implications of our findings, both with respect to domain-sensitive downstream applications, as well as for the NLP community's more general definition of domain. Our contributions thus include:

[1] `https://gucorpling.org/gum/`

- **TGeGUM** (Topic-Genre GUM), a multi-layer extension of GUM, covering 9.1k sentences triple-annotated for a diverse set of 11 genres and 10/100 topics (Section 6.3).[2]
- An in-depth exploratory data analysis of the human annotations concerning annotator disagreement, uncertainty, and overall trends for domain characteristics across different context sizes (Section 6.4).
- A case study on the capability of NLP models to discern the human notions of genre and topic, as well as an analysis of which factors affect classification performance (Section 6.5).

6.2 Related Work

Domains Initially coined as "sublanguages" (Kittredge and Lehrberger, 1982; Kittredge and Grisham, 1986), domains have long been a topic of study in traditional linguistics and NLP (Lee, 2002; Lee and Myaeng, 2002; Stein and Zu Eissen, 2006; Eisenstein et al., 2014; van der Wees et al., 2015; Plank, 2016). Some of the early work mentioning domains as textual categories include Sekine (1997); Ratnaparkhi (1999), which categorize texts into, e.g., "general fiction", "romance & love", and "press:reportage". However, as also mentioned by Lee (2002); Lee and Myaeng (2002); Plank (2011); van der Wees et al. (2015), the concept of domain is under-defined. Plank (2011) considers domains as a multi-dimensional space, spanning all kinds of variability between texts, such as genre, topic, style, medium, etc. In this work, we follow a definition of domains similar to van der Wees et al. (2015), focusing on two of the largest dimensions of variability: i.e., *genres* (the communicative purpose and style) as well as *topics* (the subject matter). The former is closely tied to the source of a text, such as academic papers versus fiction books, while the latter may include subjects such as sports, politics, and philosophy, which can occur in multiple genres.

[2] Data and code can be found at `https://bitbucket.org/robvanderg/humans-and-domains`.

Automatic Domain Detection In NLP, automatic domain detection is essential for ensuring robust downstream performance, as it degrades with increasing levels of domain shift (Ramponi and Plank, 2020). Since this issue occurs independently of the application, domain classification has been explored in many contexts. Generally, the problem is either phrased in terms of a binary task, i.e., whether a target text matches the domain of the training data or not (e.g., Tan et al., 2019; Pokharel and Agrawal, 2023), or a multi-label classification task, in which the exact domain is to be determined (e.g., Müller-Eberstein et al., 2021a). Here, we use the latter approach as it requires a more formalized operationalization of domain.

At a broader level, genre is frequently used as a proxy for domain, as it has lower internal variability than many more specific dimensions, including topic (Kessler et al., 1997; Webber, 2009). Its automatic detection has been leveraged for selecting training data for transfer learning across a broad range of applications, such as classification (Ruder and Plank, 2017; van der Goot et al., 2021a; Gururangan et al., 2020) and generative tasks (Aharoni and Goldberg, 2020). Beyond English, genre has further been shown to provide a cross-lingually consistent signal for enabling more robust transfer in syntactic parsing (Müller-Eberstein et al., 2021a).

Topics provide a more granular differentiation between texts, also with close ties to domain. Automatically detecting topics has more immediate practical implications, as knowledge of the subject matter is critical for many downstream information extraction systems (Liu et al., 2021b; Bassignana and Plank, 2022) and more datasets with topic annotations are available (Sandhaus, 2008; Maas et al., 2011; Wang and Manning, 2012; Zhang et al., 2015); however, these works typically contain source data from only a single corpus.

Going beyond prior work with limited sets of post-hoc topic labels for single-genre corpora, we build on the general-purpose DDC system (Dewey, 1952) for libraries and apply its hierarchical set of 10/100 topics to a corpus containing data from 11 genres. By building on the existing annotations of the GUM dataset (Zeldes, 2017), we further enable research not only ascertaining to domain classification for its own sake, but also with applications to other downstream NLP tasks.

Multi-annotations Given the subjective nature of domains and their associated properties of genre and topic, each text in our dataset is annotated multiple times and retains individual labels without aggregating them. This approach of *multi-annotations* (Plank, 2022) avoids obscuring human uncertainty in the annotation process and has benefits both for tasks with high variability, such as ours, as well as tasks for which a ground truth is typically assumed.

E.g., Plank et al. (2014) map part-of-speech (POS) tags from Gimpel et al. (2011) to the universal 12-tag set by Petrov et al. (2012), retaining five *crowdsourced* POS labels per token.

For Relation Classification (RC), Dumitrache et al. (2018) obtained annotations for 975 sentences for medical RC, where each sentence is annotated by at least 15 annotators on average.

For Natural Language Inference (NLI), Nie et al. (2020) released Chaobowman-etal-2015-large: A dataset with 4,645 examples and 100 annotations per example for some existing data points in the development set of bowman-etal-2015-large (Bowman et al., 2015), williams-etal-2018-broad (Williams et al., 2018), and Abductive NLI (Bhagavatula et al., 2020). For a more in-depth overview of multi-annotation datasets, we refer to Uma et al. (2021).

6.3 The Dataset

6.3.1 Source Data

The source dataset on top of which we build our domain-related annotations is the GUM corpus which in turn incorporates data from a wide variety of sources. We use the portion of the GUM corpus released as part of the Universal Dependencies project (UD; Nivre et al., 2017), i.e., excluding Reddit. Since a text's source is closely tied to its communicative purpose, we consider GUM's *data source* metadata field of each instance as the gold genre label. For the topic, no equivalent gold label is discernible from the metadata.

The entire dataset is annotated both at sentence and prose level to investigate the importance of context for genre and topic annotation. For this purpose, we follow the gold sentence segmentation provided by GUM. We opted for these blocks instead of paragraphs, as the latter

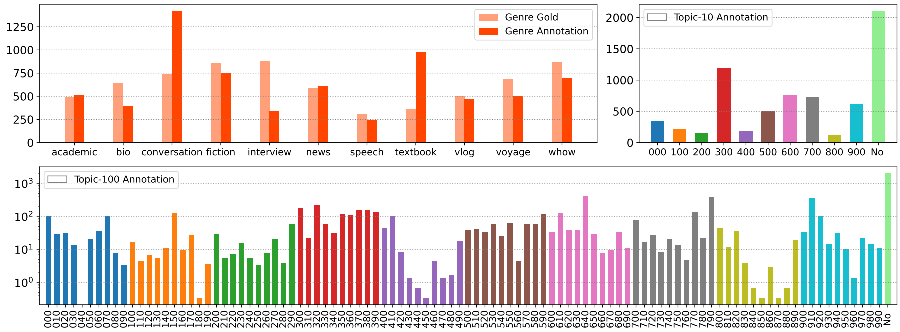


Figure 6.2: Frequency distributions of the labels in gold genre labels, annotations of genres, annotations of topic-10, and annotations of topic-100 (log scale) on sentence level. For the human annotations, the number is divided by three in order to align with the (unique) gold label. The mapping of topic-10 and topic-100 labels can be found in Section 6.8.6. The tag "No" in the topic annotations refers to *no-topic*.

are not natural dividers for all text types and can have a high variety of conventions and functions across genres. To avoid the same annotator observing the same sentence individually as well as in prose, we shuffle the dataset such that annotations of a sentence with and without context are distributed across different annotators, while maintaining coverage of the full dataset.

6.3.2 Annotation Procedure

Since there are no official descriptions of the genres in GUM, our annotation guidelines refer to the descriptions from the homepages of the websites of the source or the corresponding abstracts from Wikipedia. For topic annotation, we follow the Dewey Decimal library classification system (Dewey, 1952) consisting of 10/100/1,000 hierarchical topics of increased granularity. We consider the 10 high-level and the 100 mid-level classes for the coarse- and fine-grained topic annotations. We constrain our guidelines such that topic-100 should always be a sub-type of topic-10. For example, if topic-100 is "520 Astronomy", then topic-10 should be "500 Science". When none of the topic-100 labels fit the fine-grained topic of the instance, the annotators were allowed to leave the more specific topic blank, i.e., annotating topic-100 with the same label as topic-10. In addition, we include the *no-topic* label for when it is not possible to identify a specific topic from the provided text., such as for very short sentences, like “Ok” or “I agree with that.”

We completed an initial annotation round of 20 instances with all annotators and authors of this paper to evaluate the guidelines and annotation setup. None of this data is included in the final dataset. We continued with groups of three annotators annotating different subsets of the data. After an introductory meeting, further unclarities were discussed asynchronously throughout the process. Annotators were asked to pose their questions in general terms and to not use direct examples as to not bias the other annotators on specific instances. We did not conduct inter-annotator studies over the course of annotation and only had minor guideline revisions during the annotation process since we are mostly interested in human intuitions of genre and topic, and there are no gold labels for the topic task.

Annotators could indicate whether they were unsure about the annotation of a specific instance, and were also asked to provide notes/comments, if applicable. The annotation rate started at approximately 80–150 instances per hour. To ensure a similar amount of effort across annotators, we asked them to aim for approximately 150 instances per hour (also considering that annotation speed increases over time).

In total, we hired 12 annotators, who were paid 34,21 EUR per hour (before tax) for a total of 32 hours per person over a period of 4 weeks. The mean age was 27 (±2), and their highest completed education was equally split between a bachelor’s and a master’s degree. All rated their English skills as either C2/proficient or native. Seven annotators were reported to be female, three male, and two other/non-binary.

6.3.3 Dataset Statistics

Table 6.1 shows the final dataset statistics of **TGeGUM**. The dataset includes around 9.1K sentences, and 1.8K prose, each of them annotated by three individual annotators for genre, coarse-grained topic,

	Instances		Annotations	
	Sentence	Prose	Sentence	Prose
Train	6,911	1,358	20,733	4,074
Dev.	1,117	217	3,351	651
Test	1,096	221	3,288	663
Total	9,124	1,796	27,372	5,388

Table 6.1: Dataset Statistics: Note that each instance has three associated annotations.

and fine-grained topic.

In Figure 6.2, we report the sentence-level distribution of gold labels and human annotations, reporting the average number of annotations per label (total number of annotations divided by three annotators) to align with the singular gold genre metadata. For topic-10 and topic-100 we only report the human annotations as no gold labels exist.

Comparing gold and annotated genre labels, we observe a skew towards *conversation* and *textbook*. We hypothesize that this is due to the small amount of context an annotator receives. For example, the sentence “Is that all that’s left?” with the gold genre label *fiction* is annotated by all annotators as *conversation*. Another example is the sentence “Some of the greatest poetry has been born out of failure and the depths of adversity in the human experience.” with gold label *interview*. All annotators annotated this example as *textbook*.

For topic, we note that despite skewness, almost all 100 topics are used. The *300 Social sciences* including, e.g., *320 Political science* and *370 Education*, stand out as being the most prevalent topics. The most frequent label, however, is *no-topic*, indicating that it is challenging to identify a specific topic given only one sentence and that individual sentences can be associated with different topics, depending on the surrounding context.

The genre distribution at the prose level (Section 6.8.4) reveals a more accurate distribution for *conversation*-like utterances; however, the general skew towards *textbook* remains. Concerning topic, the main contrast to the sentence-level distributions is the reduction of

	Kappa			Maj. Acc.
	Genre	Topic-10	Topic-100	Genre
Sentences	0.5260	0.5213	0.4239	67.68
Prose	0.6582	0.5238	0.3838	81.11

Table 6.2: Agreement scores across annotators, and accuracy of majority vote among annotators compared to gold genre labels.

the *no-topic* label, confirming that more context is crucial for this task.

6.4 Exploratory Data Analysis

In addition to the previous aggregated overview, we are interested in exploring whether domain characteristics are recoverable by humans in a consistent manner. While we can compare human annotations to the original gold labels for genre, no equivalent is available for topic. Therefore, we place more emphasis on inter-annotator agreement, in the form of Fleiss' Kappa (Fleiss, 1971), to measure intuitive alignment and ease of identification. Table 6.2 and Figure 6.3 shows this agreement across the different genres, topics and levels of available context.

6.4.1 Human Genre Detection

Accuracy and Agreement Considering that annotation guidelines were phrased to avoid any intentional alignment to an existing ground truth (i.e., annotators were unaware of the existence of gold genre labels), an accuracy of 67.68% at the sentence level shows that genre is recoverable to a far higher degree than by random chance or by a majority baseline. This further increases to 81.11% given more context at the prose level and is also reflected in the increase from moderate inter-annotator agreement (0.53) to substantial agreement (0.66).

The additional context appears to help differentiate genres that have more similarities to each other. This phenomenon is especially pronounced for spoken-language data, such as *conversation, interview* and *vlog,* which differ with respect to genre-specific conventions such

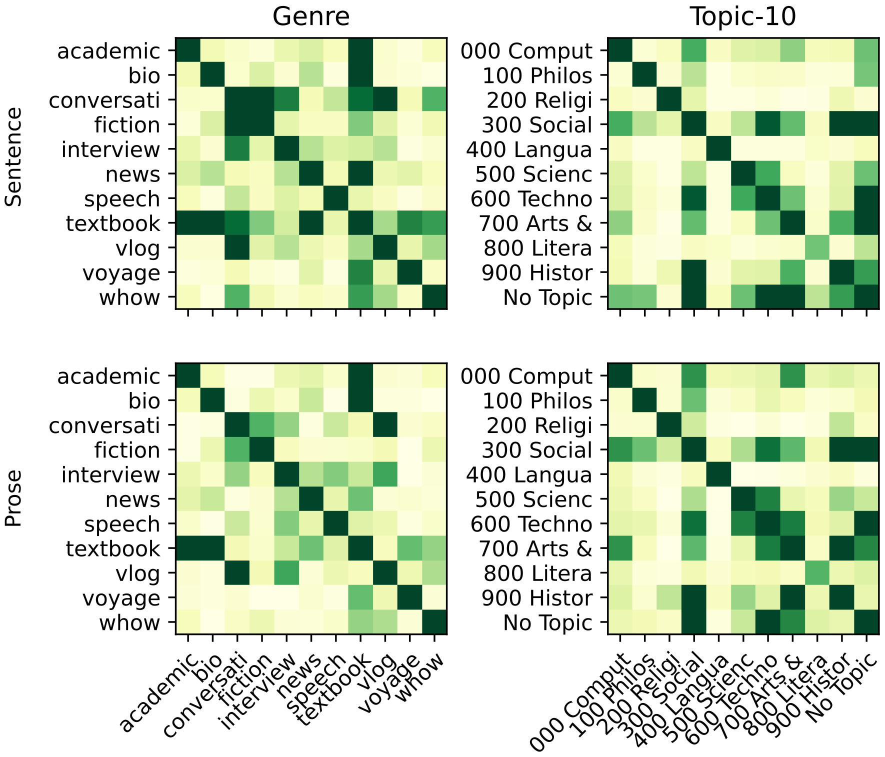


Figure 6.3: Confusion matrix with all annotated pairs of labels for Genre and Topic-10 (across all annotators) in our training data: The darker the color, the higher the number of annotations for that label pair. The diagonal can be seen as agreement, whereas off-diagonal is a proxy for disagreement.

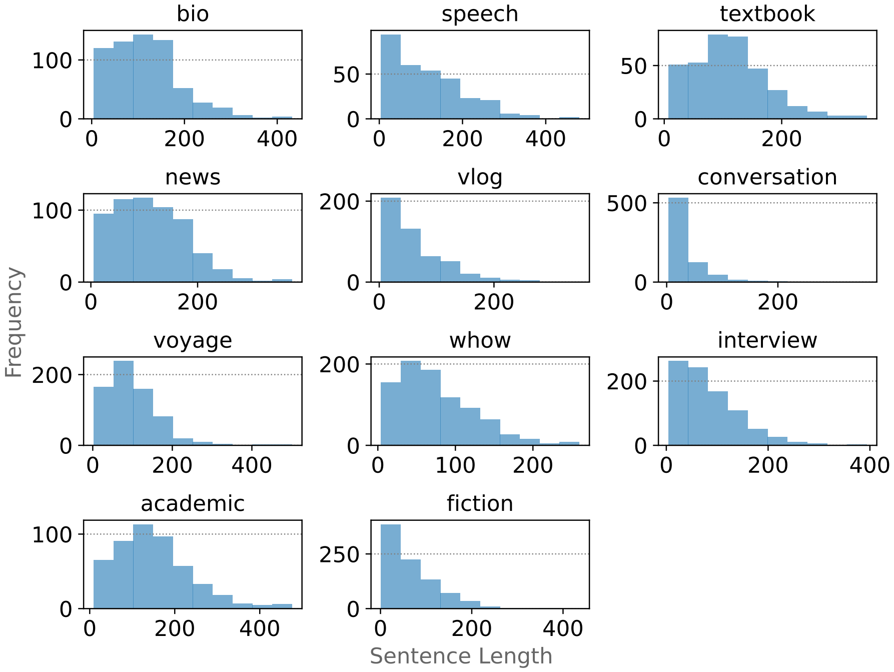


Figure 6.4: Frequency of sentence lengths, measured by the number of characters, per gold genre.

as who the speech is directed towards (i.e., bi-directional, interviewee, video viewer), or how formal the register is. Both properties are more easily discernible across multiple turns.

Nonetheless, even given more context, high amounts of confusion remain between certain genres such as non-fiction texts of the type *academic*, *biography*, and *textbook*. These are intuitively similar to each other and may require even more context to distinguish. Generally, genres appear to lie on a more continuous spectrum that is difficult to discretize in conceptually similar cases.

Human Uncertainty In case of uncertainty, annotators were encouraged to select a "best guess" label and to indicate uncertainty by ticking a checkbox. In addition to overall uncertainty, we also hypothesize that sentence length affects accuracy due to the amount of information available. To evaluate these two effects for genre detection, we measure the Pearson correlation between human accuracy concerning the gold label, with 1) sentence length, 2) the number of uncertainty flags (Table 6.3). As expected, longer sentences are annotated correctly more often. Figure 6.4 further highlights how spoken-language genres have a strong skew towards shorter sentences, and for which annotators have the lowest agreements. Additionally, sentences marked as “unsure” align with gold labels less often, showing that annotators appear to have well-calibrated judgments of their own uncertainty, even for this relatively difficult task.

6.4.2 Human Topic Detection

Agreement In the absence of gold labels, inter-annotator agreement allows us to estimate the difficulty of discerning broader vs granular topics. For the 10 broader topics, Table 6.2 shows a moderate agreement of 0.52 for both the sentence and prose levels. As expected with an order of magnitude more labels, Topic-100 sees a drop in agreement to 0.42 and an additional drop to 0.38 at the prose level. While this may seem counter-intuitive due to topic’s higher specificity compared to genre, Figure 6.3 sheds some light on this peculiarity: In contrast to genre, topic has a *no-topic* label (Section 6.3.2), which, in turn, is used frequently by all annotators at the sentence level, due to the absence of any subject matter in many shorter utterances—especially in speech. Given the additional context, topic becomes more apparent, and agreement spreads toward more topics along the diagonal. As such, sentence-level agreement mainly hinges on *no-topic*, while prose-level annotations agree more with respect to actual topics. This is less apparent for 10-topic kappa, for which this effect cancels out, but is more prevalent with 100 topics, where the shift away from *no-topic* at the prose level comes with a much wider spread of topics, thereby reducing overall agreement, despite having a higher level of true topic annotations.

Overall, topics which were most consistently identified include *social sciences*, *arts & recreation*, *technology*, *science* and *history & geography*. On the other hand, *literature* was least consistently annotated and most frequently confused with the aforementioned topics, potentially due to its broader scope compared to the others.

	Sent	Prose
length vs unsure	-0.1126*	-0.0474
length vs correct	0.1267*	-0.0385
unsure vs correct	-0.2948*	-0.3411

Table 6.3: Correlations across utterance length, correct predictions of human majority vote, and the number of unsure annotations.
* indicates statistical significance for $p < 0.05$.

1,000 Topics After completing the full set of genre and topic-10/100 annotations with three annotators per instance, the remaining time of the annotators was spent on a preliminary study to label the most fine-grained categories of DDC. With 1,000 labels, this task is substantially more difficult. We obtained a total of 904 sentences and 172 prose sequences with three annotations each.[3] Measuring inter-annotator agreement at this level of granularity, we find a Fleiss' Kappa of 0.32 for sentences and 0.26 for prose. Although substantially lower than for coarser topic granularities as well as genre, this score still indicates above-random agreement among annotators. Similarly to the previous topic results, prose-level context allows humans to detect more actual topics than *no-topic*, leading to lower overall agreement but a broader coverage of actual topics.

In general, despite the importance of topic to downstream applications (i.e., topic classification as a task in itself), there is no clear human consensus regarding discrete topic classification. Similarly to genre, topic appears to be a concept for which human intuition shares some agreement at a broader level, but is also spread along a continuum—especially as granularity increases.

6.5 Modeling Domain

Following our examination of human notions of genre and topic, we investigate automatic methods' ability to model the same properties. Ablating across different setups for representing the multiple anno-

[3] From 3,918 total annotations, we discarded instances with less than three completed annotations.

tations per instance (Section 6.5.1), we train models to classify genre and topic at different levels of granularity (Section 6.5.2) and evaluate their ability to learn the underlying distribution (Section 6.5.3). While pre-neural work typically performed document-level classification (Webber, 2009; Petrenz and Webber, 2011), contemporary trends have shifted towards the sentence-level (Aharoni and Goldberg, 2020; Müller-Eberstein et al., 2021b). Leveraging our multi-level annotations, we investigate genre and topic classification at both the sentence and prose-level, mirroring our human annotation setup.

6.5.1 Setup

Most work on modeling multiple annotators is based on tasks consisting of only two or three labels, e.g., hate speech detection, or RTE (Uma et al., 2021). An exception is Kennedy et al. (2020), who use multiple classification heads to predict a score for a variety of aspects of hate speech, which are then used to predict a final floating point score for hate speech detection. Other related work predicts multiple task labels simultaneously (e.g., Demszky et al., 2020; Kiesel et al., 2023; Piskorski et al., 2023), however these are typically discrete and do not model annotator certainty. We propose a variety of methods to model the distribution of the annotations (overview in Figure 6.5):

Majority Discretizes the labels using a majority vote, and uses a single classification head to predict it. For the distribution similarity metric (see below), we assign a score of 1.0 to the chosen label.

PerLabel-Regression Converts the human annotations to scores per label and then predicts these as a regression task. Each label has its own decoder head, trained using an MSE loss, and mapped to the [0;1] range afterwards.

PerLabel-Classification Converts the human annotations into score bins and predicts them as four possible labels: “0.0”, “0.33”, “0.66”, “1.0”.

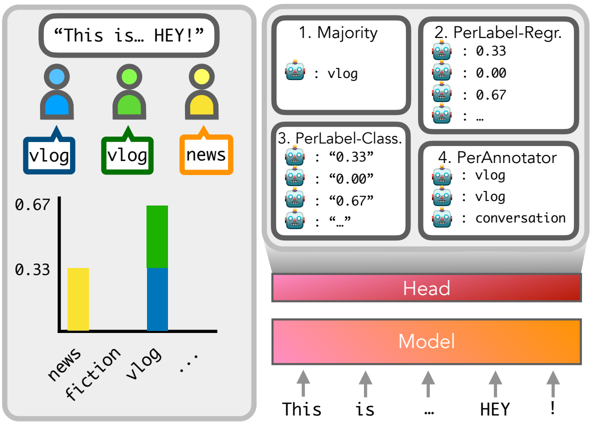


Figure 6.5: The target value each model variant is trained to predict: 1) Majority vote. 2) PerLabelRegr(ession) on label distributions. 3) PerLabel-Class(ification), on score bins per label. 4) PerAnnotator, three different annotations.

PerAnnotator One decoder head modeling each annotator, that predicts their annotation as a discrete label. Afterwards, the three predictions are converted to a distribution.

We evaluate these models using the standard accuracy over each singular predicted label (i.e., highest score or majority). In addition, we conduct a finer-grained evaluation that takes the multi-annotations into account. For this purpose, we propose a similarity metric for comparing the predicted and annotated label distribution per instance. Let n be the number of label types, and X and Y are label distributions that sum to 1, with a score for each label. Then, the distributional similarity per instance can be computed as:

$$distr\_sim = 1 - \frac{\sum_{l=0}^{n} |X_n - Y_n|}{2} \quad .$$

The resulting score between 0 and 1 represents the distributions' similarity. Note that we compare model predictions to the human annotations, which are not a gold standard; here, we aim to determine whether the human ability to discern these concepts is easy to model.

We implement all our model variants in the MaChAmp (van der

	Accuracy	Macro-F1	$\|N\|$
Sentence	67.68	59.92	1,117
Prose	81.11	74.75	217

Table 6.4: Performance of annotators' majority vote compared with the gold genre (development set).

Goot et al., 2021b) toolkit v0.4 using default parameters. MaChAmp is a toolkit focused on multi-task learning for NLP, and allowed us to implement all varieties of the tasks described earlier. Each way of phrasing the task is implemented on top of a single language model for fair comparison. From an initial evaluation of the bert-large-cased (Devlin et al., 2019), luke-large-lite (Yamada et al., 2020), deberta-v3-large (He et al., 2021), xlm-roberta-large (Conneau et al., 2020) LMs on the gold genre labels, we identify that DeBERTa has the highest accuracy; hence we use it in the following experiments.

6.5.2 Classification Results

We examine which notion of domain is more learnable and distinguishable for a model; genre or topic? Since genre has associated ground truth labels, we additionally examine whether the human annotators' perception of genre or the ground truth genre is easier to learn.

We establish a majority vote based on the human annotations; in case of a tie, the first element in the annotation list is chosen as the label, both for sentences and prose. This happens in ~10% of cases for genre and topic-10 (sentence and prose), and ~20% cases for topic-100.

Table 6.4 shows accuracy and macro-F1 scores of the annotators' majority vote evaluated against the gold genre. As noted previously, more context (prose level) helps disambiguate the genre.

To evaluate how well a model can align with the human intuition of genres and topics, we fine-tune an LM on the majority labels of the annotators. We compare the performance on the gold genre labels (the only task for which we have gold labels) and compare the accuracy and macro-F1 scores (Table 6.5). We notice the following:

		Accuracy	Macro-F1
Sent.	gold_genre	73.20± 0.02	70.74± 0.02
	maj_genre	75.88± 0.01	67.04± 0.01
	maj_topic-10	75.56± 0.02	60.54± 0.07
	maj_topic-100	64.55± 0.00	18.43± 0.02
Prose	gold_genre	89.49± 0.02	88.02± 0.03
	maj_genre	80.83± 0.01	74.97± 0.03
	maj_topic-10	67.74± 0.01	50.35± 0.03
	maj_topic-100	52.35± 0.01	16.04± 0.02

Table 6.5: Accuracy and Macro-F1 on test split, for DeBERTa models fine-tuned and evaluated on gold genre, human majority vote for genre, and human majority vote for topic-10/100 (standard deviations across five seeds).

Sentences 1) Unsurprisingly, DeBERTa fine-tuned on the gold genre labels (gold_genre) is better aligned with the ground truth genre than the human majority vote, i.e., 73.20 (Table 6.5) versus 67.68 (Table 6.4) accuracy at the sentence level (note that other LMs performed worse). 2) In contrast, the fine-tuned DeBERTa model has higher accuracy when trained and tested on the human majority vote (maj_gerne) than when using gold genre labels (gold_genre), i.e., 75.88 versus 73.20, although macro-F1 is lower. This indicates that less common genre labels are easier to learn from gold labels, while more frequent genres are easier to learn based on human intuitions. 3) Despite topic-10 having fewer classes than genre, the notion of topic appears to be more difficult for a model to learn (lower F1). 4) The skew of the fine-grained topics (maj_topic-100) and the difficulty of the long tail become apparent in the large divergence across the accuracy and macro-F1 score.

Prose 5) In contrast to the sentence level, our fine-tuned DeBERTa model generalizes better to the gold genre labels (gold_genre) than the human majority vote (maj_genre). At this level of context, the majority vote topic is also harder for a model to learn than the majority vote genre.

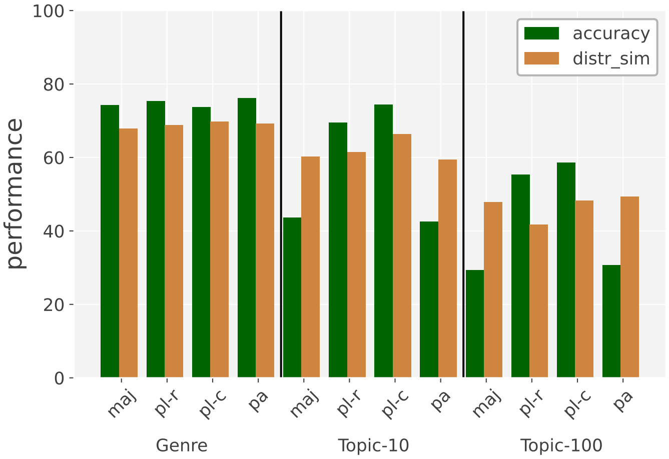


Figure 6.6: Accuracy and distributional similarity on test split, for DeBERTa models trained on target labels based on Majority vote (maj), PerLabel-Regression/Classification (pl-r/c), PerAnnotator labels (pa); standard deviations across five seeds.

6.5.3 Distributional Results

In Figure 6.6, we report the results of the models trained on all instances (sentences and prose) with DeBERTaV3-large.[4] The main trends show that the model performs better on the genre task. Unsurprisingly, for topics, the granularity of the labels impacts performance.

By modeling the annotation distributions (i.e., PerLabel-Regression/Classification), we can outperform the majority vote model. However, distributional similarity decreases with increased label granularity (i.e., from topic-10 to topic-100), showing that it is difficult for models to calibrate to diverging human judgments. Interestingly, the per-label models achieve comparable or higher scores on the $distr\_sim$ metric, showing that the examined LMs model label distributions more easily than annotator behavior.

[4] Training on sentences and prose separately leads to similar trends (Section 6.8.2).

6.6 Conclusion

To examine the widely used but scarcely defined notion of *domain*, this work provides the first investigation of human intuitions of this property in the form of **TGeGUM**: a collection of 9.1k sentences annotated with 11 genres and 10/100 topics by three annotators per instance, using an annotation procedure designed to capture human variability instead of forcing alignment (Section 6.3).

Our exploratory analysis (Section 6.4) shows that despite the subjective nature of this task, as reflected in a Fleiss' Kappa of 0.53–0.66, humans can identify certain domain characteristics consistently from one sentence alone. Nonetheless, genres with a high similarity benefit substantially from added context. This is even more crucial for identifying topics, where we observe a shift from annotators not being able to discern any topic at all to being able to reach an above-random agreement, even when presented with 100 or 1,000 topics.

Finally, our experiments of modeling these domain characteristics automatically (Section 6.5) show that genre is easier to model than topic. For the agreements between human annotators, and the performance from the automatic model, we see that context is crucial for the genre classification task, but not for topic classification, where adding context even leads to decrease in scores if the label space is large.

Overall, this work highlights that despite the importance of "domain", there is little consensus regarding its definition, both in the NLP community as well as in our human annotations. Taking a closer look at what intuition predicted, further reveals that genres and topics are difficult to discretize completely, and that a continuous space of domain variability may be more suited for characterizing these phenomena.

6.7 Ethics Statement

Our approach to modeling human label variation is intrinsically linked to the larger issue of human social bias. As highlighted by Plank (2022), significant social implications are tied to the study of label variation. In the context of our research, it is essential to acknowledge that variations in labeling might stem from societal biases and disparities. To address this, we recognize the necessity of addressing bias mitigation techniques as we aim to create more equitable and just models. However, we also contend that our focus on modeling generic subjects, such as genre and topic, may carry less severe implications compared to more subjective tasks like hate speech detection (Akhtar et al., 2021; Mostafazadeh Davani et al., 2022). The differences in annotations within our work may primarily relate to two categories: “Missing Information” and “Ambiguity” (Sandri et al., 2023).

Another ethical facet we must address is the potential biases present in the classification system we use. In particular, the Dewey Decimal Classification System, which is the de-facto standard for libraries worldwide, has been found to exhibit prejudice (Gooding-Call, 2021). For example, the classification of information related to religion, specifically within class 200, demonstrates a clear skew, with a majority of subjects (six out of ten) reserved for Christianity-related topics. The remaining four slots are designated for other dominant religions, with an *other* section meant to encompass all other belief systems. This reveals an inherent bias toward Christianity, which can affect the accessibility of non-dominant religions and belief systems. There are alternatives to knowledge organization systems like the Dewey Decimal Classification, as suggested by Franzen (2022), to promote a more inclusive and equitable information landscape.

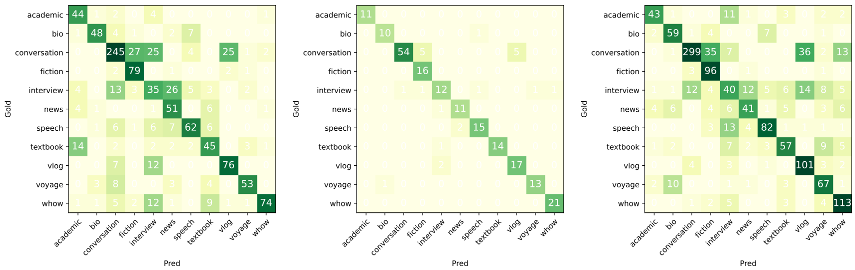
6.8 Appendix

6.8.1 Confusion Matrices Genre

In Figure 6.7a-Figure 6.7c we plot the confusion matrices of our DeBERTa model trained on the gold genre labels. The conversation genre shows to be the most difficult label; it is commonly confused with fiction, interview and vlog; which also overlap in length (Section 6.4).

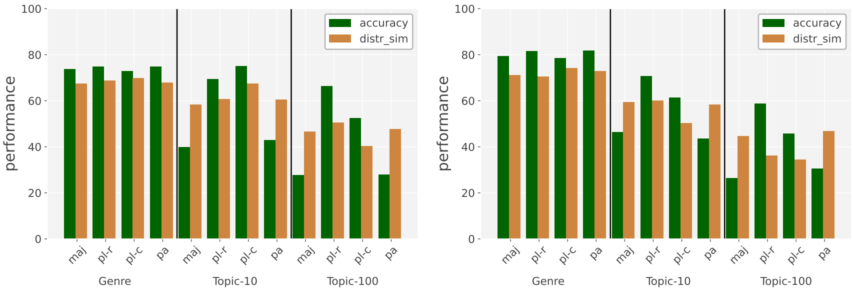
6.8.2 Sentence and Prose Results

In Figure 6.8a we show the results of our proposed models trained and evaluated only on the sentence level data. Figure 6.8b has the same evaluation on the prose level data.



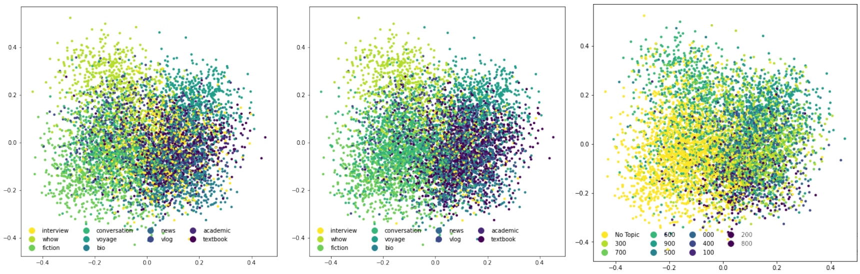
(a) Sentence Level. (b) Prose Level. (c) All Data.

Figure 6.7: Confusion matrices at different levels, with numbers summed over all five random seeds.



(a) Sentence-level Results. (b) Prose-level Results.

Figure 6.8: Results for models trained on sentence/prose-level data.



(a) Gold Genres. (b) Genre Annotations. (c) Topic Annotations.

Figure 6.9: PCA plots of sentence embeddings colored according to different majority-pooled annotation layers.

6.8.3 Visualization of Embeddings

We encode sentences using Sentence-BERT (Reimers and Gurevych, 2019), apply a PCA-downprojection, and color each sentence according to gold genres, our majority-vote genre annotations, as well as majority-vote topic-10 annotations. The results are shown in Figures 6.9a–6.9c.

6.8.4 Prose-level Statistics

Label statistics on the prose level are shown in Figure 6.10. While general trends, such as the majority genres and topics remain the same as on the sentence level, additional context spreads annotations more evenly, and allows for disambiguations such as for spoken data genres. This is also reflected in the higher alignment between gold and annotated genre labels—both in terms of number, but also in terms of accuracy (Table 6.2). For topic, we further observe almost an order of magnitude fewer no-topic annotations, which are consequently distributed across the spectrum of actual topics.

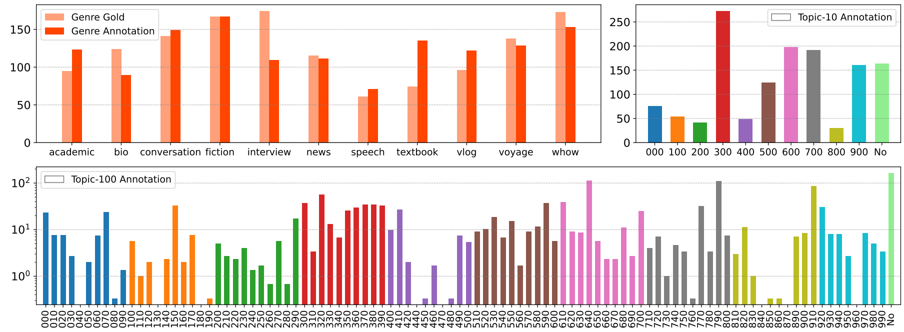


Figure 6.10: **Distribution of Labels (Prose).** Frequency distributions of the labels in gold genre labels, annotations of genres, annotations of topic-10, and annotations of topic-100 (log scale). For the annotations, the number is divided by three to get an average distribution. The mapping of topic-10 and topic-100 labels can be found in Section 6.8.6. The tag “No” in the topic annotations means “No topic”.

6.8.5 Annotator Comments

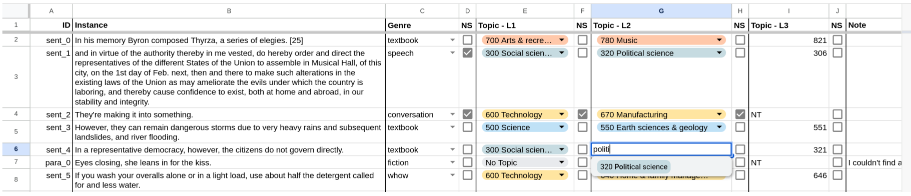
Annotators were provided with a free-form field to provide optional comments regarding each annotation. Of the final dataset, 3.9% of annotations have an annotator comment attached, with a median length of 38 characters. They primarily contain explanations of annotations which were marked with high annotator uncertainty.

6.8.6 Guidelines

The full annotation guidelines can be found at `https://personads.me/x/tgegum-guidelines`.

6.8.7 Annotation Tool

We used Google Spreadheets for annotation. The setup is shown in Figure 6.11.



	A	B	C	D	E	F	G	H	I	J	
1	ID	Instance	Genre	NS	Topic - L1	NS	Topic - L2	NS	Topic - L3	NS	Note
2	sent_0	In his memory Byron composed Thyrza, a series of elegies. [25]	textbook	□	700 Arts & recre…	□	780 Music	□	821	□	
3	sent_1	and in virtue of the authority thereby in me vested, do hereby order and direct the representatives of the different States of the Union to assemble in Musical Hall, of this city, on the 1st day of Feb. next, then and there to make such alterations in the existing laws of the Union as may ameliorate the evils under which the country is laboring, and thereby cause confidence to exist, both at home and abroad, in our stability and integrity.	speech	☑	300 Social scien…	□	320 Political science	□	306	□	
4	sent_2	They're making it into something.	conversation	☑	600 Technology	☑	670 Manufacturing	☑	NT	□	
5	sent_3	However, they can remain dangerous storms due to very heavy rains and subsequent landslides, and river flooding.	textbook	□	500 Science	□	550 Earth sciences & geology	□	551	□	
6	sent_4	In a representative democracy, however, the citizens do not govern directly.	textbook	□	300 Social scien…	□	politi 320 Political science	□	321	□	
7	para_0	Eyes closing, she leans in for the kiss.	fiction	□	No Topic	□		□	NT	□	I couldn't find a
8	sent_5	If you wash your overalls alone or in a light load, use about half the detergent called for and less water.	whow	□	600 Technology	□	640 Home & family manage…	□	646	□	

Figure 6.11: Example of annotation in Google Spreadsheets. NS = Not Sure

7 How Universal is Genre in Universal Dependencies?

The work presented in this chapter is based on the publication: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021b. How universal is genre in Universal Dependencies? In *Proceedings of the 20th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2021)*, pages 69–85, Sofia, Bulgaria. Association for Computational Linguistics.

Abstract

This work provides the first in-depth analysis of genre in Universal Dependencies (UD). In contrast to prior work on genre identification which uses small sets of well-defined labels in mono-/bilingual setups, UD contains 18 genres with varying degrees of specificity spread across 114 languages. As most treebanks are labeled with multiple genres while lacking annotations about which instances belong to which genre, we propose four methods for predicting instance-level genre using weak supervision from treebank metadata. The proposed methods recover instance-level genre better than competitive baselines as measured on a subset of UD with labeled instances and adhere better to the global expected distribution. Our analysis sheds light on prior work using UD genre metadata for treebank selection, finding that metadata alone are a noisy signal and must be disentangled within treebanks before it can be universally applied.

7.1 Introduction

Identifying document genre automatically has long been of interest to the NLP community due to its immediate applications both in document grouping (Petrenz and Webber, 2012) as well as task-specific data selection (Ruder and Plank, 2017; Sato et al., 2017).

Cross-lingual genre identification has however remained a challenge, mainly due to the lack of stable cross-lingual representations (Petrenz and Webber, 2012). Recent work has shown that pre-trained masked language models (MLMs) capture monolingual genre (Aharoni and Goldberg, 2020). Do such distinctions manifest in highly multilingual spaces as well? In this work, we investigate whether this property holds for the genre distribution in the 114 language Universal Dependencies corpus (UD version 2.8; Zeman et al., 2021) using the multilingual mBERT MLM (Devlin et al., 2019).

In absence of an exact definition of textual genre (Kessler et al., 1997; Webber, 2009; Plank, 2016), this work will focus on the information specifically denoted by the `genres` metadata tag in UD. We hope that an in-depth, cross-lingual analysis of what this label represents will enable practitioners to better control for the effects of domain shift in their experiments. Previous work using these UD metadata for proxy training data selection have produced mixed results (Stymne, 2020). We investigate possible reasons and identify inconsistencies in genre annotation. The fact that genre labels are only available at the level of treebanks makes it difficult to gather a clear picture of the *sentence-level* genre distribution — especially with some treebanks having up to 10 genre labels. We therefore investigate the degree to which instance-level genre is recoverable using only the treebank-level metadata as weak supervision.

Our contributions entail the, to our knowledge, first detailed definition of all UD metadata genre labels (Section 7.3), four weakly supervised methods for extracting instance-level genre across 114 languages (Section 7.4) as well as genre identification experiments which show that our proposed two-step procedure allows for effective genre recovery in multilingual setups where language relatedness typically outweighs genre similarities (Section 7.5).[1]

7.2 Related Work

The largest hurdle for cross-lingual genre classification is the lack of shared representational spaces. Sharoff (2007) use shared POS n-grams in order to jointly classify the genre of English and Russian documents. Petrenz and Webber (2012) similarly seek out features which are stable across languages in order to classify English and Chinese documents into four shared genres. A recent data-driven approach finds that monolingual MLM embeddings can be clustered into five groups closely representing the data sources of the original corpus (Aharoni and Goldberg, 2020). In this work, we investigate whether this holds for multilingual settings as well.

Being able to identify textual genre has been crucial for domain-specific fine-tuning (Dai et al., 2020; Gururangan et al., 2020) including dependency parsing. For parser training, in-genre data is typically selected by proxy of the data source (Plank, 2011; Rehbein and Bildhauer, 2017; Sato et al., 2017). Data-driven approaches which include automatically inferred topics based on word and embedding distribu-

[1] Code available at `https://personads.me/x/syntaxfest-2021-code`.

tions (Ruder and Plank, 2017) as well as POS-based approaches (Søgaard, 2011; Rosa, 2015; Vania et al., 2019) have also been found effective.

Universal Dependencies (Nivre et al., 2020) aims to consolidate syntactic annotations for a wide variety of languages and genres under a single scheme. The latest release contains 114 languages — many with fewer than 100 sentences. In order for languages at all resource levels to benefit from domain adaptation, it will continue to be important to identify cross-lingually stable signals for genre. While language labels are generally agreed upon, differences in genre are more subtle. Metadata at the treebank level provides some insights into genres of original data sources, however these are "neither mutually exclusive nor based on homogeneous criteria, but [are] currently the best documentation that can be obtained" (Nivre et al., 2020).

Stymne (2020) performs an initial study on using these treebank metadata labels for the selection of spoken and Twitter data. Results show that training on out-of-language/in-genre data is superior to out-of-language/out-of-genre data. However the best results are obtained using in-language data regardless of genre-adherence. This holds across multiple methods of proxy dataset selection (e.g. treebank embeddings; Smith et al., 2018a).

Recently, Müller-Eberstein et al. (2021a) have shown that combining UD genre metadata and MLM embeddings can improve proxy training data selection for zero-shot parsing of low-resource languages. The use of genre in their work is more implicit as it is mainly driven by the genre of the target data. In contrast, this work takes a holistic view and explicitly examines the classification of instance-level genre for all sentences in UD.

As genre appears to be a valuable signal, we set out to investigate how it is defined and distributed within UD. Due to the coarse, treebank-level nature of current genre annotations, we hypothesize that a clearer picture can only be obtained by moving to the sentence level. We therefore transition from prior supervised document genre prediction to weakly supervised *instance* genre prediction. Additionally, we expand the linguistic scope from mono- or bilingual corpora to all 114 languages currently in UD.

More generally, this task can be viewed as predicting genre labels for all sentences in all corpora of a collection while only being given the set of labels said to be contained in each corpus.

7.3 UD-level Genre

We analyze genre as currently used in the `genres` metadata of 200 treebanks from Universal Dependencies version 2.8 (Zeman et al., 2021). Section 7.3.1 provides an overview of all UD genre types and Section 7.3.2 analyzes how these global labels relate to the subset of treebanks which do provide treebank-specific, instance genre annotations.

7.3.1 Available Metadata

UD 2.8 (Zeman et al., 2021) contains 18 genres which are denoted in each treebank's accompanying metadata. Around 36% of treebanks contain a single genre while the remaining majority can contain between 2–10 which are not further labeled at the instance level. There is no official description of each genre label, however they can be roughly categorized as follows:

🎓 **academic** Collections of scientific articles covering multiple disciplines. Note that this label may subsume others such as *medical*.

☁ **bible** Passages from the bible, frequently from older languages (e.g. Old Church Slavonic-PROIEL by Haug and Jøhndal, 2008). Largely non-overlapping passages are used across treebanks.

📅 **blog** Internet documents on various topics which may overlap with other genres such as *news*. They are typically more informal in register. Some treebanks group social media content and reviews under this category (e.g. Russian-Taiga by Shavrina and Shapovalova, 2017).

✉ **email** Formal, written communication. This includes English-EWT's (Silveira et al., 2014) subsection based on the Enronsent Corpus (Styler, 2011) as well as letters attributed to Dante Alighieri as part of Latin-UDante (Cecchini et al., 2020).

fiction Mostly paragraphs from diverse sets of fiction books and magazines.

government The least represented genre, mainly denoting texts from governmental sources. These include political speeches (English-GUM by Zeldes, 2017) as well as inscriptions from Neo-Assyrian kings from around 900 BCE (Akkadian-RIAO by Luukko et al., 2020).

grammar-examples Sentences from teaching or grammatical reference books which are typically short, but cover a wide range of dependency relations (e.g. Tagalog-TRG by Samson and Cöltekin, 2020).

learner-essays Small genre occurring in three single-genre treebanks. Sentences were written by second-language learners and either contain original errors (English-ESL by Berzak et al., 2016), manual corrections (dinuovo2019valico by Di Nuovo et al., 2019) or both (Chinese-CFL by Lee et al., 2017).

legal Relatively frequent genre based mostly on laws and legal corpora within the public domain.

medical Scientific articles/books in the field of medicine (e.g. cardiology, diabetes, endocrinology for Romanian-SiMoNERo by Mitrofan et al., 2019). It is subsumed by *academic* for some treebanks (e.g. Czech-CAC by Hladká et al., 2008).

news The highest-resource genre by a large margin corresponding to news-wire texts as well as online newspapers on specific topics (e.g. IT-news in German-HDT by Borges Völker et al., 2019).

nonfiction Second most frequent genre with a high degree of variance, subsuming e.g. *academic* and *legal.* German-LIT (Salomoni, 2019) contains three philosophical books from the 18th century. Other *non-fiction* treebanks can originate from multiple sources (e.g. books and internet) and time spans.

poetry Smaller, yet distinct genre covering mostly older texts and language variations (e.g. Old French-SRCMF by Stein and Prévost, 2013).

reviews Medium-resource genre covering informal online reviews with unnormalized orthography (e.g. English-EWT) as well as formal reviews (e.g. newspaper film reviews in Czech-CAC).

social Encompasses social media data such as tweets (e.g. Italian-TWITTIRÒ by Cignarella et al., 2019) as well as newsgroups (e.g. English-EWT). Some *spoken* data is co-labeled with this genre when it refers to colloquial speech (e.g. South Levantine Arabic-MADAR by Zahra, 2020).

spoken Distinct genre which typically consists of spoken language transcriptions. Sentences contain filler words and may have abrupt boundaries. Sources range from elicited speech of native speakers (Komi Zyrian-IKDP by Partanen et al., 2018) to radio program transcriptions (Frisian Dutch-Fame by Braggaar and van der Goot, 2021).

web Similarly ambiguous genre as *non-fiction.* It occurs in conjunction with specific genres such as *blog* and *social* and never appears alone (e.g. Persian-PerDT by Safari et al., 2022).

W **wiki** Denotes data from Wikipedia for which cross-lingual authoring guidelines exist.

Figure 7.1 shows the approximated distribution of these genres in UD. Maximum/minimum sentence counts are inferred from the size of single-genre treebanks plus the size of all treebanks in which a

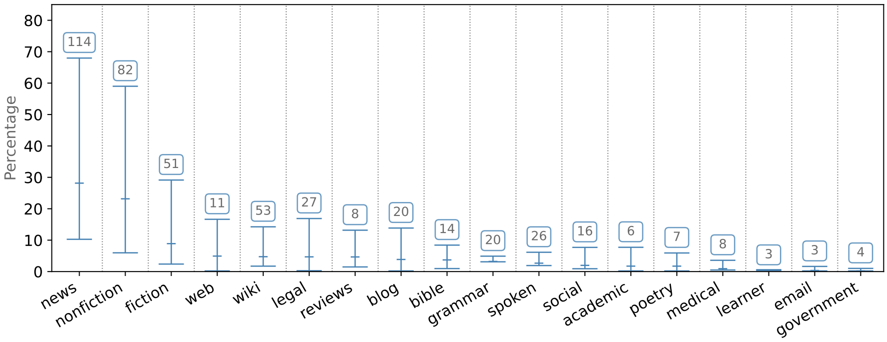


Figure 7.1: **Genre Distribution in UD Version 2.8.** Ranges indicate upper/lower bounds for sentences per genre inferred from UD metadata. Center marker reflects the distribution under the assumption that genres within treebanks are uniformly distributed. Labels above the bars indicate the number of treebanks which contain each genre.

genre is said to occur. The center line denotes the distribution under the assumption that genres are uniformly distributed within each treebank.

It is clear that *news* and *non-fiction* constitute more than half of the entire dataset. Specialized genres such as *medical* are less represented. For broader genres such as *web*, which frequently co-occurs with others, the exact number of sentences is hard to estimate, but must lie between 0–20%. Considering these large variances, access to instance-level genre will likely be crucial for effective proxy data selection and downstream domain adaptation.

7.3.2 Instance-level Annotations

In addition to the aforementioned 18 treebank-level genre labels, some treebanks provide instance-level genre annotations in the comment-metadata before each sentence. We find such annotations in 26 out of 200 treebanks in UD 2.8 amounting to 124k or 8.25% of all sentences.

Out of this set, 20 treebanks belong to the Parallel Universal Dependencies (PUD; Nivre et al., 2017). They are split 500/500 between *news* and *wiki*, as denoted by sentence IDs beginning with `n` and `w` respectively. The parallel nature of PUD makes it interesting for analyzing cross-lingual genre identification performance. However these two genres only represent a small fraction of non-fiction texts and furthermore, each PUD-treebank is test-split-only. Note also that Polish-PUD as an exception has the metadata labels *news* and *non-fiction*.

The remaining six treebanks for which we were able to identify instance-level genre annotations are Belarusian-HSE (Lyashevskaya et al., 2017), Czech-CAC (Hladká et al., 2008), English-EWT (Silveira et al., 2014), German-LIT (Salomoni, 2019), Polish-LFG (Patejuk and Przepiórkowski, 2018a) and Russian-Taiga (Shavrina and Shapovalova, 2017). They cover a wider set of 12 genres. Annotation schema vary across treebanks and are neither fully compatible amongst each other nor with the 18 UD labels. Approximate mappings can however be drawn thanks to source data documentation by the respective authors (Section 7.4.2).

Further comment-metadata which may guide genre separation within treebanks includes document, paragraph and source identifiers. Again, these are unfortunately not available for all sentences (although coverage of these metadata reaches up to 45%) and their values do not provide further indications about genre adherence.

7.4 Instance Genre from Treebank Labels

From the previous analysis, it is evident that finer-grained genre labels are needed before domain adaptation can be successful across all languages.

Formally, the task of predicting instance-level UD genre can be defined as assigning a set of labels $\mathscr{L} = \{l_0, l_1, \ldots, l_K\}$ (i.e. genres) to all instances x_n of a corpus $\mathscr{X}$ (i.e. UD). The corpus consists of S distinct subsets $\mathscr{X} = \{\mathscr{X}_0 \cup \mathscr{X}_1 \cup \ldots \cup \mathscr{X}_S\}$ (i.e. treebanks) each with a subset of labels $\mathscr{L}_s \subseteq \mathscr{L}$. As no instance-level labels $x_n \rightarrow l$ are available, models must learn this mapping based solely on the subset of labels said to be contained in each data subset $\mathscr{X}_s \rightarrow \mathscr{L}_s$.

7.4.1 Genre Prediction Methods

As instance-level labels are noisy and sparse, we investigate two classification-based and two clustering-based approaches for inferring instance genre labels from the treebank metadata $\mathscr{L}_s$ alone. Building on Müller-Eberstein et al. (2021a), our proposed methods leverage latent genre information in the pre-trained mBERT language model (Devlin et al., 2019).

Boot In order to select proxy training data which matches the genre of an unseen target, Müller-Eberstein et al. (2021a) propose a bootstrapping-based approach to genre classification (Boot). An mBERT-based classifier (Devlin et al., 2019) is initially trained on sentences from single-genre treebanks, corresponding to standard supervised classification. Above a confidence threshold (i.e. softmax probability of 0.99), sentences from treebanks containing a known genre in mixture are bootstrapped as single-genre training data for the next round. After bootstrapping sentences from all known genres, the remaining unclassified instances of any treebank containing a single unknown genre are inferred to be of that last genre. While this method was previously used for targeted data selection, we investigate the degree to which it actually recovers instance-level genre.

Class With approximate classification (Class), we simplify Boot to naively learn instance genre labels from weak supervision. It finetunes the same mBERT MLM with a 18-genre classification layer on the `[CLS]`-token. For single-genre treebanks it is possible to measure the exact cross entropy between the predicted probability and the target (i.e. $x_n \rightarrow l$ with $l \in \mathscr{L}_s$ and $|\mathscr{L}_s| = 1$). For multi-genre treebanks with $|\mathscr{L}_s| > 1$, this is not possible as the gold label is unknown. For the Class approach, each sentence from a k-genre treebank is therefore classified k times — once for each class in $\mathscr{L}_s$.

GMM In addition to classification, we also evaluate two common clustering algorithms. First we investigate whether clusters formed by untuned MLM sentence embeddings (mean over sentence sub-words) represent genre to such a degree that Gaussian Mixture Models can recover the 18 UD genre groups. For monolingual data from five genres, such clusters were shown to be recoverable (Aharoni and Goldberg, 2020). We extend this approach to the 114 language setting of UD.

LDA As all methods so far are to some degree dependent on the pre-trained MLM representations, we also evaluate the recoverability of genre using Latent Dirichlet Allocation (Blei et al., 2003) with lexical features. Feature vectors are constructed using the frequency of character 3–6-grams.

Cluster Labeling Both clustering methods produce 18 groups of sentences from UD, however these will not carry meaningful labels as with classification. While labels could be assigned manually post-hoc by matching representative sentences in each cluster to one of the 18 global UD genres, this process is bound to be subjective and also depends on the annotator to be fluent in most of the 114 languages.

In order to automate this procedure, we propose **GMM+L** and **LDA+L** which combine clustering and classification. Both methods start by clustering each treebank $\mathcal{X}_s$ into the number of genres specified by its metadata (note that standard GMM and LDA cluster all of UD at once, i.e. $\mathcal{X}$).

Next, the mean embedding of each cluster is computed such that they can be compared in a single representational space. Note that this would not be possible using monolingual models as their latent spaces are not as cross-lingually aligned. Analogous to BOOT, single-genre treebanks can then be used as a single-label signal such that the closest cluster from each treebank containing the respective genre can be extracted. Newly identified clusters are added to the pool of single-genre clusters. This process need only be repeated for three rounds before all sentences in UD can be assigned a single label.

Using these four methods, we aim to assign a single genre label to each sentence in UD. By comparing model ablations, we further depart from prior work and explicitly quantify the genre information in MLM embeddings as well as how it manifests within and across treebanks in UD.

7.4.2 Supervised Evaluation

For the 26 treebanks with instance genre labels, we are able to measure standard F1 after applying a mapping from the treebank-specific labels to the 18 global UD genre labels. The mapping was created according to the following criteria.

First, we only allowed treebank-specific genre labels to be mapped to the set of UD genre labels specified in each treebank's metadata.

Second, if possible treebank labels are mapped to UD labels of the same name (e.g. `fiction` → *fiction*) or to the closest subsuming category (e.g. `spoken (prepared)` → *spoken*).

Third, decisions involving subjective uncertainty were based on the label which covers the majority of data sources. E.g., Czech-CAC has the metadata label set {*legal, medical, news, non-fiction, reviews*} and only three types of instance labels (`aw`, `nw`, `sw`). The `sw` (scientific-written) label is attached to many medical articles, but also to articles on philosophy or music. While *academic* may be the most fitting label, it is not in the metadata. As such we chose the broader *non-fiction* as the target label.

The full mapping is in Appendix 7.7.1 and we hope future work will be able to expand upon it.

7.4.3 Unsupervised Evaluation

For the remaining 174 treebanks without sentence-level gold labels it is difficult to measure the exact quality of the predicted genre distributions. Nonetheless, treebank annotations provide enough information for approximate, global comparisons.

Based on label/cluster assignments, it is possible to compute the standard cluster purity measure (PUR; Schütze et al., 2008). Across treebanks of the same genre, the majority of sentences should belong to the same label/cluster. We measure this using the ratio of cross-treebank label agreement (AGR). As in prior work (Aharoni and Goldberg, 2020) it is important to note that the aforementioned metrics can be misleading when taken on their own: A perfect score can for example be achieved by simply assigning all instances to the same genre.

To mitigate this issue we turn to the expected overlap of inter-treebank genre distributions. For multi-genre treebanks, it is known which genres are present, but not how they are distributed. Since treebanks are expected to have a certain amount of overlap, we can however estimate a global error. A {*fiction, spoken, wiki*} treebank should for example have no clusters in common with a {*news*} treebank, but should have many sentences in the same clusters as a {*fiction, medical, spoken*} one. Assuming that genres are uniformly distributed within each treebank, the first pair would share 0 mass between distributions while the second pair would share $\frac{2}{3}$. Intuitively, a good prediction would produce a global genre distribution that falls precisely between the metadata range bars of Figure 7.1, close to the center markers.

To quantify the overlap between two treebank genre distributions p and q over the genres in $\mathscr{L}_s$, we use the discrete Bhattacharyya coefficient:

$$BC(p,q) = \sum_{l \in \mathscr{L}_s} \sqrt{p(l)q(l)} \tag{7.1}$$

which has often been applied to distributional comparisons (Choi and Lee, 2003; Ruder and Plank, 2017). It is computed for all pairs of treebanks such that the overlap error $\Delta\text{BC} \in [0, 100]$ is the mean absolute difference between the expected distributional overlap of each treebank pair and the predicted one (i.e. lower is better).

While none of these metrics can individually provide an exact measure of a prediction method's fit to the UD-specified distribution, they complement each other as to allow for global comparisons in absence of any sentence-level annotations.

7.5 Experiments

7.5.1 Setup

Data From the 1.5 million sentences in UD, we construct global training, development and testing splits. All original test splits are left unchanged and gathered into one global test split containing 204k sentences. Note that test-only treebanks and languages are thereby never seen during training or tuning. For instance-level, supervised evaluation, this means that all PUD treebanks and German-LIT are excluded, leaving five treebanks for tuning.

Next, all original training and development splits are concatenated and split 10/90 into a global training and development split with 102k and 915k sentences respectively. The reason for this small "training" split is that it is only required for training CLASS and BOOT. Within it, we again split the data 70/30 (71k and 31k sentences) for classifier training and held-out data for early stopping. All exact splits are provided in Appendix 7.7.1.

Baselines For our comparisons, we use a maximum frequency baseline (FREQ) which labels all sentences within a treebank with the metadata genre label that is most frequent overall. For example, in any treebank containing *news*, all instances are labeled as such.

In order to measure the untuned classification performance of mBERT, we propose an additional zero-shot classification baseline (ZERO). Prior research has found that classifying sentences based solely on their cosine similarity to genre label strings in MLM embedding space can be remarkably effective (Veeranna et al., 2016; Yin et al., 2019; Davison, 2020). For example, a sentence is labeled as *academic* if this is the closest embedded label out of all 18 genre strings.

Training Every method from Section 7.4.1 is run with three initializations. CLASS and BOOT are trained for a maximum of 30 epochs with an early stopping patience of 3. ZERO, GMM+L and LDA+L (by extension GMM, LDA) do not require training and can be directly applied to the target data. Implementation details and development results are reported in Appendices 7.7.2 and 7.7.3.

7.5.2 Results

Using the 8% subset of annotated instances (Section 7.4.2) in addition to the unsupervised metrics from Section 7.4.3, we can gather an estimate of each method's performance in Table 7.1. UD-level genre predictions in addition to instance-level confusions are further visualized in Figures 7.2 and 7.3.

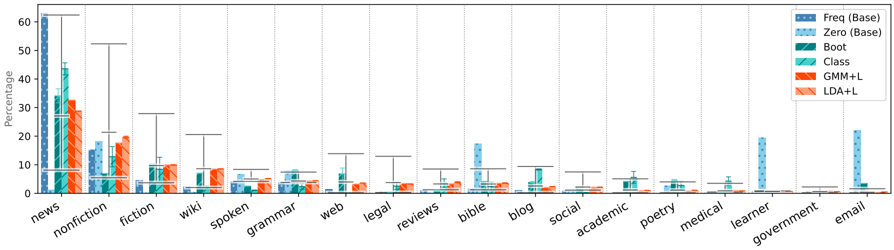


Figure 7.2: **Genre Predictions on UD (Test).** Ranges indicate upper/lower bounds inferred from UD metadata and the distribution under treebank-level uniformity at the center marker. Bars show averaged distribution predictions with standard deviations by Freq, Zero, Boot, Class, GMM+L and LDA+L.

Method	Pur	Agr	ΔBC	F1
Freq	100±0.0	100±0.0	21±0.0	47±0.0
Zero	46±0.0	56±0.0	47±0.0	12±0.0
Class	83±1.4	63±3.9	34±1.1	32±0.9
Boot	86±0.4	70±0.7	29±0.3	38±1.2
GMM	90±0.5	45±2.6	31±0.3	—
+Labels	100±0.0	100±0.0	4±0.2	54±2.1
LDA	77±0.8	34±2.6	31±0.2	—
+Labels	100±0.0	100±0.0	2±0.1	51±1.5

Table 7.1: **Results of Genre Prediction on UD (Test).** Purity (Pur ↑), agreement (Agr ↑), overlap error (ΔBC ↓) and micro-F1 over instance-labeled TBs (F1 ↑) for Freq, Zero, Class, Boot and GMM, LDA with/without cluster label predictions (+Labels). Standard deviation denoted ±.

Baselines The FREQ baseline highlights the issue of using individual unsupervised metrics for estimating performance. As it assigns all sentences per treebank to the same genre, it automatically achieves 100% single-genre treebank purity and agreement. Considering that the instance-level F1 covers 12 genres, a baseline score of 47 is also competitive. Note that this is mostly due to the data imbalance towards *news*. This unlikely distribution predicted by FREQ is also reflected in Figure 7.2.

ZERO-shot classification is not fine-tuned on UD-specific signals and as such predicts a genre distribution that does not adhere to the metadata at all (see Figure 7.2). It severely underpredicts high-frequency genres such as *news* and overpredicts less frequent genres such as *email*. This reflects in our metrics, with ZERO obtaining the lowest PUR, AGR and F1 while having the highest ΔBC of 47.

Classification With regard to explicit genre fine-tuning, CLASS increases purity by 38 points compared to ZERO. Agreement across treebanks also improves, while overlap error decreases. These differences are also reflected in Figure 7.2 in that the predicted distribution is more within the range that would be expected given the metadata.

BOOT fits the UD genre distribution more closely, resulting in a purity that is 4 points higher and agreement that is 11 points higher than CLASS. F1 also increases by 6 points while overlap error decreases by 4 points, indicating that these improvements are not merely due to e.g. assigning all sentences to the same genre. While instance-level F1 is below the FREQ baseline, both methods improve upon the untuned ZERO by a factor of 3.

The benefits of the less noisy training signal are visible in Figure 7.2: Compared to CLASS, BOOT predicts labels in a way that more closely resembles the expected distribution even when the label only occurs in multi-genre treebanks and is ambiguous (e.g. *web*). While BOOT agrees upon the same genre-label across languages (e.g. all *social* treebanks are labeled as such), CLASS tends to overassign the globally most frequent labels (e.g. half of *social* treebanks are labeled *wiki*) and has a larger variance in its assignments across initializations.

Clustering GMM clusters from untuned mBERT embeddings follow the distribution specified by UD metadata more than the LDA clusters produced from lexical information. Although sentence representations are gathered using a naive mean-pooling approach, the resulting clusters reach 90% PUR compared to 77% for LDA. AGR follows a similar pattern and ΔBC is equivalent.

Turning to our cluster labelling approaches, both GMM+L and LDA+L obtain the highest overall F1 scores, outperforming both baselines. They achieve 100% PUR and AGR by the same process as the FREQ-baseline while their overlap error is significantly lower at 4 and 2 points respectively. Figure 7.2 reflects this, as GMM+L and LDA+L are always closest to the expected genre distribution, regardless of overall genre frequency. This shows how focusing on treebank-internal differences before applying a global labelling procedure combines the benefits of local clustering with the benefits of bootstrapped classification, resulting in an effective overall method.

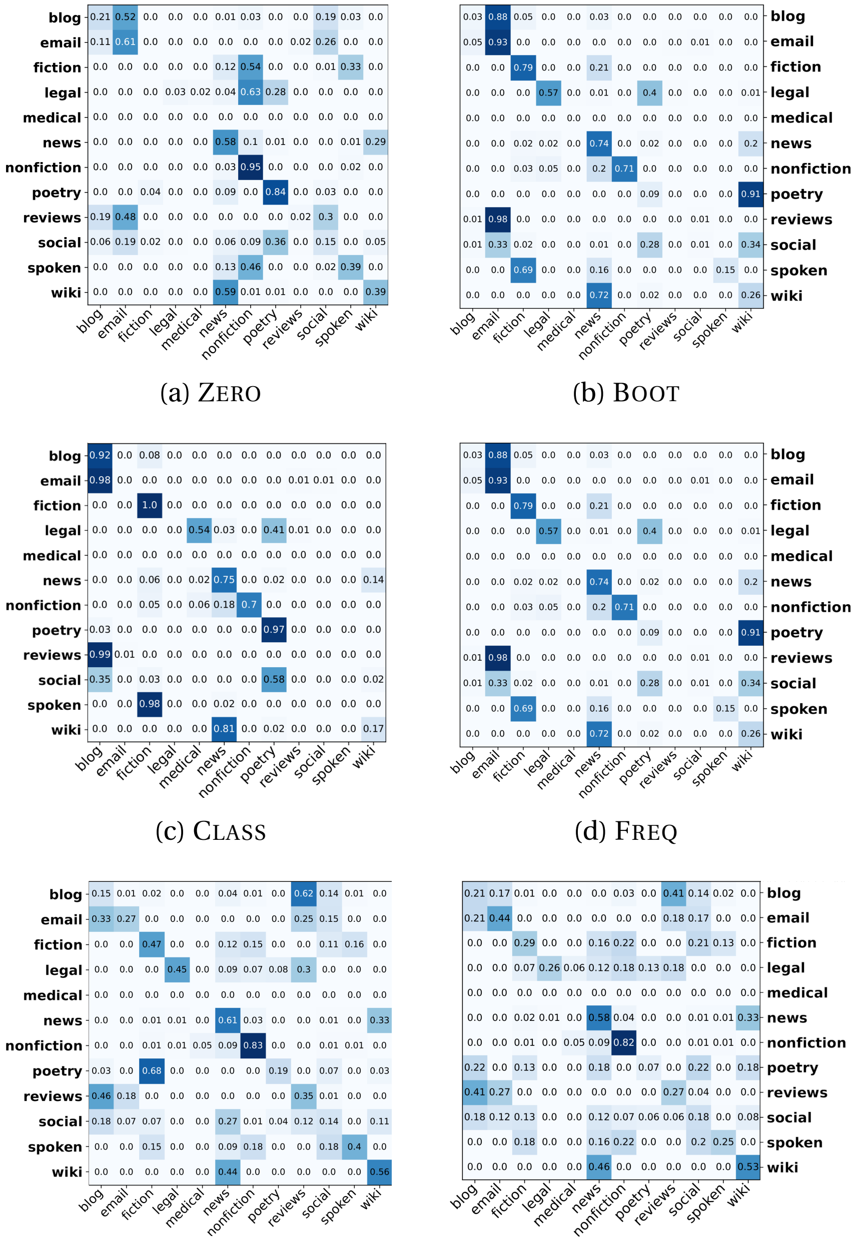
7.5.3 Analysis

From the F1 scores in Table 7.1 it is clear that predicting instance genre based on treebank metadata alone — while accounting for its skewed distribution and inter-treebank shifts of genre definitions — is a difficult task. In the following we analyze the performance characteristics of each method.

Overall, trends of the unsupervised metrics follow the supervised F1, leading us to believe that the methods would behave comparatively should labels for all instances in UD be available. The confusion matrices with prediction ratios per gold label in Figure 7.3 reflect our previous observations.

Baselines The FREQ baseline's predictions are clearly dominated by the most frequent *news* genre, followed by the similarly high frequency *non-fiction* and *blog* (see Figure 7.3d).

ZERO appears to follow a pattern similar to BOOT (e.g. *blog* and *email*), however it also makes more predictions away from the diagonal (see Figure 7.3a).



(a) ZERO

(b) BOOT

(c) CLASS

(d) FREQ

(e) GMM+L

(f) LDA+L

Figure 7.3: **Confusions of Instance-level Genre.** Ratios of predicted labels (columns) per target (row) for ZERO, BOOT, CLASS, FREQ, GMM+L, LDA+L on test splits of 26 instance-annotated treebanks.

Classification Both CLASS (Figure 7.3c) and BOOT (Figure 7.3b) assign most instances of a genre to a single prediction label, often strongly aligning with the target diagonal. CLASS more often assigns a single label per target instead of spreading out predictions across multiple labels as in BOOT. Nonetheless, both methods make some unintuitive errors such as BOOT classifying parts of *poetry* as *wiki*. For these 68 samples from Russian-Taiga, BOOT likely overfits the language signal from Russian-GSD (McDonald et al., 2013; *wiki*).

Compared to ZERO which approximates the predictions of an untuned mBERT model, BOOT and CLASS fine-tuning appears to amplify existing patterns and shifts some predictions to better align with genres as defined in UD (e.g. *fiction* and *legal* in BOOT).

Clustering Grouping all 1.5 million sentences of UD into 18 unlabeled clusters using GMM and LDA results in purity and ΔBC comparable to CLASS and BOOT. However, looking into the cluster contents of the former reveals that they are oversaturated with large treebanks such as German-HDT. Cosine similarities of cluster centroids from the mBERT-based GMM further indicate that proximity corresponds foremost to language similarity.

Some clusters predominantly contain *news*, *wiki* or *social*. This corresponds to cases such as the Italian Twitter treebank TWITTIRÒ in which specific tokens (e.g. “@user”) are distinct enough to override the language signal. Overall, most UD-level clusters do not have clear genre distinctions and are influenced more strongly by language than genre, resulting in high treebank purity while having low intra-treebank agreement. Attempting to cross-lingually cluster all sentences in UD directly is therefore not as effective for recovering instance-level genre as it was in the monolingual setting (Aharoni and Goldberg, 2020).

Initially constructing clusters within each treebank as in the GMM+L and LDA+L methods appears to restore the benefits observed in the monolingual setting. A qualitative analysis of the treebank-level LDA clusters reveals that *wiki* clusters often contain lexical indicators for the genre, such as brackets, while *news* features often contain n-grams which may be related to spoken quotes such as “said”, “Ik␣” (first person pronoun).

Attaching labels to these clusters using the globally shared mBERT space yields confusion plots for GMM+L and LDA+L which most closely follow the diagonal (see Figures 7.3e and 7.3f). Overall, their predictions follow a similar pattern indicating that clustering at the treebank-level using either mBERT embeddings or lexical features results in similar sentence groups.

Within the instance-labeled subset, all models share confusions between *news* and *wiki* (mainly from PUD). While *wiki* is often predicted as *news*, both GMM+L and LDA+L substantially improve upon this "*news*-bias" with a confusion ratio that is 13%–56% lower compared to all other methods. The sentence-bounded context from which all models must make their genre predictions nonetheless limits the amount of improvement possible. For example, using the aforementioned LDA features the algorithm would very likely be unable to distinguish between *news* and *wiki* (both non-fiction, edited texts describing facts) for cases such as, *"Weiss was honored with the literature prizes from the cities of Cologne and Bremen."*

7.6 Discussion and Conclusion

This work provided an in-depth analysis of the 18 genres in Universal Dependencies (UD) and identified challenges for projecting this treebank metadata to the instance level. As these genre labels were not part of the first UD releases, but were added in later versions, we identified large variations in the way they are interpreted and applied — resulting in far less universal definitions of genre than for syntactic dependencies. Most treebanks furthermore contain multiple genres while not providing finer-grained instance-level annotations thereof. This also sheds light on prior work which used UD metadata for training data selection, where treebank-level genre improved in-language parsing performance (Stymne, 2020) and where moving to instance-level genre signals lead to additional increases even across languages (Müller-Eberstein et al., 2021a).

Building on the latent genre information stored in MLM embeddings, we investigated four methods for projecting treebank-level labels to the instance level. In contrast to prior monolingual work, immediately clustering multilingual embeddings yielded clusters dominated by language similarity instead of genre (Section 7.5.3). Similarly, zero-shot labelling using the untuned mBERT latent space proved to be insufficient for producing a genre distribution which adheres to the UD metadata. The classification-based CLASS and BOOT methods are able to extract a stronger genre signal from mBERT than ZERO.

Our proposed GMM+L and LDA+L methods which combine local treebank clusters with the global, cross-lingual representation space reach the best overall performance, outperforming both baselines as well as both classification methods at a much lower computational cost (Section 7.5.2; Appendix 7.7.2). This highlights how the current genre annotations are far from universal, yet can still guide our local-to-global instance-level genre predictors in identifying cross-lingually consistent, data-driven notions of genre.

Future work may be able to improve instance genre prediction by using a more consistent label set or human annotations. The definition of genre macro-classes or a broader taxonomy covering existing annotations could also guide further investigations into cross-lingual language variation. Nonetheless, we expect the task of predicting sentence genre to remain difficult due to the short context within which both annotators and models must make their predictions.

Within the complex scenario of highly cross-lingual, instance-level genre classification, our methods have nonetheless demonstrated that genre is recoverable across the 114 languages in UD — shedding light on prior genre-driven work as well as enabling future research to more deliberately control for additional dimensions of language variation in their data.

7.7 Appendix

7.7.1 Universal Dependencies Setup

All experiments make use of Universal Dependencies v2.8 (Zeman et al., 2021). From the total set of 202 treebanks, we use all except for the following two (due to licensing restrictions): *Arabic-NYUAD* and *Japanese-BCCWJ*. In total 1.51 million sentences are used in our experiments.

Data Splits The experiments in Section 7.5 use the 204k global test split. Initial comparisons were performed on the 915k dev set. The 102k training split was used to fine-tune CLASS and BOOT. For early stopping, 31k sentences from the latter split were used as a held-out set. The exact instances are available in the associated code repository for future reproducibility.

Genre Mapping For 26 treebanks with instance-level genre labels in the metadata comments before each sentence, we created mappings from the treebank genre labels to the UD genre label set according to the guidelines described in Section 7.4.2. The genre metadata typically either follow the format `genre = X` or are implied by the document source specified in the sentence ID (e.g. `sent_id = genre-...`). There are a total of 91 mappings which will be made available with the codebase upon publication.

7.7.2 Model and Training Details

The following describes architecture and training details for all methods. When not further defined, default hyperparameters are used. Implementations and predictions are available in the code repository at `https://personads.me/x/syntaxfest-2021-code`.

Infrastructure Neural models are trained on an NVIDIA A100 GPU with 40 GB of VRAM.

Language Model This work uses mBERT (Devlin et al., 2019) as implemented in the Transformers library (Wolf et al., 2020) as `bert-base-multilingual-cased`. Embeddings are of size $d_{\text{emb}} = 768$ and the model has 178 million parameters. To create sentence embeddings, we use the mean-pooled WordPiece embeddings (Wu et al., 2016) of the final layer.

Classification CLASS and BOOT build on the standard mBERT architecture as follows: mBERT → `CLS`-token → linear layer ($d_{\text{emb}} \times 18$)

→ softmax. The training has an epoch limit of 30 with early stopping after 3 iterations without improvements on the development set. Backpropagation is performed using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-7} on batches of size 16. The fine-tuning procedure requires GPU hardware which can host mBERT, corresponding to 10 GB of VRAM. Training on the 71k relevant instances takes approximately 10 hours.

Clustering Both *Gaussian Mixture Models* (GMM) and *Latent Dirichlet Allocation* (Blei et al., 2003; LDA) use scikit-learn v0.23 (Pedregosa et al., 2011). LDA uses bags of character 3–6-grams which occur in at least 2 and in at most 30% of sentences. GMMs use the mBERT sentence embeddings as input. Both methods are CPU-bound and cluster all treebanks in UD in under 45 minutes.

Random Initializations Each experiment is run thrice using the seeds 41, 42 and 43.

7.7.3 Additional Results

Table 7.2 shows results on the 915k development split of UD. Performance patterns are similar to those on the test split: the labeled clustering methods GMM+L and LDA+L perform best out of our proposed methods and outperform the baselines on the majority of metrics. With respect to classification, BOOT outperforms both the noisier CLASS and ZERO. Note that the frequency baseline FREQ performs especially well on the dev set, since only 5 of 26 instance labeled treebanks are included and 4 of these have the majority genre *news*.

METHOD	PUR	AGR	ΔBC	F1
FREQ	100±0.0	100±0.0	23±0.0	27±0.0
ZERO	43±0.0	66±0.0	50±0.0	5±0.0
CLASS	87±1.2	77±3.9	29±1.9	9±4.5
BOOT	95±0.2	100±0.0	24±0.3	16±1.0
GMM	92±0.1	55±5.5	30±0.7	—
+LABELS	100±0.0	100±0.0	5±0.1	17±1.6
LDA	88±1.0	42±2.2	30±0.2	—
+LABELS	100±0.0	100±0.0	5±0.0	15±0.9

Table 7.2: **Results of Genre Prediction on UD (Dev).** Purity (PUR ↑), agreement (AGR ↑), overlap error (ΔBC ↓) and micro-F1 over instance-labeled TBs (F1 ↑) for FREQ, ZERO, CLASS, BOOT and GMM, LDA with/without labels. Standard deviation denoted ±.

8 Genre as Weak Supervision for Cross-lingual Dependency Parsing

The work presented in this chapter is based on the publication: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021a. Genre as weak supervision for cross-lingual dependency parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4786–4802, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abstract

Recent work has shown that monolingual masked language models learn to represent data-driven notions of language variation which can be used for domain-targeted training data selection. Dataset *genre* labels are already frequently available, yet remain largely unexplored in cross-lingual setups. We harness this genre metadata as a weak supervision signal for targeted data selection in zero-shot dependency parsing. Specifically, we project treebank-level *genre* information to the finer-grained sentence level, with the goal to amplify information implicitly stored in unsupervised contextualized representations. We demonstrate that genre is recoverable from multilingual contextual embeddings and that it provides an effective signal for training data selection in cross-lingual, zero-shot scenarios. For 12 low-resource language treebanks, six of which are test-only, our genre-specific methods significantly outperform competitive baselines as well as recent embedding-based methods for data selection. Moreover, genre-based data selection provides new state-of-the-art results for three of these target languages.

8.1 Introduction

Multilingual masked language models (MLMs) trained on immense quantities of heterogeneous texts (Devlin et al., 2019; Brown et al., 2020; Conneau et al., 2020) have recently made applications such as highly cross-lingual dependency parsing a reality (Kondratyuk and Straka, 2019). Adjacently, it has also been recognized that they capture characteristics relevant for training data selection (Aharoni and Goldberg, 2020) and can be efficiently fine-tuned for higher task-specific performance (Gururangan et al., 2020; Dai et al., 2020; Lauscher et al., 2020; Üstün et al., 2020). These considerations are especially important in computationally restricted environments and when data from the target distribution are unavailable.

Universal Dependencies (Nivre et al., 2020; UD) provides an extensive testing ground for such scenarios: Its language diversity is constantly increasing (from 10 in v1.0 to 104 in v2.7) and low-resource languages are often limited to a single test-set-only treebank. As most

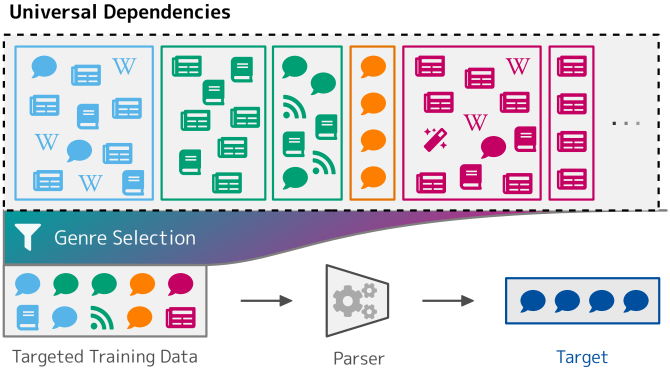


Figure 8.1: **Genre-driven Training Data Selection** for a zero-shot target treebank. In absence of annotated in-language data, we propose *genre* as a weak supervision signal for targeted instance selection from a large pool of out-of-language treebanks.

of the 7,000+ languages in the world similarly lack any annotated training data, effective zero-shot transfer learning is crucial for achieving wider linguistic coverage.

Criteria for selecting training data within such settings vary, and a practitioner may determine relevance by proxy of language relatedness or treebank content. This leads us to the question: If our goal is to develop a parser for a known domain in an unseen language, can a signal such as *genre* guide our selection of cross-lingual training data from a significantly larger, diverse pool (Figure 8.1)?

Within the heterogeneity of written and spoken (transcribed) data, genre broadly encompasses variation along the functional role of a text Kessler et al. (1997). A clear definition is complex if not impossible and communities refer to genre, domain, style or register in different ways Kessler et al. (1997); Lee (2001); Webber (2009); Plank (2011). In this work, we take a pragmatic approach and use genre as defined by the 18 community-provided categories in UD (Zeman et al., 2021). These genres are assigned at the treebank level and "are neither mutually exclusive nor based on homogeneous criteria, but [are] currently the best documentation that can be obtained" (Nivre et al., 2020).

Contributions In order to facilitate finer-grained, instance-level data selection for cross-lingual parsing in absence of in-language training data, we provide three contributions:

First, we provide an analysis of the genre distribution in UD v2.7 (Zeman et al., 2021) across 104 languages and 177 treebanks (Section 8.3).

Next, we introduce three targeted data selection strategies which amplify existing genre information in multilingual contextualized embeddings in order to enable sentence-level selection based on UD's treebank-level genre annotations (Section 8.4).

Finally, we apply the extracted genre information to proxy training data selection for 12 typologically diverse low-resource treebanks. In absence of any in-language training data, our approach outperforms selection using treebank metadata alone as well as purely embedding-based instance selection and surpasses state-of-the-art results on three treebanks (Section 8.5).[1]

8.2 Related Work

Despite advances in zero-shot performance (Devlin et al., 2019; Brown et al., 2020) and increasingly cross-lingual parsers (Kondratyuk and Straka, 2019), fine-tuning has remained a crucial step for achieving state-of-the-art performance. Meechan-Maddon and Nivre (2019) demonstrate that this holds true for low-resource languages in particular, with 200 training instances in the target or related languages producing better results on dependency parsing than a model trained on all available data. Lauscher et al. (2020) further show that as few as 10 samples in the target language can double parsing performance. Üstün et al. (2020) propose UDapters, which integrate language and task-specific adaptation modules into the parser to improve cross-lingual, zero-shot performance.

Considering factors complementary to language is equally important: MLMs can for instance be improved for specific domains such as Twitter or medical texts by fine-tuning on the same or related sources (Dai et al., 2020; Gururangan et al., 2020). For dependency parsing,

[1] Code at https://personads.me/x/emnlp-2021-code.

the use of data from matching genres has been explored by Plank (2011), who find improvements for English and Dutch. This is further confirmed for German by Rehbein and Bildhauer (2017).

Automatically inferred topics (Ruder and Plank, 2017) as well as more abstract selection criteria such as overlapping part-of-speech sequences (Søgaard, 2011; Rosa, 2015) have also proven effective at selecting syntactically similar training instances. Vania et al. (2019) further demonstrate that when word embeddings of mutually unintelligible languages align with respect to POS, cross-lingual transfer remains especially effective. With respect to data-driven domain representations, Stymne (2020) shows that treebank embeddings can be used to successfully transfer knowledge from in-domain cross-lingual source treebanks when used in conjunction with in-language, out-of-domain data. In this work, we will rely solely on treebank genre labels as weak supervision and forgo the use of in-language training data as well as instance-level annotations thereof (e.g. POS tags).

Recently, contextualized embeddings have been shown to contain useful information for training data selection. Aharoni and Goldberg (2020) find that clusters formed by embeddings from untuned, monolingual language models correspond well to the genres of their five-domain corpus. Training an English-to-German machine translation model on only the closest embedded sentences to their target 2k-sentence development set outperformed a model trained on the entire dataset.

Although all aforementioned methods assume some degree of in-language training data, our methods will not have access to any annotated target data and will be trained exclusively on out-of-language instances. Building on information stored in pre-trained contextual embeddings, we extend genre-based data selection into the massively multilingual, 104-language, 18-genre setting of Universal Dependencies (Zeman et al., 2021). While prior work assumed sentence-level genre labels (Ruder and Plank, 2017; Aharoni and Goldberg, 2020), our methods will only have access to treebank-level metadata. An instance's genre will therefore have to be inferred using weakly supervised approaches. To the best of our knowledge, this constitutes the first application of UD's instance-level genre distribution to the selection of training data for zero-shot, cross-lingual dependency parsing.

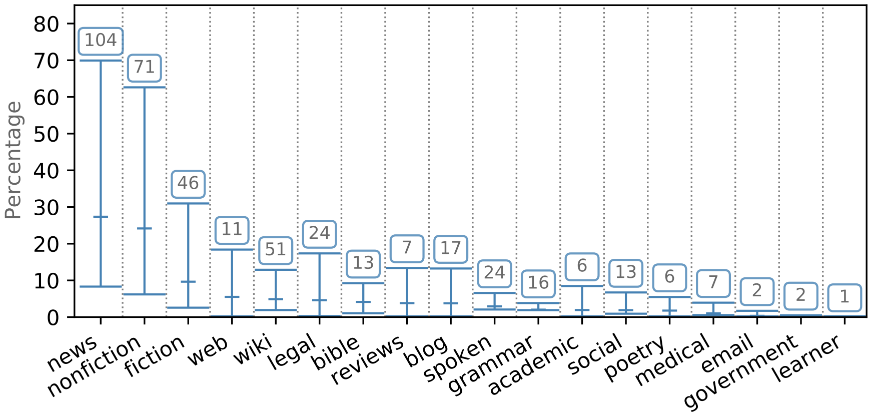


Figure 8.2: **Genre Distribution in UD.** Ranges indicate upper/lower bounds for sentences per genre inferred from UD metadata. Center marker reflects the distribution under the assumption that genres within treebanks are uniformly distributed. Labels above the bars indicate the number of treebanks which contain each genre.

8.3 Genre in Universal Dependencies

Universal Dependencies (Nivre et al., 2016) offer annotations for a broad spectrum of languages, with 104 in version 2.7 Zeman et al. (2021). Of the 1.38 million sentences from the 177 treebanks which we consider, 64 are test-set only and many in this latter third constitute the sole treebank of the language they are in. Such data sparsity becomes even more critical when both the language and the domain are highly specialized and under-resourced.

As more low-resource languages are added in this manner and as the vast majority of the world's languages remain without annotated data, it becomes important to consider new signals for selecting training data in zero-shot scenarios. If no data in the target language are available, we hypothesize that characteristics of most genres are stable enough across languages to offer a useful guiding criterion for data selection in cross-lingual dependency parsing.

For 26 of the 177 treebanks, their authors have provided sentence-level genre labels. However, these annotations cover only 6% of UD sentences and are typically incompatible across treebanks (with few exceptions such as PUD). At the treebank level, UD fortunately provides 18 approximated genre labels: *academic, bible, blog, email, fiction, government, grammar-examples, learner-essays, legal, medical, news, nonfiction, poetry, reviews, social, spoken, web, wiki.*

Genres such as *wiki* likely have stronger internal consistency due to cross-lingual creation guidelines. Others such as *fiction* or *web* may have higher variance. While these UD-provided labels are far from perfectly defined (Nivre et al., 2020), they nonetheless allow us to operationalize our hypothesis: If genre is globally consistent, it must have a positive effect on cross-lingual transfer performance.

From Figure 8.2 it is evident that these genres are heavily imbalanced. The minimum number of sentences in a genre is inferred from the sum over the number of instances in treebanks containing only that genre. The upper bound is the sum of all treebanks containing the genre among others. As indicated by these distributional bounds, news articles may constitute up to 70% of the whole UD dataset. Even assuming uniform genre distributions within each treebank (center marker), over half of all sentences in UD would fall into either the *news* or the *non-fiction* category.

Genres with highly specific lexical and/or structural features such as *spoken*, *social* or *medical* are much more underrepresented. Furthermore, they are often only a small part of larger genre mixtures (117 treebanks include multiple genre-labels). These mixtures, with up to 10 genres in one treebank, may contain related genres (e.g. *news*, *non-fiction*, *web*), but also unrelated ones (e.g. *medical*, *poetry*, *social*, *web*) depending on what data was available to authors during annotation.

Out-of-the-box, treebank-level genre labels appear to be highly noisy (see also Nivre et al., 2020). Additionally, individual treebanks are labeled with multiple genres while lacking such labels at the sentence level. We hypothesize that it is therefore necessary to predict *instance-level* genre distributions before targeted data selection can be effective.

8.4 Targeted Data Selection

In order to measure the effect of genre on the targeted selection of training data, we depart from previous treebank-level selection (Section 8.2) and introduce three new types of instance-level selection strategies in the following section. They are evaluated on the task of zero-shot dependency parsing in Sections 8.5 and 8.6. All of them build on contextualized embeddings learned by the mBERT (Devlin et al., 2019) masked language model (MLM). While MLMs still lack the full breadth of the languages covered in UD (mBERT covers 56 of the 104 languages), they have proven robust in zero-shot scenarios (Devlin et al., 2019; Brown et al., 2020) and have also been found to contain a certain amount of genre information — at least monolingually (Aharoni and Goldberg, 2020; Section 8.2). We evaluate whether UD's definition of genre is also recoverable from these data-driven representations and whether these categories hold cross-lingually.

8.4.1 Closest Sentence Selection

SENT Akin to the strategy used by Aharoni and Goldberg (2020), this SENTENCE-based method attempts to find the most relevant training data by computing the mean embedding of n unannotated target data samples and retrieving the top-k closest non-target instances according to their cosine distance in embedding space. Notable differences from their original method are the use of a much smaller target data sample ($n = 100$ versus $n = 2000$) as well as the use of mBERT instead of English-only BERT embeddings (Devlin et al., 2019) due to our cross-lingual setting.

While the monolingual BERT embeddings were found to represent genre to some degree, such MLM embeddings likely contain many more dimensions of semantic and syntactic information. The SENT method alone is therefore not guaranteed to represent data selection by genre as stronger factors may override these signals. Additionally, Aharoni and Goldberg (2020)'s setup assumed five clearly-defined genres with instance-level annotations while UD has 18 genres with varying degrees of specificity which are only defined in the treebank-level metadata.

8.4.2 Genre Selection

META Separately to MLM embedding-based selection, we evaluate the effectiveness of using the manually assigned genre labels listed in each treebank's metadata. As seen in Section 8.3, these labels can be noisy and have variable interpretations across treebanks. Furthermore, each treebank is assigned up to 10 genres, making instance-level selection as in the previous method impossible.

BOOT To bridge this gap to sentence-level selection, we introduce a bootstrapping procedure which iteratively learns an instance-level classifier for UD genre. Each sentence is encoded through mBERT's CLS token before passing to a classification layer. The model is initialized using standard mBERT weights and begins by training on single-genre treebanks (i.e. standard supervised learning). It then predicts sentence labels for treebanks containing these initial genres. Above a prediction threshold of $0.99 \in [0, 1]$, these are added as new training data for the next round of training. When only one unclassified genre remains in a treebank, all remaining instances are inferred to be of that last genre. Using this procedure, a single genre label is assigned to each sentence in UD within three steps.

Compared to closest sentence selection (SENT), both of the former methods have the added benefit that no target-data is required in order to make the final training data selection. The training corpus simply consists of all instances labelled as belonging to a genre (BOOT) or to a treebank containing the genre in question (META).

8.4.3 Closest Cluster Selection

GMM As shown by Aharoni and Goldberg (2020), monolingual BERT embeddings can be clustered into distinct domains using common clustering algorithms such as Gaussian Mixture Models (GMMs). Using mBERT embeddings, we evaluate whether this holds cross-lingually by clustering each treebank into the number of genres which it is said to contain according to the UD-provided metadata. Deviating from previous work, which only uses these clusters for preliminary analyses, we then use them directly for data selection. By computing a mean embedding for each cluster and choosing the closest one to the mean target sample embedding (same as SENT), the most similar data is selected in bulk from each treebank. By only selecting clusters from treebanks for which the metadata states that the target genre is contained, this allows us to identify clusters which most likely correspond to the target genre while avoiding the manual labelling of clusters across 104 languages.

LDA We also evaluate a clustering method based purely on lexical features (i.e. n-grams) instead of pre-trained contextual embeddings. While the selection of the most relevant cluster from each treebank is performed using the same mean embedding distance methodology as for GMM, we use Latent Dirichlet Allocation (Blei et al., 2003; LDA) for the initial clustering step. This decouples the genre-segmentation step from the multitude of non-genre dimensions in the embeddings themselves, while simultaneously not relying on LDA alone for the final data selection (as in Plank, 2011; Mukherjee et al., 2017). Furthermore, this setup allows us to extract genres from languages and scripts unknown to mBERT as well as to compare whether the GMM clusters correspond to those found via surface-level lexical information alone.

8.5 Experimental Setup

8.5.1 Target Treebanks

We evaluate the effect of genre on training data selection using a set of 12 target treebanks from the low-resource end of UD. For our purposes, low-resource is defined as treebanks with more than 200 and less than 2,000 sentences in total and with fewer than 5,000 in-language sentences in UD.

In order to distinguish the effects of genre specifically, we only use single-genre target treebanks and leave the investigation of genre-mixtures to future work. As seen in Table 8.1, the final set of targets is diverse with respect to genre, language family and their availability during mBERT pre-training.

TARGET	AUTHORS	LANGUAGE	FAMILY	MB	SIZE	GENRE
SWL-SSLC	Östling et al. (2017)	Swedish Sign Language	Signed Language	×	203	💬 spoken
SA-UFAL	Dwivedi and Easha (2017)	Sanskrit	Indo-European	×	230	📕 fiction
KPV-Lattice	Partanen et al. (2018)	Komi Zyrian	Uralic	×	435	📕 fiction
TA-TTB	Ramasamy and Žabokrtský (2012)	Tamil	Dravidian	✓	600	📰 news
GL-TreeGal	Garcia (2016)	Galician	Indo-European	✓	1,000	📰 news
YUE-HK	Wong et al. (2017)	Cantonese	Sino-Tibetan	×	1,004	💬 spoken
CKT-HSE	Tyers and Mishchenkova (2020)	Chukchi	Chukotko-Kamchatkan	×	1,004	💬 spoken
FO-OFT	Tyers et al. (2018)	Faroese	Indo-European	×	1,208	W wiki
TE-MTG	Rama and Vajjala (2017)	Telugu	Dravidian	✓	1,328	🪄 grammar
MYV-JR	Rueter and Tyers (2018)	Erzya	Uralic	×	1,690	📕 fiction
QHE-HIENCS	Bhat et al. (2018)	Hindi-English	Code-Switched	~	1,800	📡 social
QTD-SAGT	Çetinoğlu and Çöltekin (2019)	Turkish-German	Code-Switched	~	1,891	💬 spoken

Table 8.1: **Target Treebanks** with language family (FAMILY), inclusion in mBERT pre-training (MB; included (✓), excluded (×), highly-related languages included (~)), total number of sentences (SIZE) and UD-provided GENRE.

Only three of the target languages are included in mBERT pre-training, with seven not being covered at all and two having strongly related languages in mBERT's repertoire: Hindi-English (QHE) → Hindi, English as well as Turkish-German (QTD) → Turkish, German.

The six included genres cover the high-resource *news* (📰) and *fiction* (📕) as well as the medium resource *wiki* (W) and the lower resource *spoken* (💬), *grammar-examples* (🪄) and *social* (📡).

8.5.2 Data Selection Setup

In order to train parsers for these largely test-only treebanks, we compare seven proxy training data selection strategies for each target (note that only the first strategy uses in-language training data).

TARGET Where available, we use the true target training split as a performance upper bound against which to compare our methods. These are available for the six targets: SWL-SSLC, TA-TTB, GL-TreeGal, TE-MTG, QHE-HIENCS and QTD-SAGT. For three targets without training splits, we make use of proxy in-language data: SA-Vedic (Hellwig et al., 2020) for SA-UFAL, KPV-IKDP (Partanen et al., 2018) for KPV-Lattice and FO-FarPaHC (Ingason et al., 2020) for FO-OFT. For the targets YUE-HK, CKT-HSE and MYV-JR no in-language training data are currently available.

Rand selects a random sample of n_{rand} sentences from the non-target-language UD. We do not restrict this selection to treebanks containing the target genre such that data from a more diverse pool of languages may be selected. To ensure an equivalent comparison, we set n_{rand} to the mean of the number of instances selected by Boot, LDA and GMM (see Appendix 8.8.3 for values of n_{rand}).

Sent selection (see Section 8.4.1) is based on the mean embedding of 100 target sentences and retrieves the top-k closest out-of-language sentences from all of UD independently of genre. Since k needs to be chosen manually, we set it to the number of instances selected by GMM, which is equally dependent on mBERT embeddings.

Meta selects all non-target language treebanks which are denoted to contain the target genre (i.e. both single-genre treebanks as well as mixtures). These data pools make up the largest training corpora in our setup (up to 524k instances for *news*) and also subsume the other genre-based selection methods Boot, LDA and GMM. In this way, it acts as an upper bound in terms of data quantity as well as a baseline for whether treebank-level metadata alone can aid data selection.

Boot selects only the specific instances classified as being in the target genre for use as training data. The classifier is trained according to the bootstrapping method outlined in Section 8.4.2. In order to avoid the memorization of target data, we exclude all data in the target languages from the classifier training process.

GMM clusters each treebank into the number of genres denoted by its metadata using mean-pooled mBERT embeddings for each sentence. Training data is then selected according to the closest-cluster procedure outlined in Section 8.4.3.

LDA works analogously to GMM, but uses LDA to cluster sentences. It uses bags of character 3–6-grams and no language-specific resources (e.g. stop word lists) in order to remain as cross-lingually comparable as possible. Hyperparameters were tuned as outlined in Section 8.5.3.

All methods relying on unannotated target data for the data selection process use 100 random sentences from the target treebank (changes across random initializations). In practical terms, this corresponds to having access to a small amount of target-like data — without gold dependency structures — and selecting the best possible training data for which we do have annotations.

Alternatively, BOOT (as well as META and RAND implicitly) work in a fully zero-shot manner as we only assume knowledge of the intended target genre, but do not assume access to the target sentences *nor* their annotations.

8.5.3 Training Setup

We use the biaffine attention parser Dozat and Manning (2017) implementation of MaChAmp v0.2 (van der Goot et al., 2021b) with default hyperparameters. Each step involving non-deterministic components is rerun using three random seeds.

For efficiency reasons, the seven largest treebanks were subsampled to 20k instances per split. Performance is measured using the labeled attachment scores (LAS) averaged across random initializations. Additionally, we report unlabeled attachment scores (UAS), the number of selected instances as well as the variance across runs in Appendix 8.8.3. Significance is evaluated at $\alpha < 0.05$ using a paired bootstrapped sign test with 10k resampling and Bonferroni correction (Bonferroni, 1936) for the multiple comparisons across random initializations. Appendix 8.8.2 lists all additional hyperparameter settings.

It is important to note that besides the upper bound in-language setup (TARGET), no parser is trained on in-language data. For the tuning of method-specific hyperparameters (LDA features, BOOT thresholds), development sets of the five treebanks containing such splits were used: SWL-SSLC, TA-TTB, TE-MTG, QHE-HIENCS and QTD-SAGT (details in Appendix 8.8.2). During parser training, development data for early stopping is based solely on the out-of-language data selected by each method and not on the in-language target data itself (also excluding constituent languages for code switched targets). Results are reported on each target's test set without any further tuning.

SETUP	SWL 💬	SA 📖	KPV 📖	TA 📰	GL 📰	YUE 💬	CKT 💬	FO W	TE ✎	MYV 📖	QHE 📡	QTD 💬	AVG
TARGET	28.01	15.74	13.36	64.05	80.94	—	—	49.55	83.63	—	62.66	55.04	50.28
RAND	3.67	**24.81**	10.88	50.73	77.65	33.31	15.54	61.88	67.68	20.01	**27.01**	44.57	36.48
SENT	3.55	23.72	13.71	47.93	77.55	35.78	16.44	62.49	68.05	**22.90**	26.46	42.74	36.78
META	6.50	24.29	10.22	50.43	76.63	31.19	11.62	61.23	64.91	20.41	9.42	42.58	34.12
BOOT	5.20	21.80	†21.09	49.43	76.66	†49.85	18.40	†66.25	65.56	19.46	14.75	43.80	37.69
GMM	4.85	22.93	†20.91	†**51.53**	**77.75**	†**49.92**	†**19.81**	†68.25	67.87	20.15	15.09	**45.38**	38.70
LDA	**6.62**	23.70	†**22.27**	49.17	77.01	†49.40	†19.05	†**68.29**	†**68.56**	20.54	15.16	44.72	**38.71**

Table 8.2: **Zero-shot Parsing Results.** LAS for test splits of target treebanks using training data from target/proxy in-language treebanks (TARGET; where available), random sentence selection (RAND), closest sentence selection (SENT), treebanks containing target genre (META), instances classified as target genre (BOOT) and closest cluster selection (GMM and LDA). Scores marked with † significantly outperform TARGET, RAND, SENT and META.

8.6 Results

8.6.1 Zero-shot Parsing Results

As expected, Table 8.2 shows that training the parser on target data (TARGET) results in the best overall performance even though the training corpora for these setups almost never exceed 1k instances. The target treebanks for which in-language data are available, consolidate into a final average of 50.28 LAS. This highlights the overall difficulty of parsing these low-resource treebanks. As the parser is initialized using mBERT, the scores on Tamil (TA), Galician (GL) and Telugu (TE), which are included in its pre-training, are highest overall compared to non-included languages or code-switched variants.

It is noteworthy that when a same-language proxy treebank was used for parser training, scores are lower compared to the other methods. In these three cases, namely Sanskrit (SA), Komi Zyrian (KPV) and Faroese (FO), none of the proxy treebanks include the target's genre which may be a strong contributing factor to this discrepancy.

Turning to our zero-shot setups, META data selection based on treebank-level annotations alone performs worst overall at 34.12 LAS despite constituting the largest training corpora in each setup (see Appendix 8.8.3 for details). Compared to the TARGET upper bounds, it shows how training on two orders of magnitudes more data can still be insufficient if they do not follow the target distribution.

Both RAND and SENT outperform the META baseline at 36.48 and 36.78 LAS respectively. These aggregated scores also highlight that sentence-based selection alone insufficiently captures cross-lingual characteristics as to outperform random chance in most cases.

In contrast, combining latent information in the MLM embeddings with higher-level genre information leads to performance increases not achievable by either method alone. Both GMM and LDA achieve the highest scores across the majority of target treebanks and the highest cross-lingual averages of 38.70 LAS and 38.71 LAS respectively. These scores reflect their similar performance across targets, however we do observe that LDA achieves slightly higher scores on languages which are not included in mBERT pre-training: e.g. Swedish Sign Language (SWL), Sanskrit (SA) and Komi Zyrian (KPV). We hypothesize that this is a result of GMM's dependence on latent information in the mBERT embeddings while LDA constructs clusters independently, based solely on surface-level lexical features (i.e. n-grams).

Finally, amplifying genre information in the mBERT embeddings using our BOOT method also leads to performance increases compared to using untuned embeddings or the coarser grained treebank-level metadata. While it does not entirely reach the performance of the cluster selection methods, its overall average of 37.69 LAS as well as generally similar performance patterns to LDA and GMM lead us to believe that all three methods are picking up on and are amplifying similar latent genre information. As an added benefit, BOOT is able to reach this competitive performance without the need for any target data samples (as opposed to GMM and LDA which use 100 raw samples for cluster selection).

Using our proposed genre-based selection methods we are therefore able to consistently outperform in-language/out-of-genre upper bounds for these low-resource target treebanks. Comparing our results to van der Goot et al. (2021b) who train an identical parser architecture on each UD treebank's respective training split, proxy treebank (for test-only) or all of UD, our methods significantly outperform their best models on five of twelve target treebanks.[2] There are significant increases for both SA-UFAL (16.5 → 23.7 LAS) and KPV-Lattice (11.7 → 22.3 LAS).[3] For the targets YUE-HK (32.7 → 49.9 LAS), CKT-HSE (15.3 → 19.8 LAS) and FO-OFT (62.7 → 68.3 LAS), these scores furthermore constitute — to the best of our knowledge — state-of-the-art results without requiring annotated in-language data.

8.6.2 Analysis of Selected Data

Further analyzing the patterns of data selection allows us to identify some of the reasons behind the differences in performance (visualizations can be found in Appendix 8.8.4).

RAND closely follows the overall data distribution in UD, selecting the most instances from the largest treebanks such as German-HDT (Borges Völker et al., 2019) and selecting none to almost none from low-resource treebanks. SENT follows a similar distribution albeit rarely selecting zero instances from any given language. This behaviour does not change substantially between targets, indicating less targeted data selection.

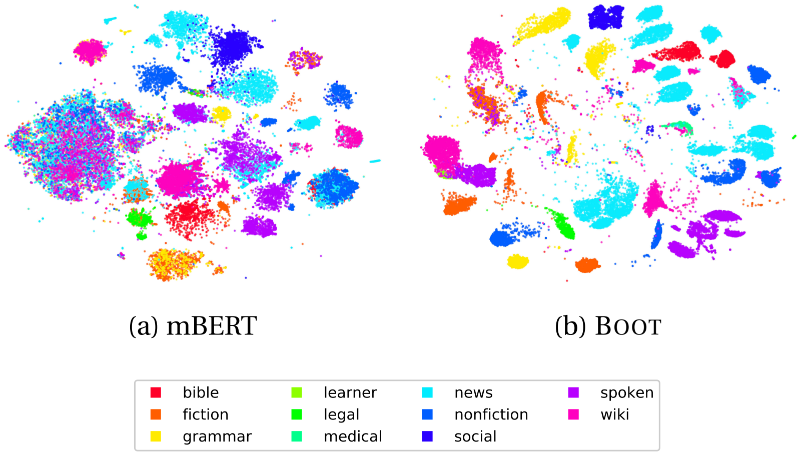
While the larger language diversity of the aforementioned RAND and SENT does not seem to be enough to outperform genre-selection in most cases, it can be helpful when in-genre data is not as linguistically diverse. For the targets SA-UFAL and MYV-JR (*fiction*) both methods outperform genre-based selection by around 2% LAS.

A clear example of insufficient in-genre data is the QHE-HIENCS target. It represents a highly-specialized variation of the *social* genre, specifically Twitter data. Although the genre-based selection methods correctly identify and cluster the Italian Twitter data from IT-PoSTWITA (Sanguinetti et al., 2018) and IT-TWITTIRO (Cignarella et al., 2019), there is a lack of such in-genre data from other languages,[4] leading

[2] We compare against the highest score across all of their proposed models for each treebank.

[3] Dehouck and Denis (2019) achieve higher scores using a parsing architecture with POS and morphological features.

[4] More non-official Twitter-based treebanks in UD style exist (Sanguinetti et al., 2020) which were left out of this study as they are not part of UD and contain annotation divergences.



(a) mBERT

(b) Boot

Figure 8.3: **UD Genres in Embedding Space** of (a) untuned mBERT and (b) genre-tuned Boot. Sentences from single-genre treebanks (up to 1k each) colored by genre, plotted using tSNE (van der Maaten and Hinton, 2008). Tuning using genre as weak supervision clearly amplifies genre information.

these parsers to overfit on Italian specifically. This once again highlights the difficulty of selecting proxy training data which covers all desired characteristics — even from a dataset as diverse as UD.

In general, the genre-driven methods make fairly similar selections given their shared baseline pool of treebanks containing the target genre in-mixture (see Appendix 8.8.4). Since using all of these data however results in the worst overall performance (Meta) while Boot, GMM and LDA perform best, the targeted selection of relevant subsets within the larger Meta pool appears to be key. Frequently, large treebanks such as Polish-LFG (Patejuk and Przepiórkowski, 2018b) with 14k instances from *fiction*, *news*, *nonfiction*, *social* and *spoken* are subsampled to a much smaller fraction (around 3k instances in this example). The fact that these proportions as well as the selected instances themselves are relatively consistent across same-genre targets corroborates that all our methods are picking up on similar, data-driven notions of genre.

Figure 8.3 further visualizes the presence of latent genre using t-

SNE plots of up to 1k randomly sampled sentence embeddings from each of UD's single-genre treebanks. In their untuned state (Figure 8.3a), some local genre clusters do manifest. However, these mainly correspond to specialized treebanks such as the aforementioned Italian Twitter treebanks (*social*). Most other genres occur in language-level mixtures or in a large overall "blob" on the left. By amplifying genre explicitly using the BOOT procedure, each individual genre is much more clearly segmented (Figure 8.3b).

In conclusion, the presence of similar performance patterns across all our proposed genre-driven methods — while having separate approaches to treebank segmentation (weakly supervised tuning for BOOT, treebank-internal embedding distances for GMM, n-grams for LDA) — confirms our hypothesis that instance-level genre can be identified cross-lingually from contextualized representations and aids zero-shot parsing.

8.7 Conclusions

In absence of in-language training data, we have explored UD-specified genre as an alternative signal for data selection. While prior work had indicated the presence of genre information in monolingual contextualized embeddings (Aharoni and Goldberg, 2020), an analogous strategy using mBERT embeddings proved insufficient in the cross-lingual parsing setting (SEN), performing close to the random baseline (RAND). Relying on manual, treebank-level genre labels (META) proved even less performant, producing the lowest scores despite corresponding to a practitioner's typical first choice of selecting the largest number of training instances.

In order to enable finer-grained, instance-level data selection, we proposed three methods for combining latent genre information in the unsupervised contextualized representations with the treebank metadata: weakly supervised BOOT, sentence embedding-based GMM and n-gram-based LDA. Despite their different approaches to treebank segmentation, each method significantly outperformed the purely embedding-based SENT as well as the metadata (META) and random baselines (RAND). Their similar performance patterns and selected data distributions further indicate that each method is identifying a shared, data-driven notion of genre.

For future work, it will be important to extend our proposed approaches beyond single-genre targets towards genre-mixtures and more treebanks overall. As the data selected by these methods is further limited by the number of treebanks in each respective genre, combining a larger set of selection signals will be equally crucial.

8.8 Appendix

8.8.1 Universal Dependencies Setup

All experiments make use of Universal Dependencies v2.7 (Zeman et al., 2021; UD). From the total set of 183 treebanks, we use all except for the following six (due to licensing restrictions): *AR-NYUAD, EN-ESL, EN-GUMReddit, FR-FTB, JA-BCCWJ, GUN-Dooley*. In total 1.38 million sentences are used in our experiments.

Target Treebanks As listed in the main paper, our target treebanks are *Swedish Sign Language-SSLC* (Östling et al., 2017), *Sanskrit-UFAL* (Dwivedi and Easha, 2017), *Komi Zyrian-Lattice* (Partanen et al., 2018), *Tamil-TTB* (Ramasamy and Žabokrtský, 2012), *Galician-TreeGal* (Garcia, 2016), *Cantonese-HK* (Wong et al., 2017), *Chukchi-HSE* (Tyers and Mishchenkova, 2020), *Faroese-OFT* (Tyers et al., 2018), *Telugu-MTG* (Rama and Vajjala, 2017), *Erzya-JR* (Rueter and Tyers, 2018), *Hindi-English-HIENCS* (Bhat et al., 2018) and *Turkish-German-SAGT* (Çetinoğlu and Çöltekin, 2019).

Development Data For the initial tuning of LDA input features as well as the bootstrapping threshold, we used the only five treebanks with development data: *SWL-SSLC, TA-TTB, TE-MTG, QHE-HIENCS, QTD-SAGT*.

For the early stopping of parser training, no such in-language validation data is used (to ensure a pure zero-shot setup). Instead, the data selected by each selection method is split in an 80%/20% fashion and is used as a proxy, out-of-language development split.

Similarly, the training of the bootstrapping classifier (Boot) uses only the non-target-language portion of UD (i.e. excluding all treebanks of the 12 target languages plus constituent languages for code-switched). For efficiency reasons, this data is further subsampled to 40k total instances. Both the training and validation (used for early stopping) of BOOT are therefore similarly conducted without any target-language data.

Subsets Since data selection is at the core of this research, the exact instance IDs of each subset are available in the supplementary code.

8.8.2 Model and Training Details

The following describes architecture and training details for all methods. When not further defined, default hyperparameters are used. Implementations are available in the supplementary code.

Infrastructure Neural models are trained on an NVIDIA A100 GPU with 40 GB of VRAM. Since most of our experiments do not require MLM sentence embeddings to be updated, we compute them once and store them on disk to save GPU cycles.

Multilingual Language Model The MLM used in this work is mBERT (Devlin et al., 2019) as implemented in the Transformers library (Wolf et al., 2020)[5]. Embeddings are of size $d_{\text{emb}} = 768$ and the model itself has 178 million total parameters. To create sentence embeddings in the SENT and GMM methods, we use the mean-pooled WordPiece embeddings (Wu et al., 2016) of the final layer.

Clustering Methods Both *Gaussian Mixture Models* (GMM) and *Latent Dirichlet Allocation* (Blei et al., 2003; LDA) use implementations from scikit-learn v0.23 (Pedregosa et al., 2011). LDA uses bags of character 3–6-grams which occur in at least two and in at most 30% of sentences. The n-gram sizes were initially tuned on target treebanks with available development sets (see Appendix 8.8.1). We found character 1–5-grams to perform approximately 2.5 LAS worse and word unigrams to perform approximately 2 LAS worse than the final method.

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

GMMs use the mBERT sentence embeddings directly as input. Both methods are CPU-bound and complete the clustering of all treebanks in UD in under 45 minutes.

Bootstrapping (BOOT) builds on the standard mBERT architecture as follows: mBERT → CLS → linear layer ($d_{\text{emb}} \times 18$) → softmax. The training has an epoch limit of 100 with early stopping after 3 iterations without improvements on the development set. No target-language data is used during this process. An alternate bootstrapping threshold of 0.9 was evaluated and found to perform approximately 1 LAS worse on the development subset (see Appendix 8.8.1) than the final value of 0.99. Backpropagation is performed using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-7} on batches of size 16. The fine-tuning procedure requires GPU hardware which can host mBERT, corresponding to 10 GB of VRAM. Training on the subsampled 40k instance, non-target-language data takes approximately seven hours.

Dependency Parsers Every parsing experiment in the main paper uses a biaffine attention parser (Dozat and Manning, 2017) implemented in the MaChAmp v0.2 framework (van der Goot et al., 2021b) using default hyperparameters. The sentence encoder is initialized with standard mBERT weights. The training duration is foremost dependent on input data quantity. For the largest corpus (META for TA-TTB with 524k instances) this corresponds to 55 hours. Our proposed methods create smaller, targeted training corpora (around 80k instances on average) such that a better performing parser can be trained in approximately 90 minutes on the same hardware.

Random Initializations Each experiment is run thrice using the seeds 41, 42 and 43. This relates to the random subsampling of data as well as to model initialization (both parsers and selection).

8.8.3 Additional Results

In addition to the labeled attachment scores (LAS) reported in the main paper, we list LAS standard deviation across random initializations in

SETUP	SWL 💬	SA 📕	KPV 📕	TA 📰	GL 📰	YUE 💬	CKT 💬	FO W	TE 🪄	MYV 📕	QHE 📡	QTD 💬	AVG
TARGET	87	3k	132	400	600	—	—	1k	1k	—	1k	285	839
RAND	31k	81k	84k	249k	244k	30k	30k	50k	21k	86k	12k	30k	79k
SENT	33k	95k	101k	271k	236k	31k	30k	58k	23k	113k	14k	31k	86k
META	62k	274k	274k	524k	523k	62k	62k	125k	35k	274k	57k	61k	194k
BOOT	29k	59k	59k	256k	254k	28k	28k	35k	21k	58k	7k	29k	72k
GMM	33k	95k	101k	271k	236k	31k	30k	58k	23k	113k	14k	31k	86k
LDA	32k	89k	95k	238k	233k	33k	33k	56k	21k	96k	14k	30k	81k

Table 8.3: **Training Corpus Sizes** (number of selected instances) for zero-shot parsing experiments from target/proxy in-language treebanks (TARGET; where available), random sentence selection (RAND) and closest sentence selection (SENT), treebanks containing target genre (META), instances classified as target genre (BOOT), closest cluster selection (GMM and LDA).

Table 8.5, unlabeled attachment scores (UAS) in Table 8.4 as well as the number of selected training instances per method in Table 8.3.

Predictions We additionally provide the instance-level predictions of each method and each random initialization as CoNLL-U files in the supplementary material in order ensure that future work can evaluate the statistical significance of performance differences.

8.8.4 Data Selection Analysis

Figure 8.4 displays the distribution of selected instances across all treebanks of UD per target treebank and method. Proportions are normalized to [0, 1] for each method (i.e. across each column). Due to the large number of cells, we recommend viewing this figure digitally.

Setup	SWL 💬	SA 📖	KPV 📖	TA 📰	GL 📰	YUE 💬	CKT 💬	FO w	TE ✎	MYV 📖	QHE 📶	QTD 💬	AVG
Target	40.66	38.74	26.70	75.83	85.51	—	—	58.78	91.26	—	73.62	66.75	61.98
Rand	22.81	**47.06**	25.97	72.14	**84.68**	49.70	29.39	71.66	83.73	36.88	**40.63**	58.97	51.97
Sent	24.47	44.98	31.69	71.28	84.63	51.11	31.95	71.92	83.03	**41.73**	40.19	58.85	52.99
Meta	24.94	44.62	25.77	72.26	84.26	47.91	22.66	70.54	82.06	36.67	19.83	57.93	49.12
Boot	24.83	42.00	39.40	73.38	84.19	**60.72**	35.42	75.21	84.05	39.03	27.59	57.15	53.58
GMM	25.18	44.19	37.77	**74.33**	84.55	60.61	**37.53**	77.00	82.89	38.09	26.65	**59.52**	54.02
LDA	**27.42**	44.84	**40.33**	72.93	84.27	60.06	35.68	**77.23**	**84.70**	38.78	27.61	58.46	**54.36**

Table 8.4: **Unlabeled Attachment Scores** for zero-shot parsing experiments on test splits of target treebanks using training data from from target/proxy in-language treebanks (Target; where available), random sentence selection (Rand) and closest sentence selection (Sent), treebanks containing target genre (Meta), instances classified as target genre (Boot), closest cluster selection (GMM and LDA).

Setup	SWL 💬	SA 📖	KPV 📖	TA 📰	GL 📰	YUE 💬	CKT 💬	FO w	TE ✎	MYV 📖	QHE 📶	QTD 💬	AVG
Target	0.71	0.54	0.77	1.16	0.24	—	—	1.32	0.97	—	0.26	1.10	0.79
Rand	1.60	0.46	0.16	0.72	0.09	1.33	0.89	1.02	0.64	1.09	0.55	0.55	0.76
Sent	2.13	2.00	0.58	1.76	0.18	0.67	0.27	0.63	0.92	0.37	0.37	0.91	0.90
Meta	0.90	0.75	0.73	1.24	0.27	0.41	1.19	0.82	0.42	0.44	0.44	0.73	0.73
Boot	0.54	0.85	0.55	1.07	0.27	0.14	0.51	0.92	0.42	0.28	1.08	0.43	0.59
GMM	1.14	1.02	0.75	1.00	0.18	0.28	0.80	1.30	1.35	1.28	0.63	0.47	0.85
LDA	0.74	2.29	0.23	1.96	0.14	0.65	1.32	0.41	0.44	0.81	1.23	0.25	0.87

Table 8.5: **Standard Deviations of LAS** for zero-shot parsing experiments on test splits of target treebanks using training data from from target/proxy in-language treebanks (Target; where available), random sentence selection (Rand) and closest sentence selection (Sent), treebanks containing target genre (Meta), instances classified as target genre (Boot), closest cluster selection (GMM and LDA).

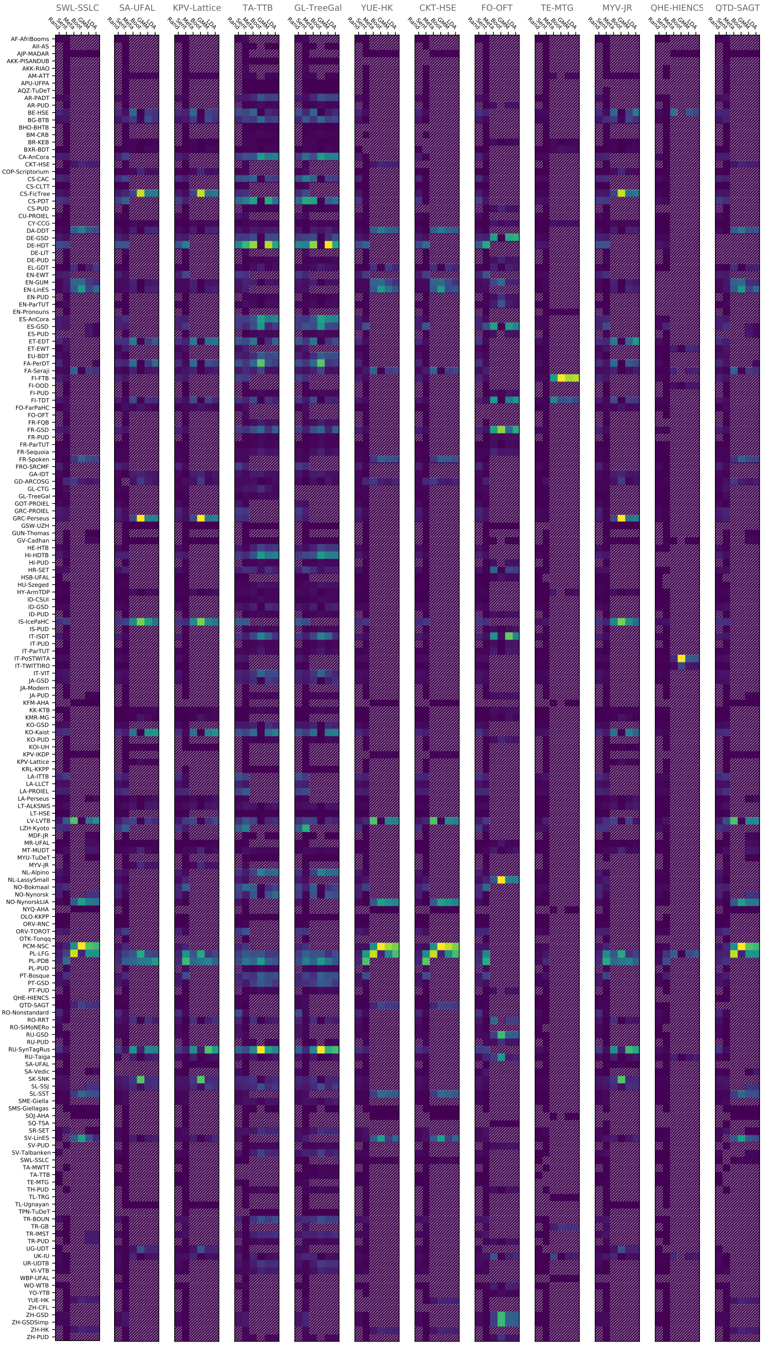


Figure 8.4: **Selection Proportions** per target treebank and data selection method across all of UD. Zero instances were selected from shaded regions.

Part IV

Task Variation

9 Subspace Chronicles

The work presented in this chapter is based on the publication: Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208, Singapore. Association for Computational Linguistics.

Abstract

Representational spaces learned via language modeling are fundamental to Natural Language Processing (NLP), however there has been limited understanding regarding *how* and *when* during training various types of linguistic information emerge and interact. Leveraging a novel information theoretic probing suite, which enables direct comparisons of not just task performance, but their representational subspaces, we analyze nine tasks covering syntax, semantics and reasoning, across 2M pre-training steps and five seeds. We identify critical learning phases across tasks *and* time, during which subspaces emerge, share information, and later disentangle to specialize. Across these phases, syntactic knowledge is acquired rapidly after 0.5% of full training. Continued performance improvements primarily stem from the acquisition of open-domain knowledge, while semantics and reasoning tasks benefit from later boosts to long-range contextualization and higher specialization. Measuring cross-task similarity further reveals that linguistically related tasks share information throughout training, and do so more during the critical phase of learning than before or after. Our findings have implications for model interpretability, multi-task learning, and learning from limited data.

9.1 Introduction

Contemporary advances in NLP are built on the representational power of latent embedding spaces learned by self-supervised language models (LMs). At their core, these approaches are built on the distributional hypothesis (Harris, 1954; Firth, 1957), for which the effects of scale have been implicitly and explicitly studied via the community's use of increasingly large models and datasets (Teehan et al., 2022; Wei et al., 2022). The learning dynamics by which these capabilities emerge during LM pre-training have, however, remained largely understudied. Understanding *how* and *when* the LM training objective begins to encode information that is relevant to downstream tasks is crucial, as this informs the limits of what can be learned using current objectives.

For identifying task-relevant information in LMs, probing has become an important tool (Adi et al., 2017; Conneau et al., 2018a; Giu-

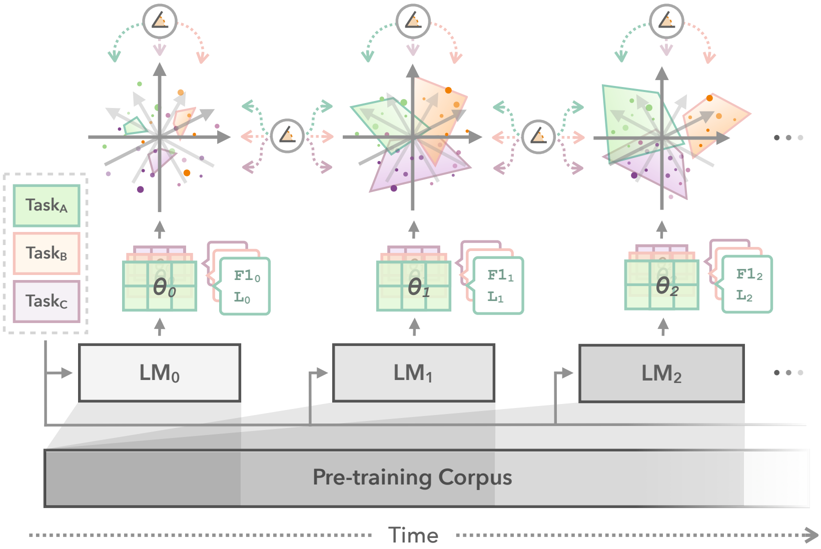


Figure 9.1: **Subspace Chronicles** via probes θ across LM training time, as measured by F1, codelength L, and subspace angles across tasks and time.

lianelli et al., 2018; Rogers et al., 2020), which has already shown promise for revealing LM learning dynamics (Saphra and Lopez, 2019; Chiang et al., 2020; Liu et al., 2021a). The predominant approach of quantifying linguistic information via task performance, however, misses important interactions at the representational level, i.e., whether actual linguistic information is present at any given training timestep (Hewitt and Liang, 2019; Voita and Titov, 2020), and how these representational subspaces change independently of model performance. Furthermore, performance alone does not indicate how different tasks and their related linguistic information interact with each other during training, and how much information they actually share.

By leveraging information-theoretic probes as characterizations of task-specific subspaces within an LM's overall embedding space (Figure 9.1), we aim to answer these questions, and contribute:

- An information theoretic probing suite for extracting not just performance, but entire task-specific representational subspaces within an LM's embedding space, allowing us to measure changes to linguistic information over time *and* across tasks (Section 9.3).[1]

- A study of task subspace emergence, shifts and interactions across nine diverse linguistic tasks, 2M pre-training steps and five random initializations (Section 9.5).
- An analysis of these learning dynamics, which focuses on practical implications beyond performance to identify what can be learned given limited data, and how to effectively leverage this knowledge (Section 9.6).

9.2 Related Work

Although prior work has not specifically focused on the emergence of representational spaces, but rather on task performance, there has been an increased interest in the emergent capabilities of LMs. The community-wide trend of increasing model and dataset size has shown that certain skills (e.g., arithmetic) are linked to scale (Wei et al., 2022; Schaeffer et al., 2023), however Teehan et al. (2022) simultaneously identify a lack of work investigating skill emergence across the training time dimension.

To promote research in this direction, some projects have released intermediate training checkpoints. This notably includes MultiBERTs (Sellam et al., 2022) which studies cross-initialization variation in BERT (Devlin et al., 2019), as well as Mistral (Mistral, 2022), Pythia (Biderman et al., 2023) and BLOOM (Scao et al., 2023) which release checkpoints across training steps, seeds and/or pre-processing procedures.

In seminal work investigating learning dynamics, Saphra and Lopez (2019) measure how strongly hidden representations of a BiLSTM LM correlate with parts-of-speech (PoS), semantic tags and topic information. By probing these characteristics across LM training time, they identify that PoS is acquired earlier than topics. For Transformer models, Chiang et al. (2020) and Liu et al. (2021a) probe intermediate checkpoints of ALBERT (Lan et al., 2020) and RoBERTa (Zhuang et al.,

[1] Code at https://github.com/mainlp/subspace-chronicles.

2021) respectively. They similarly identify that performance on syntactic tasks increases faster than on world knowledge or reasoning tasks, and that this pattern holds across pre-training corpora from different domains. This points towards consistent LM learning dynamics, however using only performance or correlations, it is difficult to interpret how the representation of this knowledge changes over time as well as how it overlaps across tasks.

Similar issues have arisen in single-checkpoint probing where seemingly task-specific information is identified even in random models, requiring control tasks (Hewitt and Liang, 2019), and causal interventions, such as masking the information required to solve a task (Lasri et al., 2022; Hanna et al., 2023). By using the rate of a model's compression of linguistic information, Voita and Titov (2020) alternatively propose an information theoretic measure to quantify the consistency with which task-relevant knowledge is encoded.

Another limitation of prior learning dynamics work is the inability to compare similarities *across* tasks—i.e., whether information is being shared. Aligned datasets, where the same input is annotated with different labels, could be used to measure performance differences across tasks in a more controlled manner, however they are only available for a limited set of tasks, and do not provide a direct, quantitative measure of task similarity.

By combining advances in information-theoretic probing with measures for subspace geometry, we propose a probing suite aimed at quantifying task-specific linguistic information, which allows for subspace comparisons across time as well as across tasks, without the need for aligned datasets.

9.3 Emergent Subspaces

To compare linguistic subspaces over time, we require snapshots of an LM across its training process (Section 9.3.1), a well-grounded probing procedure (Section 9.3.2), and measures for comparing the resulting subspaces (Section 9.3.3).

9.3.1 Encoders Across Time

For intermediate LM checkpoints, we make use of the 145 models published by Sellam et al. (2022) as part of the MultiBERTs project. They not only cover 2M training steps—double that of the original BERT (Devlin et al., 2019)—but also cover five seeds, allowing us to verify findings across multiple initializations. The earliest available trained checkpoint is at 20k steps. Since our initial experiments showed that performance on many tasks had already stabilized at this point, we additionally train and release our own, more granular checkpoints for the early training stages between step 0–20k, for which we ensure alignment with the later, official checkpoints (details in Section 9.4.1).

9.3.2 Probing for Subspaces

Shifting our perspective from using a probe to measure task performance to the probe itself representing a task-specific subspace is key to enabling cross-task comparisons. In essence, a probe characterizes a subspace within which task-relevant information is particularly salient. For extracting a c-dimensional subspace (with c being the number of classes in a linguistic task) from the d-dimensional general LM embedding space, we propose a modified linear probe $\theta \in \mathbb{R}^{d \times c}$ which learns to interpolate representations across the l LM layers using a layer weighting $\alpha \in \mathbb{R}^{l}$ (Tenney et al., 2019a). By training the probe to classify the data $(x, y) \in \mathscr{D}$, the resulting θ thus corresponds to the linear subspace within the overall LM which maximizes task-relevant information.

To measure the presence of task-specific information encoded by an LM, it is common to use the accuracy of a probe predicting a related linguistic property. However, accuracy fails to capture the amount of effort required by the probe to map the original embeddings into the task's label space and achieve the relevant level of performance. Intuitively, representations containing consistent and salient task information are easier to group by their class, resulting in high performance while requiring low probe complexity (and/or less training data). Mapping random representations with no class-wise consistency to their labels on the other hand requires more effort, resulting in higher probe complexity and requiring more data.

Information-theoretic probing (Voita and Titov, 2020) quantifies this intuition by incorporating the notion of probe complexity into the measure of task-relevant information. This is achieved by recasting the problem of training a probing classifier as learning to transmit all $(x, y) \in \mathcal{D}$ in as few bits as possible. In other words, it replaces probe accuracy with codelength L, which is the combination of the probe's quality of fit to the data as measured by $p_\theta(y|x)$ over $\mathcal{D}$, and the cost of transmitting the probe θ itself.

The variational formulation of information-theoretic probing, which we use in our work, measures codelength by the bits-back compression algorithm Honkela and Valpola (2004). It is given by the evidence lower-bound:

$$\begin{aligned} L = &- \mathbb{E}_{\theta \sim \beta}\left[\sum_{x,y \in \mathcal{D}} \log_2 p_\theta(y|x)\right] \\ &+ \mathrm{KL}(\beta || \gamma) \ , \end{aligned} \tag{9.1}$$

where the cost of transmitting the probe corresponds to the Kullback-Leibler divergence between the posterior distribution of the probe parameters $\theta \sim \beta$ and a prior γ. The KL divergence term quantifies both the complexity of the probe with respect to the prior, as well as regularizes the training process towards a probe which achieves high performance, while being as close to the prior as possible. Following Voita and Titov (2020), we use a sparsity-inducing prior for γ, such that the resulting θ is as small of a transformation as possible.

In contrast to measuring task information via performance alone, codelength L allows us to detect how consistently linguistic information is encoded by the LM (i.e., the shorter the better). At the same time, the linear transformation θ becomes an efficient characterization of the task-specific representational subspace. To further ground the amount of learned information against random representations, we use the probes of the randomly initialized models at checkpoint 0 as control values.

9.3.3 Subspace Comparisons

As each probe θ characterizes a task-specific subspace within the overall embedding space, comparisons between subspaces correspond to measuring the amount of information relevant to both tasks. To ensure that the subspaces extracted by our probing procedure are fully geometrically comparable, they are deliberately linear. Nonetheless, multiple factors must be considered to ensure accurate comparisons: First, matrices of the same dimensionality may have different rank, should one subspace be easier to encode than another. Similarly, one matrix may simply be a scaled or rotated version of another. Correlating representations using, e.g., Singular Vector or Projection Weighted Canonical Correlation Analysis (Raghu et al., 2017; Morcos et al., 2018) further assumes the underlying inputs to be the same such that only the effect of representational changes is measured. When comparing across datasets with different inputs x and different labels y, this is no longer given.

To fulfill the above requirements, we make use of Principal Subspace Angles (SSAs; Knyazev and Argentati, 2002). The measure is closely related to Grassmann distance (Hamm and Lee, 2008) which has been used to, e.g., compare low-rank adaptation matrices (Hu et al., 2022). It allows us to compare task-specific subspaces θ in their entirety and independently of individual instances, removing the requirement of matching x across tasks. The distance between two subspaces $\theta_A \in \mathbb{R}^{d \times p}$ and $\theta_B \in \mathbb{R}^{d \times q}$ intuitively corresponds to the amount of 'energy' required to map one to the other. Using the orthonormal bases $Q_A = \text{orth}(\theta_A)$ and $Q_B = \text{orth}(\theta_B)$ of each subspace to compute the transformation magnitudes $M = Q_A^T Q_B$ furthermore ensures linear invariance. The final distance is obtained by converting M's singular values $U\Sigma V^T = \text{SVD}(M)$ into angles between 0° and 90° (i.e., similar/dissimilar):

$$\text{SSA}(A, B) = \arccos(\text{diag}(\Sigma)) \; . \tag{9.2}$$

We use SSAs to quantify the similarity between subspaces of the same task across time, as well as to compare subspaces of different tasks.

9.4 Experiment Setup

9.4.1 Early Pre-training

Model MultiBERTs (Sellam et al., 2022) cover 2M training steps, however our initial experiments showed that the earliest model at step 20k already contains close-to-final amounts of information for many tasks (see Section 9.5). This was even more pronounced for the early checkpoints of the larger LMs mentioned in Section 9.2. As such we train our own early checkpoints starting from the five MultiBERTs initializations at step 0. By saving 29 additional checkpoints up to step 20k, we aim to analyze when critical knowledge begins to be acquired during early training. To verify that trajectories match those of the official checkpoints, we train up to step 40k and compare results. Furthermore, we measure whether the subspace angles between the original and additional models are within the bounds expected for models sharing the same random initialization.

Data BERT (Devlin et al., 2019) is reportedly trained on 2.5B tokens from English Wikipedia (Wikimedia, 2022) and 800M tokens from the BookCorpus (Zhu et al., 2015). As the exact data are unavailable, Sellam et al. (2022) use an alternative corpus by Turc et al. (2019), which aims to reproduce the original pre-training data. Unfortunately, the latter is also not publicly available. Therefore, we gather a similar corpus using fully public versions of both corpora from the HuggingFace Dataset Hub (Lhoest et al., 2021). We further provide scripts to re-create the exact data ordering, sentence pairing and subword masking to ensure that both our LMs and future work can use the same data instances in exactly the same order. Note however that our new checkpoints will have observed similar, but slightly different data (Appendix 9.8.1).

Training Given the generated data order, the new LMs are trained according to Sellam et al. (2022), using the masked language modeling (MLM) and next sentence prediction (NSP) objectives. One update step consists of a batch with 256 sentence pairs for NSP which have been shuffled to be 50% correct/incorrect, and within which 15% of subword tokens (but 80 at most) are masked or replaced. For the optimizer, learning rate schedule and dropout, we match the hyperparameters of the original work (details in Appendix 9.8.2). Even across five initializations, the environmental impact of these LM training runs is low, as we re-use the majority of checkpoints from MultiBERTs beyond step 20k.

Pre-training Results For LM training, we observe a consistent decrease in MLM and NSP loss (see Appendix 9.8.3). As our models and MultiBERTs are not trained on the exact same data, we find that SSAs between our models and the originals are around 60°, which is within the bounds of the angles we observe within the same training run in Figure 9.4 of Section 9.5.2. This has no effect on our later analyses which rely on checkpoints from within a training cohort. Surprisingly, we further find that although these models start from the same initialization, but are trained on different data, they are actually more similar to each other than models trained on the same data, but with different seeds. SSAs for checkpoints from the same timestep and training run, but across different seeds consistently measure >80° (Figure 9.13 in Appendix 9.8.3), highlighting that initial conditions have a stronger effect on representation learning than minor differences in the training data. Finally, our experiments in Section 9.5 show that our reproduced checkpoints align remarkably well with the official checkpoints, and we observe a continuation of this trajectory for the overlapping models between steps 20k and 40k.

9.4.2 Probing Suite

Given the 29 original MultiBERTs in addition to our own 29 early checkpoints, both covering five initializations, we analyze a total of 290 models using the methodology from Section 9.3. In order to cover a broad range of tasks across the linguistic hierarchy, we extract subspaces from nine datasets analyzing the following characteristics: parts-of-speech (PoS), named entities (NER) and coreference (COREF) from OntoNotes 5.0 (Pradhan et al., 2013), syntactic dependencies (DEP) from the English Web Treebank (Silveira et al., 2014), semantic tags (SEM) from the Parallel Meaning Bank (Abzianidze et al., 2017), TOPIC from the 20 Newsgroups corpus (Lang, 1995), sentiment (SENTI) from the binarized Stanford Sentiment Treebank (Socher et al., 2013), extractive question answering (QA) from the Stanford Question Answering Dataset (Rajpurkar et al., 2016), and natural language inference (NLI) from the Stanford Natural Language Inference Dataset (Bowman et al., 2015). Each task is probed and evaluated at the token-level (Saphra and Lopez, 2019) to measure the amount of task-relevant information within each contextualized embedding. In Appendix 9.8.1 we further provide dataset and pre-processing details.

For each task, we train an information theoretic probe for a maximum of 30 epochs using the hyperparameters from Voita and Titov (2020). This results in 2,610 total probing runs for which we measure subword-level macro-F1, and that, more importantly, yield task subspaces for which we measure codelength, layer weights, and SSAs across tasks and time (setup details in Appendix 9.8.2).

9.5 Results

Across the 2,610 probing runs, we analyze subspace emergence (Section 9.5.1), shifts (Section 9.5.2), and their interactions across tasks (Section 9.5.3). In the following figures, results are split between the official MultiBERTs and our own early pre-training checkpoints, which, as mentioned in Section 9.4.1, follow an overlapping and consistent trajectory, and have high representational similarity. Standard deviations are reported across random initializations, where we generally observe only minor differences in terms of performance.

9.5.1 Subspace Emergence

Starting from macro-F1 performance, Figure 9.2 shows clear learning phases with a steep increase between 1k and 10k update steps, followed by shallower growth until the end. Within subsets of tasks, we observe subtle differences: PoS and Topic appear to converge (i.e., >90% of final F1) within the 10k range. Sem follows a similar pattern, but has higher variance due to its many smaller classes. Dep and Coref, also gain most in the 1k–10k phase, but continue to slowly climb until converging later at around 100k steps. NER shares this slow incline and sees a continued increase even after 100k steps. Senti

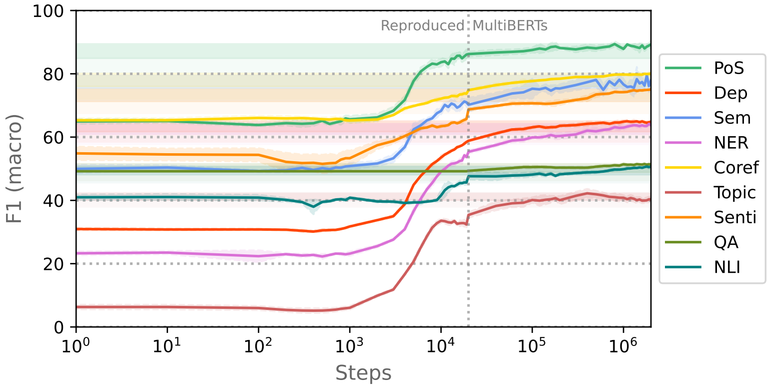


Figure 9.2: **F1 (macro) over LM Training Time** on each task's dev split (standard deviation across seeds). Dark/light shaded areas indicate 95%/90% of maximum performance. Reproduced checkpoints until 19k, MultiBERTs from 20k to 2M steps.

and NLI also have early growth and see another small boost after 100k steps which the other tasks do not. Finally, QA also improves after 100k steps, however results are mostly around the 50% random baseline, as the linear probe is unable to solve this more complex task accurately. These results already suggest task groupings with different learning dynamics, however with F1 alone, it is difficult to understand interactions of the underlying linguistic knowledge. Furthermore, even the random models at step 0 reach non-trivial scores (e.g., PoS and Coref with >60% F1), as probes likely memorize random, but persistent, non-contextualized embeddings similarly to a majority baseline. This highlights the challenge of isolating task-specific knowledge using performance alone.

Turning to codelength in Figure 9.3, we measure the actual amount of learned information as grounded by the level of compression with respect to the random intialization. This measure confirms the dynamics from the performance graphs, however subspaces also continue changing to a larger degree than their performance counterparts suggest. Even after the 10k step critical learning phase PoS information continues to be compressed more efficiently despite F1 convergence. In addition, codelength provides more nuance for Coref, QA and NLI, which see little performance gains, but for which subspaces are

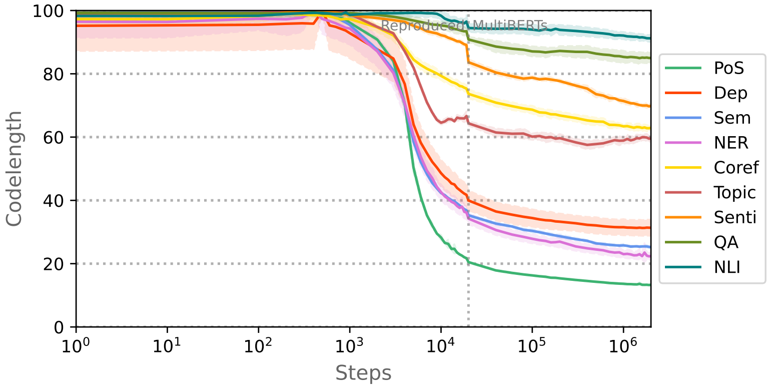


Figure 9.3: **Codelength Ratio over LM Training Time** as percentage of bits required to encode model and data with respect to the random model (standard deviation across seeds). Lower codelength corresponds to higher compression and more task-relevant information.

continuously changing and improving in terms of compression rate.

9.5.2 Representational Shifts

Both performance and codelength indicate distinct learning phases as well as continual changes to linguistic subspaces. Figure 9.4 plots these shifts by measuring subspace shifts as a function of SSAs from one timestep to the next. We observe particularly large updates after the first 100 steps, followed by smaller changes until the beginning of the steep learning phase at around 1k steps. During the large improvements between 1k and 10k steps, subspaces shift at a steady rate of around 45°, followed by a decrease to 5–20° at the end of training. Once again, subspaces are shifting across the entire training process despite F1 suggesting otherwise. Note that while learning rate scheduling can have effects on the degree of change, SSAs do not strictly adhere to it, exhibiting larger differences while the learning rate is low and vice versa. Figure 9.13 in Appendix 9.8.3 additionally shows how SSAs across checkpoints at the same timestep, but across different random seeds are consistently greater than 80°, indicating high dissimilarity. Compared to models starting from the same initialization, even if trained on slightly different data (i.e., reproduced and origi-

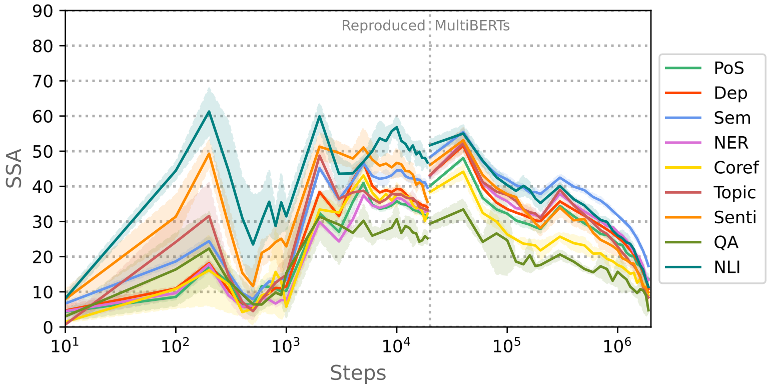


Figure 9.4: **Step-wise SSAs between Probes** indicating the degree of subspace change per update (standard deviation across seeds). Subspaces between original and reproduced checkpoints at step 20k are not comparable.

nal), these angles indicate task subspaces' higher sensitivity to initial representational conditions than to training data differences.

Another important dimension for subspaces in Transformer-based LMs is layer depth. We plot layer weighting across time in Figure 9.5 using the center of gravity (CoG) measure from Tenney et al. (2019a), defined as $\sum_{i=0}^{l} \alpha_i i$. It summarizes the depth at which the probe finds the most salient amounts of task-relevant information. Appendix 9.8.3 further shows the full layer-wise weights across time in greater detail. These weightings shed further light onto the underlying subspace shifts which surface in the other measures. Note that while SSAs across random initializations were large, variability of CoG is low. This indicates that the layer depth at which task information is most salient can be consistent, while the way it is represented within the layers can vary substantially.

CoG at step 0 essentially corresponds to the contextualization level of a task; memorizing non-contextualized embeddings at layer 0 or leveraging some random, but consistent mixing in the later layers. At the start of learning between 100 and 1k steps, all task subspaces dip into the lower half of the model. Together with the previously observed high initial subspace shift, this indicates that, beginning from the non-contextual embeddings in layer 0, contextualization becomes

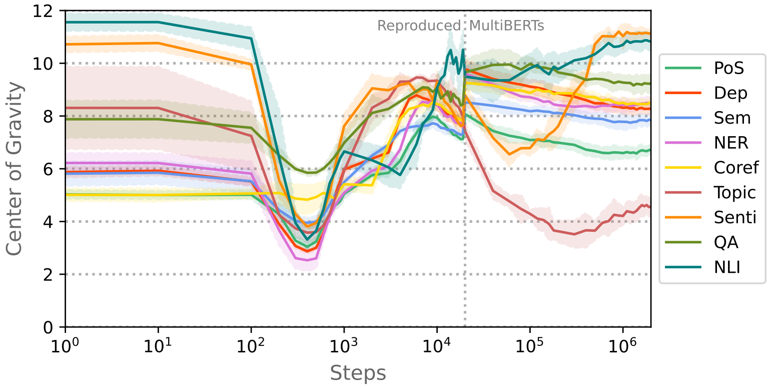


Figure 9.5: **Center of Gravity over Layers** measured following Tenney et al. (2019a), across LM training time (standard deviation across seeds). For detailed weightings of all layers, refer to Appendix 9.8.3.

increasingly useful from the bottom up. Paralleling the steep improvements of the other measures, CoG similarly climbs up throughout the model until the end of the critical learning phase in the 10k range. Until 500k steps, the layers for different tasks disentangle and stabilize. At this point syntactic and lower-level semantic tasks converge towards the middle layers. This specialization is especially prominent for Topic moving to the lower layers, while Senti and NLI (and to a lesser degree QA), which require more complex intra-sentence interactions, move towards the higher layers. Recall that codelength and performance for Senti, NLI and QA also improve around the same time. These dynamics show that the 'traditional NLP pipeline' in LMs (Tenney et al., 2019a) actually emerges quite late, and that tasks appear to share layers for a large duration of LM training before specializing later on. Given that LMs are frequently undertrained (Hoffmann et al., 2022), these continual shifts to task-specific subspaces highlight that probing single checkpoints from specific timesteps may miss important dynamics regarding how tasks are represented with respect to each other.

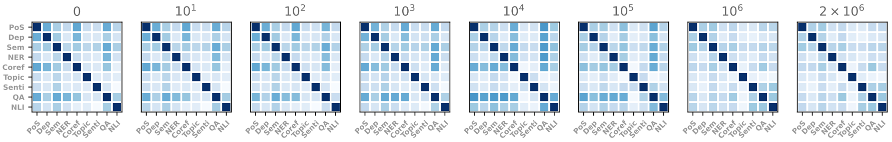


Figure 9.6: **Subspace Angles across Tasks** at start, end and at each order of magnitude of LM training time. Large angles (darker) and small angles (lighter) correspond to low and high similarity respectively.

9.5.3 Cross-task Interactions

The previous measures suggest that tasks could be grouped based on their shared learning dynamics. Our subspace-based approach for measuring cross-task SSAs (Figure 9.6) allows us to quantify this intuition. Comparing how much the subspaces of different tasks overlap provides a more holistic picture of task-relatedness than relying on specific data instances, and further shows how these similarities change over the duration of training.

Overall, angles between tasks follow linguistic intuitions: the syntax-related PoS and Dep subspaces span highly similar regions in the general representational space. Coref, at least initially, also exhibits high similarity to the syntactic tasks, likely exploiting the same features to resolve, e.g., pronominal matches. The NER subspace overlaps with PoS and Sem, but less with Dep, potentially focusing on entity-level information, but less on the functional relationships between them. QA, NLI, and later Senti, also share parts of their subspaces, while Topic is more distinct, having more overlap with the token-level NER and Sem tasks. While these patterns are already present at step 0, they become more pronounced during the early stages from 10k to 100k steps, and then become weaker towards the end as the subspaces disentangle and specialize within the model. Overall, subspaces appear to be related in a linguistically intuitive way, sharing more information during the critical learning phase, followed by later specialization.

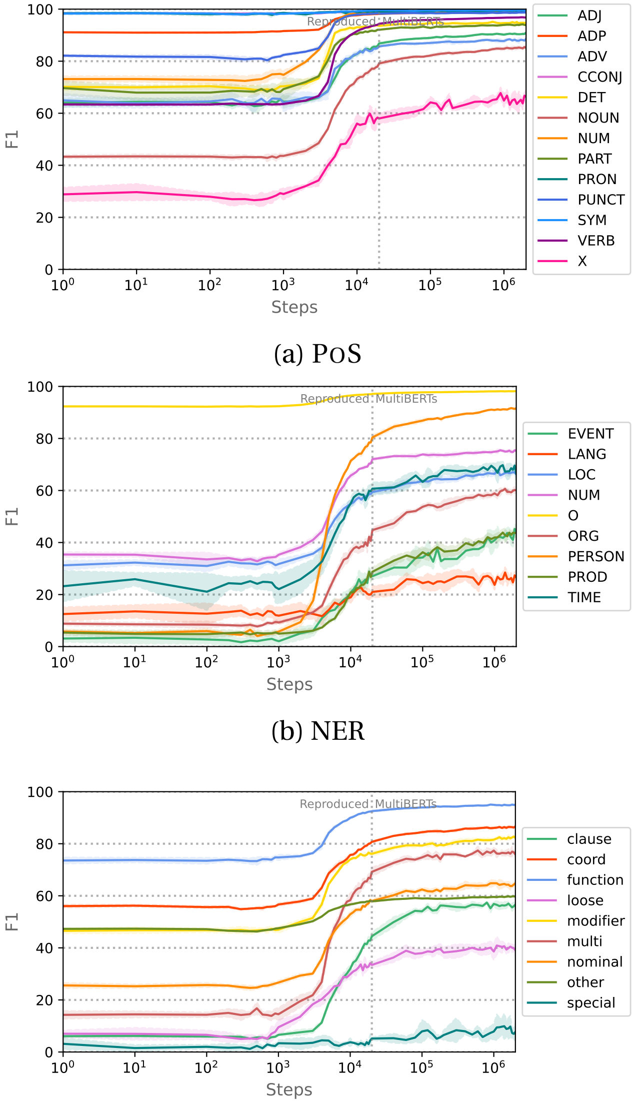
These learning phases suggest that introducing linguistically motivated inductive biases into the model may be most beneficial during the early critical learning phase, rather than at the end, after which subspaces have specialized. This is an important consideration for multi-task learning, where the prevalent approach is to train on related tasks after the underlying LM has already converged. As it is also often unclear which tasks would be beneficial to jointly train on given a target task, our subspace-based approach to quantifying task similarity could provide a more empirically grounded relatedness measure than linguistic intuition.

9.6 Practical Implications

Based on these learning dynamics, we next analyze their impact on downstream applications. As the previous results suggest, around 10k–100k steps already suffice to achieve the highest information gains for most tasks and reach close to 90% of final codelength and probing performance. At the same time, performance and subspaces continue changing throughout training, even if to a lesser degree. In order to understand what is being learned in these later stages, we analyze finer-grained probe performance (Section 9.6.1), whether these dynamics are domain-specific (Section 9.6.2), and what effects they have on full fine-tuning (Section 9.6.3).

9.6.1 Class-wise Probing

First, we take a closer look at what is being learned during later LM training via class-wise F1. For PoS (Figure 9.7a), most classes follow the general learning dynamics observed in Section 9.5, with no larger changes occurring beyond 10k steps. One outlier is the NOUN category, which continues to increase in performance while the other classes converge. Similarly for NER (Figure 9.7b), we observe that most classes stagnate beyond 10k steps, but that events (EVENT), products (PROD), and especially persons (PERSON) and organizations (ORG), see larger performance gains later on. What sets these classes apart from, e.g., determiners (DET) and pronouns (PRON), as well as times (TIME) and numbers (NUM), is that they represent open-domain knowledge. Performance on these classes likely improves as the LM observes more entities and learns to represent them better, while the closed classes are acquired and stabilize early on.



(a) PoS

(b) NER

(c) DEP

Figure 9.7: **Class-wise F1 over LM Training Time** for PoS, NER and DEP as measured on each task's dev split (standard deviation across seeds). For readability, classes are grouped according to Appendix 9.8.1.

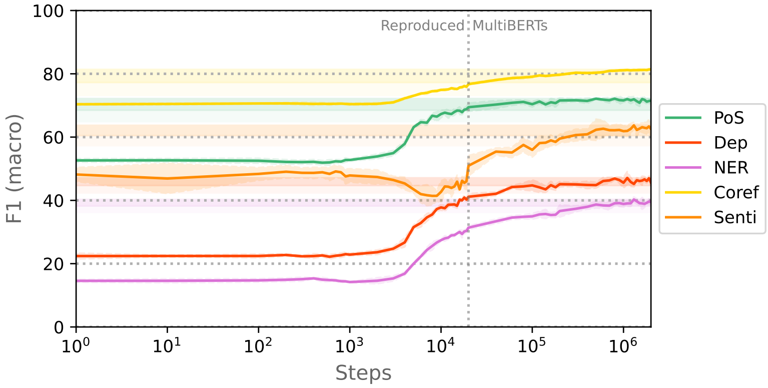


Figure 9.8: **OOD F1 (macro) over LM Training Time** with dark/light areas indicating 95%/90% of maximum performance (standard deviation across seeds).

Turning to DEP, we observe similar continued improvements for, e.g., `nominal` relations, while `functional` has already converged. In addition, these class-wise dynamics further reveal that `clausal` relationships converge later than other relations, towards 100k steps. At the same time, `modifier` relations also see stronger gains starting at 100k steps. Both coincide with the performance and codelength improvements of QA, NLI and especially SENTI, as well as with the layer depth and subspace specializations. We hypothesize that this is another learning phase during which the existing lower-level syntactic knowledge is embedded in better long-range contextualization.

9.6.2 Domain Specificity

Next, we investigate whether the previous findings hold across domains, and whether LM training duration has an effect on cross-domain transferability by gathering out-of-domain (OOD) evaluation sets for five of the tasks. For POS, NER and COREF, we split OntoNotes 5.0 (Pradhan et al., 2013) into documents from `nw` (newswire), `bn` (broadcast news), `mg` (magazines) for in-domain training, and `bc` (broadcast conversations), `tc` (telephone conversations), `wd` (web data) for OOD evaluation, based on the assumption that these sets are the most distinct from each other. In addition, we use English

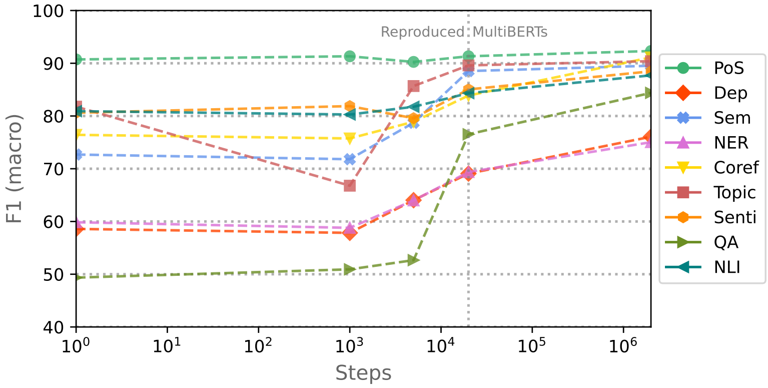


Figure 9.9: **F1 (macro) after Full Fine-tuning** of LMs initialized from checkpoints across LM pre-training time, as measured on each task's dev split (standard deviation across seeds too small to plot). Altered score range for readability.

Tweebank v2 (Liu et al., 2018) for DEP, and TweetEval (Rosenthal et al., 2017) for SENTI. The learning dynamics in Figure 9.8 show that transfer scores are generally lower, but that previously observed trends such as the steep learning phase between 1k and 10k steps, and NER's steeper incline compared to POS and DEP continue to hold. SENTI sees an overall greater F1 increase given more training than in the in-domain case. From this view, there are however no obvious phases during which only OOD performance increases. Instead, the OOD tasks seem to benefit from the same types of information as the in-domain tasks.

9.6.3 Full Fine-tuning

Finally, in terms of downstream applicability, we evaluate the effect of pre-training duration on fully fine-tuned model performance. Due to the higher computational cost, we conduct a sweep over a subset of checkpoints for which the probing experiments indicated distinctive characteristics: Starting from scratch at 0 steps, at 1k steps before the critical learning phase begins, at 5k steps around the steepest incline, at 20k after most growth plateaus, and at the final 2M step checkpoint (training details in Appendix 9.8.2). The results in Figure 9.9 show generally higher performance than for probing, as is to be expected.

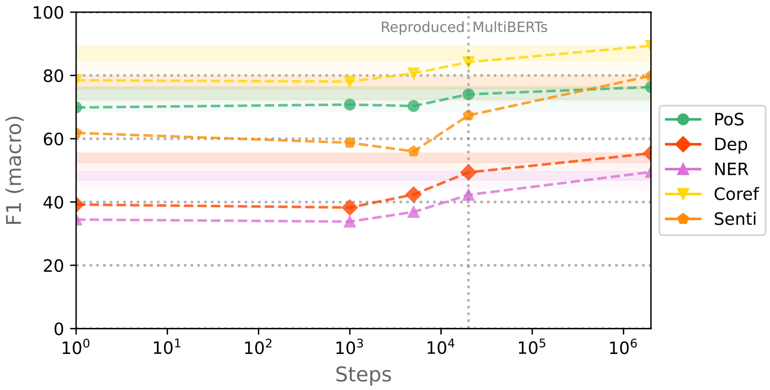


Figure 9.10: **OOD F1 (macro) after Full Fine-tuning** of LMs initialized from checkpoints across LM pre-training time, as measured on each task's dev split (standard deviation across seeds too small to plot).

Interestingly, for the majority of tasks, full fine-tuning follows the same learning dynamics as probing suggests—exhibiting the greatest improvements in the 1k–20k range.

While performance for most tasks is within the 90–95% range at 20k steps, starting full fine-tuning from the point of steepest information gain at 5k steps does not seem to suffice to reach this target. Pre-training beyond 10k steps therefore seems crucial in order for the LM encoder to become beneficial. While it is possible to reach final performance on PoS starting from a random encoder, even the similarly syntactic DEP sees substantial improvements up to the last pre-training step, likely connected to the improvement in longer-range contextualization observed in Section 9.6.1. Similar patterns can be observed for the other tasks, and especially for QA which only reaches usable performance beyond 20k steps. In terms of out-of-domain generalization for full fine-tuning, Figure 9.10 shows that continued pre-training does increase OOD performance, especially for more complex tasks. As such, LM pre-training, in combination with full fine-tuning, appears to be beneficial starting at around 10k steps for most traditional NLP tasks, followed by slower, but continued increases thereafter. As 10k updates constitute only 0.5% of full training, or 133M observed subword tokens, mid-resource languages may still benefit from language-specific, pre-trained LM encoders, even without access to English-level amounts of data. For example, within the multilingual OSCAR corpus (Ortiz Suárez et al., 2020), at least 60 languages (~40%) reach this threshold.

9.7 Conclusion

Our subspace-based approach to analyzing linguistic information across LM training has yielded deeper insights into their learning dynamics: In addition to the critical learning phase from 1k–10k steps, indicated by probing (Section 9.5.1) and fully fine-tuned performance (Section 9.6.3), our information theoretic approach identifies how linguistic subspaces continue changing, even if performance suggests otherwise (Section 9.5.2). For interpretability studies using single-checkpoint probing, this is crucial, as the information identified from a model may not be representative of final subspaces, especially if the model is undertrained (Hoffmann et al., 2022). Leveraging probes as characterizations of task-specific subspaces further allows us to quantify cross-task similarity, and surfaces how information is shared according to a linguistically intuitive hierarchy. This is particularly prominent during the critical learning phase, followed by later specialization, as more open-domain knowledge is acquired and the contextualization ability of the encoder improves. For multi-task learning, these dynamics imply that information sharing may be most effective early on, but more difficult after subspaces have specialized (Section 9.5.3). Finally, our analyses of OOD and full fine-tuning corroborate the previous learning dynamics (Section 9.6), showing that mid-resource languages could still benefit from pre-trained LM encoders at a fraction of the full fine-tuning costs, while simultaneously highlighting which information gains (i.e., open-domain, reasoning) require larger amounts of data.

Limitations

Despite aiming for a high level of granularity, there are certain insights for which we lack compute and/or training statistics. Obtaining the highest resolution would require storing instance-level, second-order gradients during LM training, in order to identify how each data point influences the model (similarly to Achille et al., 2019). Neither our study, nor other checkpoint releases contain this information, however we release intermediate optimizer states from our own LM training to enable future studies at the gradient-level.

Another consideration is the complexity of our probes. To enable cross-task comparisons, we deliberately restricted our probes to linear models, as any non-linearities make it difficult to apply subspace comparison measures. As such they may not be able to capture more complex information, such as for QA and NLI. By using information theoretic probing, we are able to observe that task-relevant information is still being learned, albeit to a lower degree than for the other tasks. To nonetheless measure codelength for these more complex tasks, the same variational framework could be applied to non-linear models (Voita and Titov, 2020), however the resulting subspaces would also be non-linear, negatively impacting comparability.

As MultiBERTs forms our underlying LM architecture, more research is needed to verify whether the observed dynamics hold for larger, autoregressive LMs. While we believe that these larger models likely follow similar learning dynamics, extracting comparable subspaces will be even more difficult as scale increases. In initial experiments, we investigated checkpoints of the LLMs listed in Section 9.2, however similarly finding that performance had already converged at the earliest available checkpoint. Furthermore, checkpoints lacked information regarding how much data, i.e., subword tokens, had been observed at each step.

Finally, we would like to highlight that this study is correlatory, and that practitioners interested in a particular task should verify its specific learning dynamics by training LMs with targeted interventions (e.g., Lasri et al., 2022; Chen et al., 2024; Hanna et al., 2023). For this work, such interventions would have been out of scope, as we aim for wide task coverage to analyze interactions between them. Nonetheless, we hope that our findings on these learning dynamics can inform future task-specific studies of this type.

Broader Impact

As this study examines the learning dynamics of LM representational spaces, its findings have wider downstream implications. Here, we specifically focus on three: First, for model interpretability, we identified that single-checkpoint probing does not provide the full picture with respect to how task-specific representations may change over time (e.g., layer depth in Section 9.5.2). This is critical when probing for sensitive information in LMs (e.g., bias), as representations may shift substantially, decreasing the effectiveness of interventions such as null-space projection (Ravfogel et al., 2020).

Second, we see a potential for multi-task learning to benefit from a better understanding of cross-task learning dynamics. Often it is unclear which task combinations boost/degrade each others' performance and what the underlying reasons are. Our findings suggest that similarities of task subspaces generally follow linguistic intuitions, but that there are distinct phases during which they share more or less information. Later specialization appears to be particularly important for more context-sensitive tasks, but may also make multi-task training more difficult, as tasks share less information at this stage. A question for future work may therefore be whether early-stage multi-task learning could lead to better downstream performance.

Third, for learning from limited data, this work identifies distinct phases during which different types of linguistic information are learned, as well as what their effects on fully fine-tuned and OOD performance are. We especially hope that the early acquisition of large amounts of information relevant for many traditional NLP tasks, encourages practitioners to revisit training encoders for under-resourced languages, despite the current trend towards larger models.

9.8 Appendix

9.8.1 Data Setup

Language Modeling

BookCorpus (Zhu et al., 2015) was originally collected to train a sentence embedding model for aligning passages in books to scenes in their movie adaptations. It reportedly contains around 11k books in 16 genres, both stemming from the public domain as well as more contemporary works. In total, the corpus contains 74M pre-tokenized sentences. In our experiments, we use the public version from the HuggingFace Dataset Hub (Lhoest et al., 2021) which is available as `bookcorpus`.

English Wikipedia (Wikimedia, 2022) was used for training both original BERT (Devlin et al., 2019) as well as MultiBERTs (Sellam et al., 2022), however neither the version used, nor the pre-processing steps have been reported. In our experiments, we make use of the `20220301.en` split of the `wikipedia` dataset from the HuggingFace Dataset Hub (Lhoest et al., 2021). It is available in a format where Wikipedia-specific markup has been removed, and each instance corresponds to the full text of an article. We further split these 6.5M articles into 144M sentences using spaCy v3.5.2 (Montani et al., 2023) and its `en_core_web` pre-processing pipeline.

Pre-training Corpus The final LM pre-training corpus consists of 109M sentence pairs from BookCorpus and Wikipedia, which are shuffled to be consecutive or randomly combined 50% of the time. They contain a total of 5.7B subword tokens of which 15% (and 80 at most) are replaced with a special `[MASK]` or another random token from the vocabulary. In practice, this results in 801M masked tokens, or 14.07% of the training data.

Probing

OntoNotes 5.0 (Pradhan et al., 2013) contains documents from six domains, which we split into an in-domain and out-of-domain set for our analysis in Section 9.6.2. Each sentence is annotated with multiple layers of which we use: parts-of-speech (PoS), named entities (NER), and coreference (COREF). It is split into 115,812 train, 15,680 dev, and 12,217 test instances.

PoS follows the Penn Treebank schema (Marcus et al., 1993) with 51 classes. The class-wise analysis in Section 9.6.1 uses a mapping from this labeling scheme to the Universal Part-of-Speech tagset (Petrov et al., 2012).

NER covers 18 entity types which are labeled using a `BIO` schema. For grouping these entities, we create the following custom mapping:

- `EVENT`: `EVENT`;
- `LANG`: `LANGUAGE`;
- `LOC`: `GPE`, `LOC`, `NORP`;
- `NUM`: `CARDINAL`, `DATE`, `MONEY`, `ORDINAL`, `PERCENT`, `QUANTITY`;
- `ORG`: `ORG`, `FAC`;
- `PERSON`: `PERSON`;
- `PROD`: `PRODUCT`; `WORK_OF_ART`, `LAW`;
- `TIME`: `TIME`.

COREF is built by extracting sentences with self-contained coreferences. The coreferring tokens are labeled as `I`, while all other tokens are labeled as `O`.

English Web Treebank (Silveira et al., 2014) contains syntactic dependencies (DEP) from 36 classes. To linearize the task, each token is labeled with the relation to its head word. For grouping the dependency relations, we use the nine categories from the official Universal Dependencies taxonomy (de Marneffe et al., 2014). The dataset consists of 12,543 train, 2,001 dev, and 2,077 test instances.

English Tweebank v2 (Liu et al., 2018) constitutes the OOD setup for DEP—specifically Twitter data. It adheres to the same Universal Dependencies relation label set as EWT, and consists of 1,639 (unused) train, 710 dev, and 1,201 test instances.

Parallel Meaning Bank (Abzianidze et al., 2017) contains three semantic annotation layers, of which we use the Universal Semantic Tags (SEM; Abzianidze and Bos, 2017). They denote 69 cross-lingual, lexical semantic categories at the token level. For grouping these tags, we combine the taxonomies of (Bjerva et al., 2016) and (Abzianidze et al., 2017) into 14 higher-level categories. The dataset consists of 7,745 train, 1,174 dev, and 1,053 test instances.

20 Newsgroups (Lang, 1995) contains email threads from 20 mailing lists, grouped by their TOPIC. Our experiments use the `bydate`-version which is sorted by date and removes duplicate entries and email headers containing the topic title. As the official data does not contain a dev split, we subdivide the training data, resulting in 9,051 train, 2,263 dev, and 7,532 test instance.

Stanford Sentiment Treebank (Socher et al., 2013) contains movie reviews along with their constituency parses and SENTI labels. We use the binarized SST-2 version with `positive/negative` labels. The dataset consists of 67,349 train, 872 dev, and 1,821 test instances.

TweetEval (Rosenthal et al., 2017) consitutes the OOD setup for SENTI. It contains seven annotation layers on Twitter data. We use the general sentiment labels and binarize them to match SST-2. The dataset consists of 45,615 (unused) train, 2,000 dev, and 12,284 test instances.

Stanford Question Answering Dataset (Rajpurkar et al., 2016) contains user-generated questions for which answer passages can be found in the corresponding Wikipedia articles, i.e., extractive question answering (QA). In our experiments, a question forms the first input sequence to the model, followed by a separator `[SEP]` token, and the relevant Wikipedia passage. Tokens within the answer passage are labeled as I, while everything else is O. The dataset consists of 87,599 train, and 10,570 dev instances.

Stanford Natural Language Inference Dataset (Bowman et al., 2015) contains premise-hypothesis pairs for natural language inference (NLI). Given a sentence pair, separated by [SEP], the task is to predict whether the relation of the two inputs is an entailment, contradiction, or neutral. The dataset consists of 550,152 train, 10,000 dev, and 10,000 test instances.

9.8.2 Experimental Setup

Language Modeling

Architecture The LM architecture used in our experiments is MultiBERTs (Sellam et al., 2022), which follows BERT$_{\text{base}}$ (Devlin et al., 2019), i.e., 12 layers with $d = 768$. We use the checkpoints published on HuggingFace Hub (Wolf et al., 2020) under google/multiberts-seed_[0-4]- step_[0-2000k]. For our own early checkpoints, we start from step_0 of each respective seed and train on the same data in the same order. From this training run, we store checkpoints and optimizer states at steps 10, 100–1,000 in increments of 100, 1,000–20,000 in increments of 1,000, and 40,000 for overlap comparisons, resulting in 29 additional models per initialization.

Training The LM training procedure for our own early checkpoints follows Sellam et al. (2022) as closely as possible. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a 10^{-4} learning rate, β_1 = 0.9, β_2 = 0.999 and a 10^{-2} weight decay. The learning rate is coupled to a polynomial schedule with a 10^4 step warm-up and consequent decay until the end. Each batch contains 256 sentence pairs with a maximum length of 512 subword tokens. The model is trained to fill in masked tokens (MLM) using a language modeling head, as well as to predict whether one sentence follows another (NSP) based on the [CLS] token and a separate linear classification head.

Probing

Architecture Each probe receives embeddings $\{h_0, \ldots, h_l\} \in \mathbb{R}^d$ from all l layers (including the non-contextualized layer 0) as input, and summarizes them into a learned weighted average using $\alpha \in \mathbb{R}^l$ following (Tenney et al., 2019a). This representation $h' = \sum_{i=0}^{l} \alpha_i h_i$ is then multiplied by a linear transformation $\theta \in \mathbb{R}^{d \times c}$ to produce logits for the c output classes. Following the variational MDL formulation of Voita and Titov (2020), each parameter w in θ is drawn from a normal distribution $w \sim \mathcal{N}(z\mu, z^2\sigma^2)$ with learned mean μ and variance σ^2, both scaled by z. There is one z per input dimension d, which is is also drawn from a normal distribution $z \sim \mathcal{N}(\mu_z, \sigma_z^2)$ with its own learned mean μ_z and variance σ_z^2. During training this process is made differentiable using the reparametrization trick (Kingma et al., 2015). Each w and z pair is coupled to a joint normal-Jeffreys prior $\gamma(w, z) \propto \frac{1}{|z|}\mathcal{N}(w|0, z^2)$, according to Figueiredo (2001) and Louizos et al. (2017). It induces sparsity in θ by encouraging values of w which are close to zero and have low variance. While the probe could theoretically be made more complex (i.e., non-linear), we specifically use a linear model in order to enable geometric comparisons between the resulting subspaces.

Training Both α and θ are jointly optimized by minimizing cross-entropy between predictions and gold labels. In addition, the KL divergence between θ's posterior β and its sparsity inducing prior γ are minimized according to Equation 9.1 to ensure maximum compression of both the data and the probe itself. Following (Voita and Titov, 2020), we use the Adam optimizer (Kingma and Ba, 2014) with a 10^{-3} learning rate, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and 0 weight decay. Probes are trained with a batch size of 64 for a maximum of 30 epochs, and with early stopping on the development data if losses do not decrease.

Evaluation Following Saphra and Lopez (2019), we probe for task-specific information at the subword-level, meaning that for token-level tasks each token label is repeated across all of its constituent subwords, while for sequence-level tasks, the sequence label is repeated across all subwords. This corresponds to identifying task-specific information that is consistent across all contextualized embeddings within a sequence. To evaluate performance, we use macro-F1, as it is easier to interpret overall class-wise performance in-spite of class imbalances. E.g., NER and QA have a high number of O labels which are classified correctly with above 95% F1. With micro-F1, performance would appear unreasonably high, even if no named entities or answers would be identified.

Full Fine-tuning

Architecture For the full fine-tuning experiments in Section 9.6.3, we train all parameters in the LM encoders from steps 0, 1k, 5k, 20k and 2M. For token-level tasks, we add a linear layer on top of the final contextualized embedding layer, while for sequence-level tasks, a linear layer is fed each input sequence's [CLS] token. These linear classification heads have the same dimensionality as the linear probes, but are not sampled variationally.

Training The fully fine-tuned models are trained using cross-entropy loss. Based on the recommendations in Devlin et al. (2019), we set the learning rate of the Adam optimizer (Kingma and Ba, 2014) to 3×10^{-5}, and retain the other hyperparameters. Models are trained for a maximum of 30 epochs, with early stopping on the development data when the loss stagnates.

Implementation

Implementations use PyTorch v1.13 (Paszke et al., 2019) and NumPy v1.24 (Harris et al., 2020). Visualizations use matplotlib v3.6 (Hunter, 2007). Models were trained on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. Probe training takes 10–60 minutes per checkpoint, dependent on the size of the dataset. Data pre-processing for language modeling (i.e., sentence splitting, tokenization, sampling and masking) takes 160 hours for the full dataset. The LM training process itself takes around 50 hours for 40k steps. The random seeds used in our experiments follow MultiBERTs (Sellam et al., 2022) and are: 0, 1, 2, 3, 4. Further, the code for reproducing

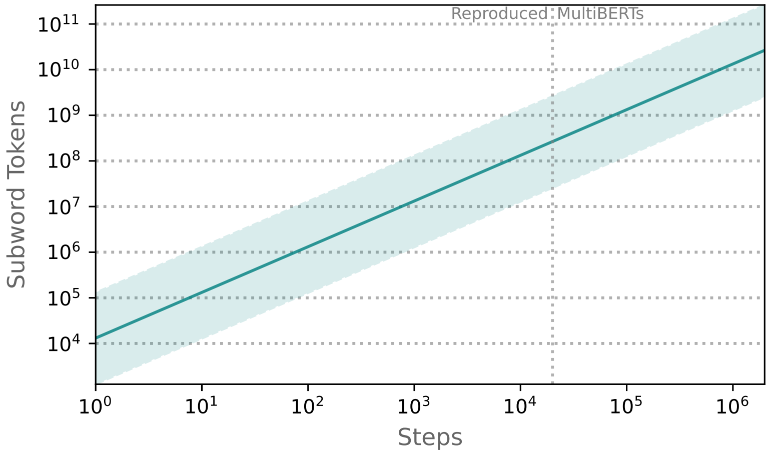


Figure 9.11: **Number of Subword Tokens Observed** during LM training, following the trajectory of our re-created dataset, plus the estimated upper/lower bounds for prior work.

our experiments is available at https://github.com/mainlp/subspace-chronicles.

9.8.3 Additional Results

Language Modeling

Figure 9.11 plots the number of subword tokens observed by the LM over the course of training. While this corresponds to the statistics of our re-created LM training corpus, the fact that the curve lies exactly between the feasible upper and lower-bounds given batch size and minimum/maximum subword tokens per sequence, we estimate that the original models also followed this trajectory.

For our LM training runs from step 0 to 40k, Figure 9.12 shows how NSP and especially MLM losses start decreasing already after 100 pre-training steps. In general, losses on the actual LM pre-training tasks appears to decrease earlier by one order of magnitude, before probing performance and codelength improve.

Across LM training, task subspaces extracted at the same timestep, but across different seeds, are close to orthogonal to each other, as

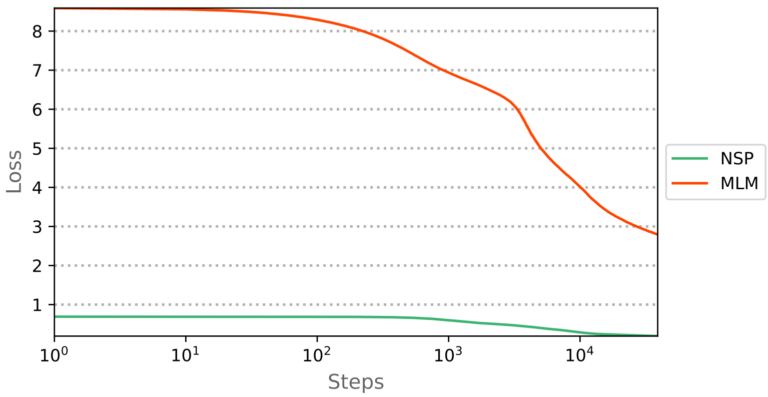


Figure 9.12: **MLM and NSP Losses** during the pre-training procedure described in Section 9.4.1, as measured for each batch from step 0 to 40k. Checkpoints, training statistics and optimizer states released on publication.

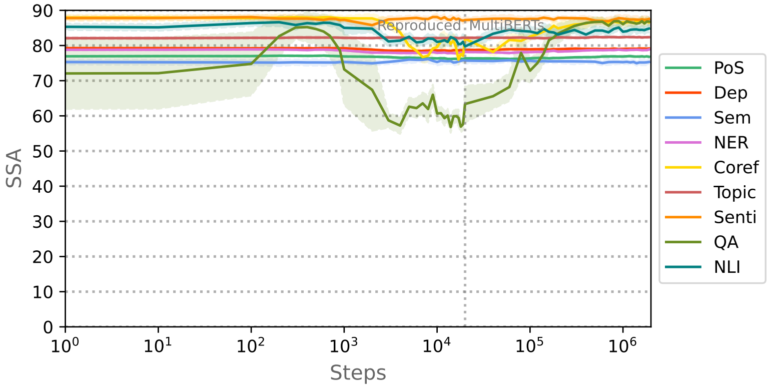
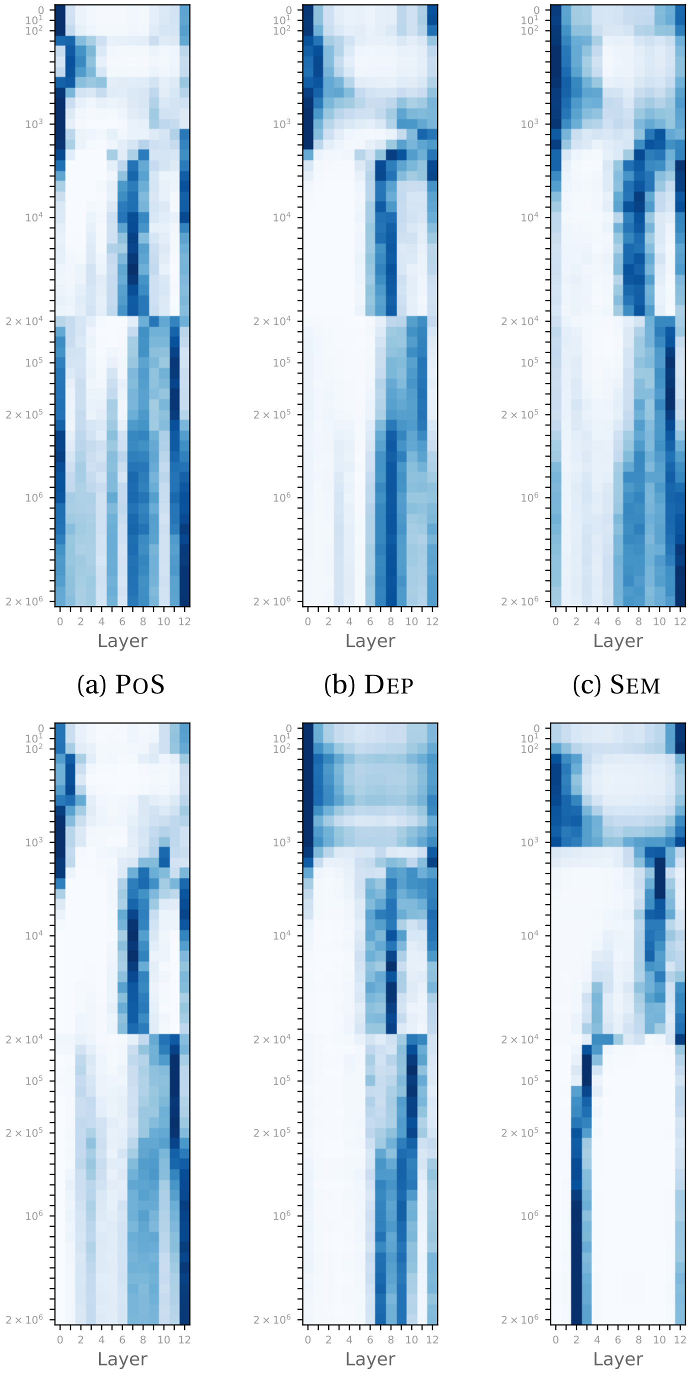


Figure 9.13: **SSAs across Random Initializations** for task-wise probes at each timestep (standard deviation across pairwise comparisons).

seen in Figure 9.13. This contrasts checkpoints starting from the same initialization (see Section 9.5.2), where the rate of change differs substantially across training, but generally remains under 60°, even when trained on slightly different data (Section 9.4.1).

Layer Weightings

In addition to the center of gravity (COG) measure used in Section 9.5.2, the full layer weightings α for each task and timestep are shown in Figure 9.14. They provide a more detailed picture for cases in which multiple layers are weighted similarly, e.g., for the first checkpoints of DEP (Figure 9.14b, SEM (Figure 9.14c) and NER (Figure 9.14d). Both the earlier and later layers are weighted strongly, meaning that probes are making use of non-contextual plus mixed representations at these early stages. In contrast, SENTI and NLI almost exclusively rely on the last layer, while COREF and QA are initially spread out across all layer depths. Later on in training, these weightings also exhibit the specialization observed in Section 9.5.2, however weights are more spread out and do not collapse onto a single layer. This is most prominent with POS, but also DEP, SEM, NER and COREF, which all make use of information from across a wider range of depths.



(a) PoS

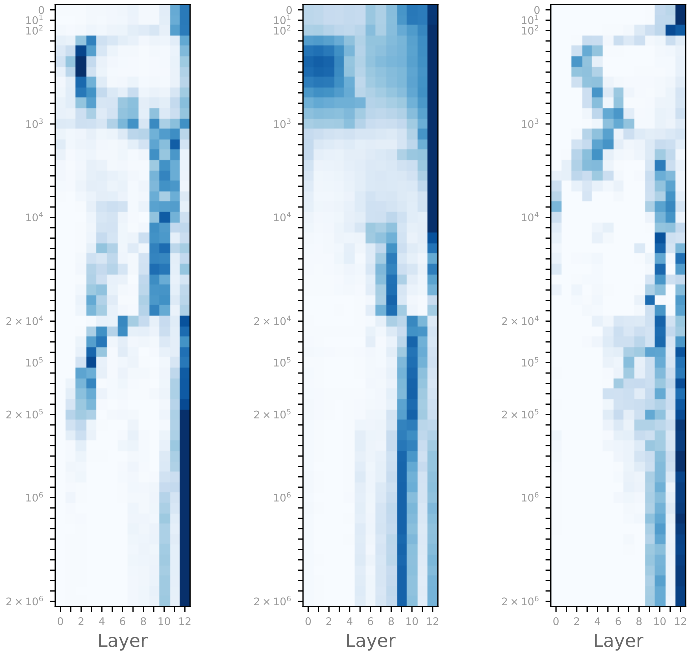
(b) Dep

(c) Sem

(d) NER

(e) Coref

(f) Topic



(g) SENTI (h) QA (i) NLI

Figure 9.14: **Layer Weightings α over LM Training Time** for all tasks, corresponding to the center of gravity measure in Section 9.5.2. Darker/lighter fields correspond to more/less weight respectively.

Spectral Probing 10

The work presented in this chapter is based on the publication: Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022c. Spectral probing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7730–7741, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

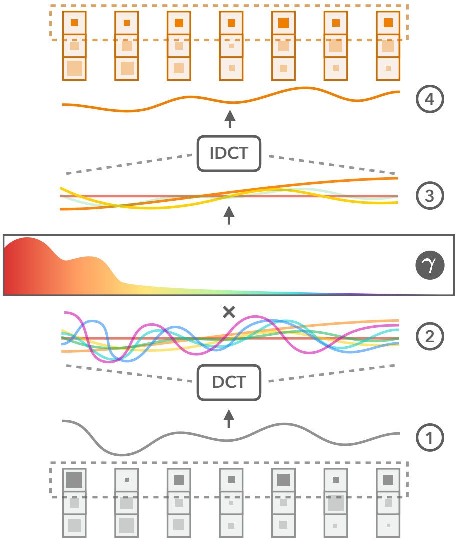


Figure 10.1: **Visualization of Spectral Probing**. Given a sequence of embedding values (1), decompose into composite frequency waves using DCT (2), apply the learned filter ($\boldsymbol{\gamma}$), retaining a subset of waves (3), for which IDCT returns the filtered sequence of values (4).

Abstract

Linguistic information is encoded at varying timescales (subwords, phrases, etc.) and communicative levels, such as syntax and semantics. Contextualized embeddings have analogously been found to capture these phenomena at distinctive layers and frequencies. Leveraging these findings, we develop a fully learnable frequency filter to identify *spectral profiles* for any given task. It enables vastly more granular analyses than prior handcrafted filters, and improves on efficiency. After demonstrating the informativeness of spectral probing over manual filters in a monolingual setting, we investigate its multilingual characteristics across seven diverse NLP tasks in six languages. Our analyses identify distinctive spectral profiles which quantify cross-task similarity in a linguistically intuitive manner, while remaining consistent across languages—highlighting their potential as robust, lightweight task descriptors.

10.1 Introduction

Analyzing the contextualized embedding representations of pretrained language models (LMs) using lightweight probes (Hewitt and Liang, 2019; Voita and Titov, 2020) has identified latent features in the untuned encoders which are highly relevant to downstream NLP tasks at various layer depths (Tenney et al., 2019a). Orthogonally, linguistic phenomena are also encoded at different timescales: i.e., rapidly changing (sub-)word-level information versus slower changing sentence or paragraph-level information. Decomposing contextualized embeddings into frequencies with different rates of change has yielded initial insights into the timescales at which these task-specific latent phenomena occur (Tamkin et al., 2020). These findings currently rely on handcrafted spectral filters and are limited to English. To enable more efficient analyses of finer-grained, continuous frequency spectra in contextualized representations covering more tasks and languages, this work contributes:

- A fully differentiable spectral probing framework for *learning* which frequencies are relevant for specific NLP tasks (Section 10.2).[1]
- A multilingual probing study examining timescale characteristics of seven diverse NLP tasks across six languages (Section 10.3).
- An analysis of the relationships between the spectral profiles of different tasks and their consistency across languages (Section 10.4).

10.2 Probing for Spectral Profiles

Spectral Probing (Figure 10.1) builds on established signal processing methods (Ahmed et al., 1974) and recent findings on the manual frequency filtering of contextual embeddings (Tamkin et al., 2020). The

[1] Code at https://github.com/mainlp/spectral-probing.

method automatically learns spectral profiles which measure the relevance of specific frequencies to a given task by amplifying or reducing contextual information with different rates of change.

Discrete Cosine Transform (Ahmed et al., 1974; DCT) is an invertible method for decomposing any sequence of real values $\{x_0, \dots, x_{N-1}\}$ (e.g., all values of an embedding dimension) into a weighted sum over cosine waves with different frequencies. The number of frequencies equals the sequence length N, as the lowest frequency wave is a constant ($k = 0$) and the highest frequency wave completes one cycle every timestep ($k = N - 1$). The coefficient $X_n^{(k)}$ for a wave at DCT index k at timestep n is calculated as:

$$X_n^{(k)} = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]. \tag{10.1}$$

Inverting the DCT (IDCT) using all $X_n^{(k)}$ will return the original sequence. However, weighting coefficients for some k by 0 will return a filtered version. Zeroing out all k above a threshold will only retain lower frequencies and make values oscillate with a slow rate of change. Vice-versa, zeroing out all k below a threshold will only retain higher frequencies—amplifying short-term changes.

Fixed-band Filters Applying frequency filters to a sequence of contextualized embeddings extracts linguistic information at different timescales. Within this formulation, the values across each embedding dimension are gathered into a real-valued sequence to which transformations such as the DCT can be applied. In seminal work, Tamkin et al. (2020) apply manually defined low ($k \in [0, 1]$), mid-low ($k \in [2, 8]$), mid ($k \in [9, 33]$), mid-high ($k \in [34, 129]$) and high frequency filters ($k \in [130, 511]$) to English BERT embeddings (Devlin et al., 2019) to investigate how accurately a linear probe can extract task-specific information within certain spectra. Capturing the full picture using manual, fixed-band filters is however not computationally feasible: Relevant frequencies might not lie in a contiguous band, and furthermore, frequencies can not only be turned on or off (i.e., weighted 0 or 1), but can actually be weighted continuously in $[0, 1]$.

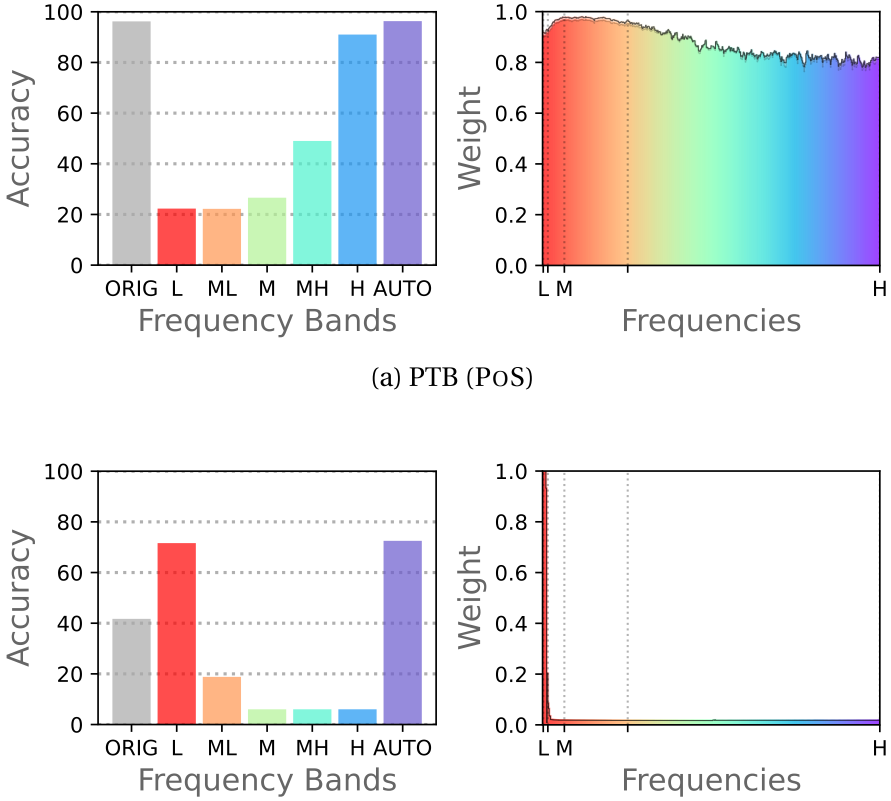
Learnable Filters To capture the complete picture, we propose spectral probing which *learns* a continuous weighting of frequencies relevant to a task. In effect, the spectral filter is a vector $\boldsymbol{\gamma} \in \mathbb{R}^M$ for which each entry corresponds to the weight assigned to a particular frequency. Before inverting the DCT, each $X_n^{(k)}$ is multiplied by the sigmoid-scaled weight $\gamma^{(k)} \in [0,1]$ which will then retain or filter out frequencies at index k. As M depends on the sequence length N, which changes across inputs, the spectral probe dynamically scales $\boldsymbol{\gamma}$ to the length at hand using adaptive mean pooling. In practice, we set M to the maximum input length for our given encoder (e.g., 512 for BERT) and shrink $\boldsymbol{\gamma}$ appropriately, as a wave cannot cycle more often than there are values. It would however be equally possible to set M smaller than N and interpolate the filter up to the length required. Overall, $\boldsymbol{\gamma}$ is a lightweight parameter which can be easily incorporated between the frozen encoder and probing head, and uses the existing training objective to jointly learn which frequencies to amplify or filter out.

10.3 Experiments

10.3.1 Monolingual

Setup Initially, we compare spectral probing to previous fixed-band filters by reproducing the highest and lowest frequency experiments by Tamkin et al. (2020). These are the tasks of tagging parts-of-speech (PoS) in the Penn Treebank (Marcus et al., 1993; PTB) as well as classifying TOPICS in the 20 Newsgroups corpus (Lang, 1995; 20News).

On the modeling side, we follow Tamkin et al. (2020) and train a linear probe (Alain and Bengio, 2017) on top of the frozen LM encoder to classify each manually/automatically filtered contextual embedding in an input sequence. This corresponds to probing and evaluating for the amount of task-relevant information in each sub-word across a sequence (e.g., underlying topic contextualization). The bands for the five manual filters follow the original definitions (see Section 10.2), and we compare them to unfiltered (ORIG) as well as automatically filtered (AUTO) embeddings from our spectral probe (details in Section 10.6.2).



(a) PTB (PoS)

(b) 20News (Topic)

Figure 10.2: **Monolingual Results on PTB and 20News.** Acc of unfiltered (Orig), low (L), mid-low (ML), mid (M), mid-high (MH), high (H), and the spectral probe's automatic filters (Auto) with frequency weightings.

Results Figure 10.2 shows the accuracy (ACC) of the six prior filtering strategies in addition to the learned frequency weightings of the spectral probe. The unfiltered and manually filtered embeddings corroborate previous findings Tamkin et al. (2020), with high frequencies performing best on POS, and the lowest frequencies performing best on TOPIC.

The spectral probe achieves 95.9% ACC for POS, outperforming ORIG by a 0.1% margin and the best manual filter by 5.2%. The spectral profile in Figure 10.2a (right) sheds light on why this may be the case: While it also prioritizes high (sub-)word-level frequencies, the learned filter additionally includes surprising amounts of mid-high and lower frequencies, emphasizing the need for both local and global context to achieve high performance.

For TOPIC, the spectral probe achieves 72.1% ACC, outperforming both ORIG (41.3%) and the fixed low-band filter (71.2%). The learned filter (see Figure 10.2b, right) mirrors the fixed-band results: Only the lowest bands are active, while all higher ones are not. As mid-low frequencies still appear to contain weaker amounts of topic information, the soft inclusion of this region by the spectral probe could account for its performance boost. Overall, spectral probing confirms and refines frequency ranges from prior work while surfacing more detail and requiring no manual probe engineering, with only a single probing run instead of five.

10.3.2 Multilingual

Leveraging spectral probing, we extend timescale analyses beyond English and investigate spectral profiles across more diverse tasks and languages.

Setup Each experiment covers German (DE), English (EN), Spanish (ES), French (FR), Japanese (JA) and Chinese (ZH). The tasks are POS-tagging and dependency relation classification (DEP) from Universal Dependencies (Zeman et al., 2021); named entity recognition (NER) from WikiANN (Pan et al., 2017); question answering (QA) from MKQA (Longpre et al., 2021); sentiment analysis (SENTI) and TOPIC classification from Multilingual Amazon Reviews (Keung et al., 2020); natural

TASK	ORIG	AUTO
POS	92.4±1.9	92.5±1.8
DEP	78.6±4.3	79.3±4.3
NER	88.0±2.7	88.1±2.6
QA	62.9±1.6	67.1±1.2
SENTI	57.4±0.9	64.3±1.1
TOPIC	27.1±8.1	37.2±8.2
NLI	44.1±4.1	56.3±5.6

Table 10.1: **Multilingual Results** (ACC) of unfiltered (ORIG) and automatically filtered (AUTO) embeddings. Means ± standard deviations over languages and random initializations (details in Section 10.6.3).

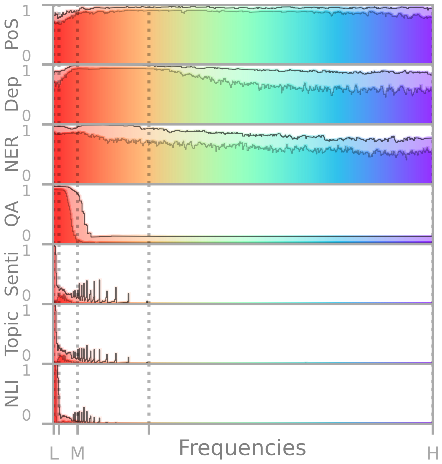
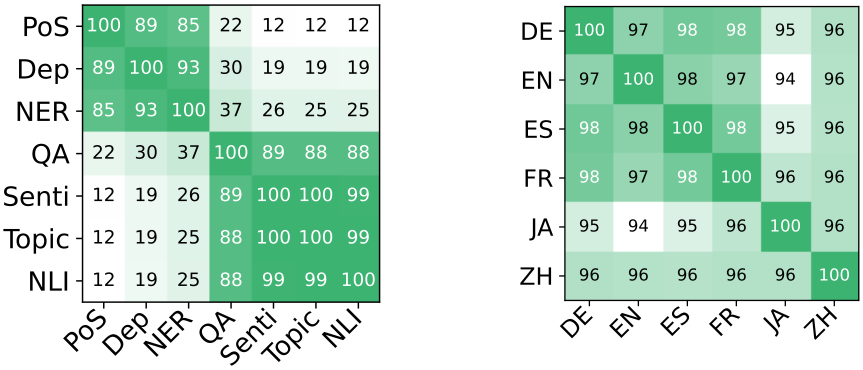


Figure 10.3: **Spectral Profiles** of all tasks (weight per frequency), with lower and upper bounds across languages.

language inference (NLI) from XNLI (Conneau et al., 2018b) and JSNLI (Yoshikoshi et al., 2020) for JA (details and examples in Section 10.6.1).

For each language-task combination we train a linear probe on the unfiltered embeddings of multilingual BERT (Devlin et al., 2019; mBERT) and on the automatically filtered representations from our spectral probe. The remaining settings are identical to the monolingual setup (details in Section 10.6.2).

Results Table 10.1 shows equivalent or higher ACC for the spectral filter compared to the unfiltered embeddings for all tasks and languages. This increase is less pronounced for token-level tasks, but much larger for tasks where sequence-level information is critical. Figure 10.3 visualizes how POS, DEP and NER retain large parts of the original spectrum, while QA, SENTI, TOPIC and NLI appear to benefit from filtering out higher frequencies. This shows how tasks exhibit structures at different timescales and that spectral probing is able to identify these communicative levels consistently not only in English, but also across languages—an effect which we analyze more extensively next.



(a) Tasks

(b) Languages

Figure 10.4: **Filter Overlap across Tasks/Languages** as measured in percentage-normalized L1 distance.

10.4 Spectral Profiling Analysis

Each task's distinct spectral profile (Figure 10.3) allows us to analyze their relation to the timescale hierarchy of linguistic structures, and quantify cross-task similarities within and across languages. For this we use the percentage-normalized L1 distance (i.e., 0%–100% overlap) between filters (Figure 10.4).

Cross-task Overlap Overall, we observe a dichotomy between broad-frequency, token-level tasks and low-frequency, sequence-level tasks (Figures 10.3 and 10.4a). In addition, there appears to be a hierarchy which depends on the timescales of the linguistic structures involved. Notably, compared to prior fixed-band filters, none of the learned filters fully excludes low frequencies. For instance, high-frequency information is most important to retrieve PoS, but reaching the performance of the original embeddings also requires some lower-frequency information—most likely to disambiguate difficult cases based on sentence-level context.

DEP appears to benefit the least from both lower and higher-frequency information. Instead, the strong weight on mid-high frequencies matches the fact that dependency relations span multiple words and benefit from information at the phrase-level. NER sees a further decrease in high-frequency information, coupled with an uptick in lower frequencies. We hypothesize that phrase and sentence-level information become more important for disambiguating certain entity types (e.g., `ORG` and `LOC`). Across the token-level tasks this shift from higher to lower frequencies is also reflected in filter overlap which decreases from syntactic to semantic token-level tasks, while their overlap with sentence-level tasks increases (Figure 10.4a).

The sequence-level tasks share low-frequency spectral profiles which overlap more with each other than do the token-level tasks. In fact, SENTI and TOPIC overlap almost perfectly (although the latter involves less mid-range frequencies). This similarity is unlikely to be the result of the shared underlying dataset as both tasks also overlap with the unrelated XNLI and JSNLI datasets. At the same time, the POS and DEP tasks, which also share datasets, have a lower overlap despite being based on the exact same inputs. Overall, SENTI, TOPIC and NLI all appear to rely on information which is consistent across a sequence—explaining why simple methods such as mean-pooled sentence embeddings can be effective in these scenarios.

QA provides an intermediate case: While it is reliant on low frequencies it also includes more mid-low and a small amount of higher frequency information. This is reflected in Figure 10.4a, where it shares more overlap with the token-level tasks than all other sequence-level tasks. Since probing for the correctness of a question-answer pair is dependent on finer-grained information than the general sentiment, topic or semantic coherence of a sequence, this inclusion of higher frequency information matches linguistic intuitions.

Cross-lingual Consistency Finally, we investigate the similarity of learned spectral profiles across languages. While Figure 10.3 shows that there is some variance between the filters of different languages within a task, Figure 10.4b shows that actual quantitative overlap between languages is high, ranging from 94%–98%. This holds even across distinctive pairs such as JA-EN which differ substantially in factors such as sub-word length and distance between syntactic dependents. This strong consistency highlights the potential for spectral profiles to provide language-agnostic features for task characterization and comparison.

10.5 Conclusion

Linguistic information at different timescales is an, as of yet, underexplored dimension in contextualized embeddings. We propose a fully differentiable *spectral probe* which automatically learns to weigh frequencies that are relevant to a specific task and improves over prior fixed-band filters by capturing continuous mixtures over frequencies (Section 10.2). This enables us to not only outperform the manual filters while using one probe instead of five, but to also identify that high-frequency tasks still benefit from low-frequency information (Section 10.3.1).

Extending spectral probing to seven tasks in six languages, we trained task-specific filters which outperformed the original, unfiltered embeddings. The resulting spectral profiles furthermore shed light onto how linguistic information at different timescales relates to different task types (Section 10.3.2). They not only match the linguistic intuitions underlying each task, but also enable quantitative comparisons between them. The analysis of the filters' overlap surfaced a clear dichotomy between token and sequence-level tasks, but also highlighted intersecting frequency ranges which contain information relevant across task types. Finally, the language-agnostic nature of these spectral profiles highlights future avenues towards more robust task descriptors (Section 10.4).

Limitations

Our experiments cover a diverse, but non-exhaustive set of NLP tasks and languages. While more extensive than prior related work (Tenney et al., 2019a; Tamkin et al., 2020), we elaborate in the following regarding the motivation of the final setup: As the aim of our study was to investigate the cross-lingual properties of the underexplored timescale dimension of contextualized representations, the set of languages and tasks used in our experiments emphasizes consistency across languages. This limits us to high-resource languages for which datasets covering every task are available. However, with cross-lingual stability confirmed in our experiments, the study of lower-resourced languages is a clear avenue for future research.

Despite using a set of well-established datasets, it is important to keep data quality in mind when interpreting the results—even for these high-resource languages. In our initial exploratory data analyses, we identified and confirmed limitations known to the original dataset authors in that many include silver, or weakly filtered annotations driven by automatic matching and translation (e.g., WikiANN, XNLI, JSNLI). As we are less interested in benchmarking performance and rather focus on the feasibility and analysis of our spectral profiles, individual data instances of lesser quality should however be of limited concern. Section 10.6.1 details how each dataset was constructed originally, and also how it was pre-processed by us, such that results can be interpreted in the appropriate context.

In terms of modeling, we hope that future work will investigate spectral probes and their resulting task profiles across more encoder models with different architectures and pre-training strategies. Finally, while we have demonstrated spectral profiles to be suitable for characterizing different tasks consistently across languages, future research could supplement them with other descriptors such as embedding layer depth in order to identify even more distinctive profiles.

Ethics Statement

Given the theoretical nature and wide applicability of this work—both in terms of data domains and model architectures—it is difficult to anticipate broader impacts and future ethical implications. In general, benefits and harms in the field of probing originate from the information being investigated: While we are interested in linguistic timescale characteristics, probe-like methods have also been applied to protected attributes of data subjects in order to, for example, debias LMs (Ravfogel et al., 2020). Since this process involves personal information, any experiments extracting such characteristics should be sufficiently vetted for ethical acceptability. With spectral profiles being a relatively broad descriptor however, we do not expect them to identify frequencies exclusive to personal information or to replace existing, domain-specific probing methods.

Token-level Tasks									
PTB	In	Tokyo	,	trading	is	halted	during	lunchtime	.
	`IN`	`NNP`	`,`	`NN`	`VBZ`	`VBN`	`IN`	`NN`	`.`
UD	Can	rabbits	and	chickens	live	together	?		
	`AUX`	`NOUN`	`CCONJ`	`NOUN`	`VERB`	`ADV`	`PUNCT`		
	`aux`	`nsubj`	`cc`	`conj`	`root`	`advmod`	`punct`		
WikiANN	The	Zeros	formed	in	Chula	Vista	in	1976	.
	`B-ORG`	`I-ORG`	`O`	`O`	`B-LOC`	`I-LOC`	`O`	`O`	`O`

Sequence-level Tasks		

MKQA	when did love become a part of marriage? | 18th century
	when did love become a part of marriage? | 2016
		`1 (true)`
		`0 (false)`

AMR	All socks had large holes after a few months.	`apparel` `negative`
20News	[...] Does anyone know how to size cold gas roll control thruster tanks for sounding rockets? [...]	`sci.space`
XNLI	I've got more than a job. | I don't have a job or any hobby.	`contradiction`
JSNLI	地下鉄を待っている間に本を読む男。| 男は地下にいる。 The man reads a book while waiting for the subway. The man is underground.	`entailment`

Table 10.2: **Example Dataset Instances** annotated with respective token/sequence-level `labels`.

10.6 Appendix

10.6.1 Data Setup

In the following, we provide details about the versions, splits and preprocessing of each dataset. Additionally, we present example instances together with their token/sequence-level annotations in Table 10.2 (in English, where available). In our experiments, each model is tuned on the training split and only evaluated on the validation split as we are not interested in obtaining state-of-the-art results, but rather aim to analyze overall performance patterns across tasks. We use the original splits where provided and generate our own otherwise.

Penn Treebank (Marcus et al., 1993) We use Penn Treebank version 2 (PTB) as published in OntoNotes 4.0. Sections 02-21 were used for training, section 22 for validation, and section 23 for test, totaling 30,060, 1,336 and 1,640 instances respectively. The label space covers 48 part-of-speech tags. Note that Tamkin et al. (2020) use PTB version 3 in their experiments which we were unable to obtain due to licensing constraints. As such the exact data and splits may differ.

Universal Dependencies (Zeman et al., 2021) From Universal Dependencies version 2.9 (UD), we select the following treebanks: German-GSD Brants et al. (2004), English-EWT Silveira et al. (2014), Spanish-GSD McDonald et al. (2013), French-GSD Guillaume et al. (2019), Japanese-GSD Asahara et al. (2018), Chinese-GSD Shen et al. (2016b) with standard splits, totaling 66,040 training and 6,683 validation instances. The label set comprises the 17 UPOS classes and the 36 dependency relations which can occur between a word and its head.

WikiANN (Pan et al., 2017) This dataset contains silver NER data for 282 languages which was extracted from Wikipedia using URL references as a proxy for named entities. It contains the entity types location (`LOC`), person (`PER`) and organization (`ORG`) which are annotated in BIO-format. Our experiments use the existing data splits with 20,000 training and 10,000 validation instances.

MKQA (Longpre et al., 2021) Multilingual Knowledge Questions and Answer (MKQA) is an open-domain question answering dataset which covers 10,000 questions and their corresponding answers in an aligned corpus spanning 26 languages. After removing unanswerable questions, we use each correct QA pair to generate an additional incorrect pair for the same question, yielding a total set of 13,516 instances used in our experiments. To generate an incorrect answer, we sample an alternative answer of the same type (e.g., time, number) which does not equal the correct answer. Finally, we randomly split the data 80/20 into training and validation portions for which the instances are aligned across languages (i.e., the same questions and answers). The final task is a binary classification task for whether a QA pair is true or false, with a random baseline of 50%.

Multilingual Amazon Reviews (Keung et al., 2020) MAR are used for both sentiment analysis and topic classification. For SENTI, we convert the 1–5 star rating into $\{1, 2\} \rightarrow$ `negative`, $\{3\} \rightarrow$ `neutral` and $\{4, 5\} \rightarrow$ `positive`. For TOPIC, we consider the 30 product categories as topics. All original splits are kept, resulting in 200,000 training and 5,000 validation instances per language.

20 Newsgroups (Lang, 1995) This dataset contains English emails from 20 newsgroups and their corresponding topics. We use the `bydate`-version which is sorted by date and removes duplicate entries and email headers (which give away the topic). Of the official training and testing data, we subdivide the former 11,314 instances into an 80/20 train/validation split. Note that there may differences to the version used in Tamkin et al. (2020) due to alternative splitting strategies.

XNLI (Conneau et al., 2018b) The Cross-lingual Natural Language Inference (XNLI) dataset covers 15 languages translated from and including English (as it lacks Japanese data, we supplement it with JSNLI). The task is to identify the relation between a premise and a hypothesis as: `contradiction`, `entailment` or `neutral`. Our setups use the original training and validation splits with 392,702 and 2,490 input pairs respectively.

JSNLI (Yoshikoshi et al., 2020) This dataset contains premise-hypothesis pairs from the Stanford Natural Language Inference corpus (Bowman et al., 2015) which were translated automatically into Japanese and filtered for correctness. It contains 533,005 training and 3,916 validation instances with the same three classes as XNLI.

10.6.2 Experiment Setup

Models In the monolingual English experiments, we use `bert-base-cased` (Devlin et al., 2019; BERT) following Tamkin et al. (2020). For the multilingual experiments we use `bert-base-multilingual-cased` (Devlin et al., 2019; mBERT). For both LMs, we use respective checkpoints from the Transformer library's model hub (Wolf et al., 2020).

Manual, fixed-band filters as well as the automatically learned filters are applied to the contextualized embeddings produced by the last layer of either model. As visualized in Figure 10.1, we decompose the sequence of values from each embedding dimension (768 in both LMs) using the DCT (Ahmed et al., 1974; DCT-II), weight the appropriate k by a fixed amount or by the learned weight in $\boldsymbol{\gamma}$, before applying the IDCT to reconstruct a sequence of real values. These make up each dimension of the filtered embeddings.

Following Tamkin et al. (2020), the original/filtered embeddings are passed to a linear probe (Alain and Bengio, 2017) consisting of two parameters: a transformation $\boldsymbol{W} \in \mathbb{R}^{E \times C}$ and a bias $\boldsymbol{b} \in \mathbb{R}^{C}$, where E is the embedding dimension and C is the number of classes specific to each task.

Training As we run probing experiments, neither the 108M-parameter BERT, nor the 178M-parameter mBERT are fine-tuned. We only train the linear probe which has 1,538–36,912 parameters depending on the task, plus the 512 parameters of the learned spectral filter $\boldsymbol{\gamma}$. As in Tamkin et al. (2020), we use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $10^{-}3$ which decays by 0.5 every time the loss plateaus. Updates are applied in batches of size 32 across a maximum of 30 epochs, with an early stopping patience of 1. Each setup is run with the five random seeds: 1932, 2771, 7308, 8119, 9095. On our hardware consisting of an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU, training a probe takes approximately 10 minutes.

Evaluation In order to probe a sequence of contextualized embeddings for information at different timescales, it is necessary to apply each filter at the sub-word level. To measure the effect of different frequencies, we follow Tamkin et al. (2020) and evaluate all tasks using accuracy (ACC) on the sub-word level. Note that for token-level tasks each token label is therefore repeated across all of its sub-words, while for sequence-level tasks, each sub-word is classified with the label of its sequence.

TASK	ORIG	LOW	MID-LOW	MID	MID-HIGH	HIGH	AUTO
POS	95.8±0.1	21.9±0.0	21.8±0.1	26.2±0.1	48.6±0.1	90.6±0.0	95.9±0.0
TOPIC	41.3±0.2	71.2±0.4	18.4±0.3	5.6±0.3	5.6±0.3	5.6±0.4	72.1±0.3

Table 10.3: **Detailed Monolingual Results** (ACC) for unfiltered (ORIG), low (L), mid-low (ML), mid (M), mid-high (MH), high (H), and automatically learned filters (AUTO), on the tasks of POS-tagging and TOPIC classification. Reported are the mean over five random initializations ± standard deviations. The same results plus the spectral profiles (frequency weightings) learned by AUTO are plotted in Figure 10.2.

Implementation All models are implemented using PyTorch v1.10 (Paszke et al., 2019) and NumPy v1.22 (Harris et al., 2020). Additionally, we use a modified version of the `torch-dct` package (Hu, 2018) to perform the DCT and IDCT. Visualizations are generated using matplotlib v3.5 (Hunter, 2007). Further, the code for reproducing our experiments is available at https://github.com/mainlp/spectral-probing.

10.6.3 Detailed Results

The following supplements the results presented in Section 10.3 with more detailed scores. Table 10.3 lists the exact scores for the monolingual English experiments on POS and TOPIC using the ORIG embeddings, the fixed-band filters and the learned AUTO filter. Table 10.4 lists the detailed scores for the ORIG and AUTO-filtered embeddings per language, in addition to the cross-lingual mean and standard deviation, across our seven tasks.

While the scores across random initializations never exceed a standard deviation of 1.0, it is important to note that scores may have higher variance across languages. This is to be expected due to different data across languages as well as pre-training availability. However we note that overall performance patterns (i.e., higher AUTO and relative task performance) are consistent across languages.

Task	Emb	DE	EN	ES	FR	JA	ZH	Avg
PoS	Orig	92.0±0.0	91.6±0.1	93.8±0.0	95.1±0.1	92.5±0.0	89.5±0.1	92.4±1.9
	Auto	92.1±0.1	91.6±0.0	93.9±0.0	95.1±0.0	92.7±0.1	89.8±0.1	92.5±1.8
Dep	Orig	79.0±0.1	78.4±0.1	81.2±0.1	83.0±0.1	79.6±0.1	70.6±0.2	78.6±4.3
	Auto	79.5±0.2	78.4±0.1	81.8±0.1	83.8±0.1	80.8±0.1	71.3±0.2	79.3±4.3
NER	Orig	90.3±0.0	85.3±0.1	90.4±0.0	88.1±0.1	84.1±0.1	89.5±0.0	88.0±2.7
	Auto	90.4±0.0	85.5±0.1	90.5±0.0	88.3±0.0	84.4±0.1	89.7±0.0	88.1±2.6
QA	Orig	63.2±0.2	64.5±0.1	64.1±0.2	63.9±0.3	61.0±0.8	60.7±0.8	62.9±1.6
	Auto	66.8±0.1	68.1±0.5	67.9±0.2	68.1±0.2	65.1±0.1	66.1±0.4	67.1±1.2
Senti	Orig	56.0±0.2	57.1±0.2	58.7±0.2	57.1±0.2	57.2±0.2	58.0±0.2	57.4±0.9
	Auto	64.0±0.2	63.5±0.5	65.4±0.2	64.7±0.5	65.4±0.5	62.7±0.3	64.3±1.1
Topic	Orig	22.7±0.1	26.8±0.4	22.9±0.3	24.0±0.3	22.9±0.5	43.3±0.4	27.1±8.1
	Auto	34.3±0.7	39.8±0.4	30.2±0.2	30.7±0.4	35.8±0.5	52.3±0.5	37.2±8.2
NLI	Orig	41.5±0.2	43.6±0.3	43.2±0.2	42.7±0.2	52.3±0.3	41.7±0.2	44.1±4.1
	Auto	51.3±0.8	56.4±0.7	54.3±0.7	54.5±0.5	67.2±0.4	53.8±1.0	56.3±5.6

Table 10.4: **Detailed Multilingual Results** (Acc) for unfiltered (Orig) and automatically learned filters (Auto) on the tasks of PoS-tagging, dependency relation classification (Dep), named entity recognition (NER), question answering (QA), sentiment analysis (Senti), Topic classification, and natural language inference (NLI). Each task covers the languages German (DE), English (EN), Spanish (ES), French (FR), Japanese (JA) and Chinese (ZH). Reported are the mean over five random initializations ± standard deviations as well as the mean over languages (Avg) ± the standard deviation across languages. The latter results are reported in Table 10.1, in addition to the spectral profiles (frequency weightings) learned by Auto in Figure 10.3.

Part V

Conclusion

Bridging the Divide 11

The previous chapters have demonstrated the importance of understanding Variety Space at a fundamental level, across the dimensions of typology, domain and higher-level tasks. Our framework of quantifying linguistic variation based on well-grounded qualitative formalisms not only contributes towards model interpretability, but also towards improving LM robustness and trustworthiness. Returning to our research questions from Section 1.3, we now summarize our findings.

11.1 Typological Variation

In Part II, we first turned our attention to typological variation. Focusing on its sub-component of syntax, we linked one of its qualitative formalisms, in the form of Universal Dependencies, to structural information in quantitative LM latent spaces, in order to elucidate:

RQ1 How is syntactic information from different typologies represented in data-driven latent spaces?

Understanding typological variation qualitatively is essential to ensuring interpretability that is linguistically grounded. We therefore first reviewed existing formalisms of syntax, before constructing high-specificity probing methods for extracting them from data-driven latent spaces.

RQ1.1 Which qualitative definitions exist for typological variation?

Our survey in Background Section 2.1.1 showed that even for this well-established variety dimension, its formalization lacks a definitive consensus. Most existing definitions are discrete (e.g., ISO codes), and do not capture typological features continuously. Measuring typology on a spectrum is especially important to characterizing under-resourced languages and dialects, as these are often not represented in discrete taxonomies, yet share features with their closely related, high-resource neighbors. Focusing on the underlying property of syntactic information in particular, we identified its importance to downstream natural language understanding tasks. To analyze the variability of this property, we built on the Universal Dependencies formalism, which characterizes syntax via sentence-level dependency structures, that can be applied across languages. The cross-lingual consistency of this formalism allowed us to capture more holistically how syntactic information for different language varieties is represented in continuous, data-driven latent spaces, agnostically of discrete language labels.

RQ1.2 *Does quantitative LM latent space contain sufficient typological information to extract fully directed and labeled dependency trees?*

Chapter 4 presented our high-specificity DEPPROBE, tailored to extract fully directed and labeled syntactic dependency trees. In addition to being the first to fully capture the UD dependency tree formalism, its linear nature make it possible to apply quantitative subspace comparison methods. As such, DEPPROBE not only recovers fully directed and labeled dependency trees, but can also quantify how similarly this information is represented across languages with respect to the specific properties of tree structure, depth and dependency relations. The different amounts of information recovered for each of these properties further reveals what a fully tuned parser actually learns on top of the host LM: e.g., long-range dependencies, as well as relations, which are rare and can take on a wide variety of surface forms. Overall, DEPPROBE allows us to extract syntactic information more effectively than prior approaches, reaching up to 73 LAS without any full-model fine-tuning, confirming that syntactic information following the full UD formalism can be recovered from LM latent space.

RQ1.3 *How well does syntactic probing predict the cross-lingual transferability of a full parser?*

In experiments covering 13 typologically diverse languages, we demonstrated that the subspace overlaps estimated using DEPPROBE are highly predictive of transferability across language varieties (Chapter 4). For the task of dependency parsing, it is able to identify the best source language for zero-shot transfer 94% of the time, outperforming competitive quantitative and qualitative baselines, as well as prior work. It is furthermore highly efficient, saving three orders of magnitude worth of fine-tuned parameters compared to training a full parser for each language combination. In terms of interpretability, our subspace comparisons additionally reveal that the most predictive type of syntactic information for downstream performance is relational information, compared to tree structure, or tree depth.

RQ1.4 How well does syntactic probing predict which LM is best suited for dependency parsing in a specific language?

The similarities measured in these typological variety subspaces not only correlate highly with transferability across language data, but also with the suitability of LMs as an initialization for training a parser for a given language (Chapter 5). Across 46 typologically and architecturally diverse LM-language pairs, DEPPROBE predicts the best LM choice 79% of the time using orders of magnitude less compute than training a full parser, and allows us to investigate reasons behind the performance characteristics of alternative LM architectures. This general "probing-to-rank" approach improves on the prior state-of-the-art for making these important modeling decisions, namely practitioner intuition.

11.2 Domain Variation

In Part III, we investigated how variation manifests in the dimension of domain. Qualitatively, we found it to be even less clearly defined than typology, thus benefiting from being quantified on a continuous spectrum, leading to the question:

RQ2 How does domain information manifest in data-driven latent spaces across languages?

Mirroring typology, we first investigated existing qualitative definitions of domain, as well as human intuitions of the property. Despite its less concise formalization compared to syntax, we found that domain information, in the form of genre, can be amplified in quantitative latent spaces to improve model transferability across languages.

RQ2.1 *Which qualitative definitions exist for domain variation?*

While we found domain not to be lacking qualitative definitions in traditional linguistics literature (Background Section 2.1.2), they typically describe a wide array of sub-dimensions, which are neither comprehensive, nor particularly overlapping (Biber, 1988). Equally in NLP, the large amount of prior work investigating cross-domain transferability either leaves out or remains ambiguous as to what is considered a domain, focusing on more specific sub-dimensions. Across these different properties, which combine to form domains, we found genre and topic to be of particular interest to NLP, as they are mostly orthogonal to typology and each other, while having large impacts on model performance across almost all task types. In practice, genre is typically linked to the source of a text, while the topic is defined as the subject matter, which can be expressed independently of genre. To first garner a better understanding of how these properties may be qualitatively grounded, we started by investigating:

RQ2.2 *To what extent can humans qualitatively identify domain from text alone, and how well does this align with machines?*

In Chapter 6, we examined human intuitions towards the concept of domain to ground how well we can expect to qualitatively define variation along this dimension. Focusing on genre and topic at the level of individual and multiple sentences, our study demonstrated that humans reach above-random agreement in identifying these properties in absence of guidelines, which enforce conformity. This indicated that genre and topic are not just hypothetical, but are encoded with some degree of consistency in human language understanding. Nonetheless, agreement was far from perfect, and across finer-grained categories, we observed that each property may be more realistically measured as continuous mixtures with some level of inherent human disagreement. As for the machine modeling of these properties, prior work has mainly focused on detecting genre and topic at the document level (Sharoff, 2007; Petrenz and Webber, 2011; Sharoff, 2021). We found that at our more granular level, additional context beyond a single sentence is crucial to disentangling highly similar genres and topics, and that human uncertainty closely correlates with model uncertainty.

Despite higher-than-random agreement across human annotations and models thereof, discretizing these properties may therefore be obscuring more intricate interactions in a continuous space. Scaling up our efforts to understand interactions of domain and typology by bridging qualitative genre signals with data-driven LM latent spaces, we next surveyed:

RQ2.3 *Can cross-lingual genre information be amplified in LM latent spaces using weak supervision?*

Combining the variety dimensions of typology and domain in a controlled way, we proposed several methods for leveraging the existing genre metadata in 200 treebanks of Universal Dependencies to extract genre information from the self-supervised latent spaces of multilingual LMs across 114 languages (Chapters 7 and 8). As the dataset is primarily designed for annotating typological properties consistently, these genre annotations are known to contain large amounts of noise (Nivre et al., 2020), only being available for entire treebanks, and rarely identifying the genre of individual sentences. Nonetheless, we were able to use this cross-lingual signal to map UD's 18 treebank-level genre labels to the instance level by proposing weakly-supervised clustering methods for amplifying latent genre information in the multilingual embeddings. Here, we found a bottom-up approach to be key—incrementally bootstrapping genre for data points with high certainty, before moving across languages, and to data with higher degrees of genre mixing. The resulting instance-level genre distribution provided a clearer picture of UD, confirming a previously hypothesized bias towards news-wire and Wikipedia data (Plank, 2016) at a larger scale, while simultaneously revealing more data points from under-resourced genres in the long-tail. This leads to the question of how to best apply these new findings, i.e.:

RQ2.4 *Can amplified genre guide our selection of cross-lingual training data from a significantly larger, more diverse pool?*

Examining whether controlling for one variety dimension can improve transferability in another, Chapter 8 applied our previous weakly-supervised genre amplification methods to the cross-lingual transfer of syntactic dependency parsers. Our experiments covered 12 extremely low-resource data settings, with some of the smallest treebanks in UD. We then further restricted the transfer setup to include no in-language data at all, selecting proxy training data based on knowledge of the target genre alone. Compared to typical baselines of using more data, which include the target genre in mixture, or using unmodified embedding similarity, our bootstrapping-based methods for latent genre amplification significantly outperformed all other methods, while being eight times more data efficient. These experiments highlight how performance across the typological variety dimension can be improved by leveraging domain.

11.3 Task Variation

Having established methods for extracting variety subspaces from LM latent spaces, and having further demonstrated their applicability to improving robustness on downstream tasks, Part IV took a step back to better understand the connection between Variety Space and tasks in NLP, by asking:

RQ3 **Can data-driven measures of linguistic variation be leveraged to quantify task similarity in an interpretable way?**

Once again, tackling this question first requires a deeper qualitative understanding of what a task even is, before moving on to attempts to quantify them. Building on the fact that NLP tasks rely on different mixtures of linguistic information to map an input to its output, we hypothesized that the overlap of relevant variety subspaces—which we showed to be measurable via their respective probes—corresponds to task similarity.

RQ3.1 What constitutes a task in NLP?

Although this question may appear trivial, it is worth investigating for the purposes of grounding our interpretability efforts. It has furthermore gained new importance during the ongoing paradigm shift from manually-engineered tasks to few-shot learning approaches. In Background Sections 2.2.2 and 2.2.3, we therefore offered an attempt at defining tasks as variations over output space, and follow this conclusion through to its implications on model robustness and trustworthiness. This definition further establishes a clear link between how linguistic variation (i.e., input variation), and our previously examined data-driven notions of Variety Space impact downstream task performance. Towards a fundamental understanding of how this connection comes to be, we next investigated the natural follow-up question:

RQ3.2 *When does task-specific linguistic information emerge during LM training?*

Using probes as interpretable metrics of linguistic variation heavily relies on the automatically learned representational spaces of LMs. To establish the trustworthiness of this approach, we examined whether and when linguistic information arises in LMs over the course of their training (Chapter 9). By leveraging information-theoretic probing, we extracted subspaces with a higher consistency than standard linear probes. Applied across an LM's training from random initialization to completion, this approach allowed for comparisons of how linguistic information emerges and shifts over time. Our experiments showed that task-relevant information arises consistently during LM training, and result in representations which match linguistic intuitions (e.g., layer depth ↔ complexity). However, contrary to prior works, we also identified that these patterns of separation across tasks only emerge at the later stages of training, with most information gains happening in a critical learning phase early on, highlighting opportunities for training LMs with fewer training data.

RQ3.3 *Which linguistic information is shared across tasks, and how do their subspaces interact across LM training time?*

The use of high-consistency probes not only allowed us to compare representational subspaces across time, but, for the first time, also across tasks. By treating tasks as mixtures over different variety dimensions with different ratios, we were able to show that, while there is a high degree of task specialization in LMs at the end of training, tasks that require intuitively similar linguistic information, also share more representational overlap. This observation holds across time, and is particularly prominent during the early critical learning phase, potentially due to linguistic knowledge sharing being more beneficial while the LM is under-trained. In the final step, we verified these cross-task similarities across more languages:

RQ3.4 How can the same task be characterized consistently across different languages?

While information-theoretic probing allows us to compare representational subspaces over time within one model, it is still too specific to the exact LM weights to allow for comparisons across models or languages. As such, Chapter 10 presented Spectral Probing, which leverages the phenomenon of information consistency over time. It allowed us to characterize tasks via their spectral frequency profiles, and demonstrated how LMs learn representations, which spread relevant information across their embeddings in a way which matches linguistic intuitions (e.g., short-range parts-of-speech, long-range sentiment). Importantly, we showed how spectral profiles are consistent enough to allow for comparisons across tasks and languages, displaying distinctive characteristics for different task types, but high consistency, even across typologically distant languages.

12 Outlook

The inherent variability of natural language necessitates controlling for at least some variety dimensions to ensure mutual intelligibility—both for humans and machines. Therefore, we expect the overall topic of measuring variation to remain relevant for the foreseeable future. Of course, despite the broad nature of our general framework for quantifying variation, specific methods will have to be adapted in tandem with the rapid evolution of modern NLP model architectures. For future work using probes to quantify variation, we therefore see two main avenues, mirroring the categorization of high specificity and high consistency probes from Section 3.3.

High-specificity Probes for Larger LMs From the angle of specificity, the rise of increasingly large LM architectures requires new probing methodologies, which go beyond the linear methods explored in this work. While some linguistic properties, such as spectral time-dependencies, are probably relatively universal, extracting variety subspaces from these deep and highly non-linear models likely requires more expressive, architecture-specific probes. Even with smaller architectures, we already observe issues probing latent representations of decoder-only models, as they appear to contain less relevant information, but nonetheless perform well on downstream tasks (Wang et al., 2022). Research on probing intermediate representations of such generative models is ongoing, and has focused on decoding latent embeddings into the output space pre-maturely to probe for certain linguistic behaviors (Nostalgebraist, 2020; Belrose et al., 2023). However, a consensus on how to reformulate linguistic probing tasks into a generative format has not yet been reached, and it furthermore remains unclear how much actual linguistic information versus spurious correlations these iterative inference methods are detecting. Identifying sub-networks using modular deep learning approaches could allow for the extraction of more complex linguistic subspaces (Ruder et al., 2022), however their non-linearity makes them difficult to compare, limiting their use as measures for, e.g., cross-task similarity.

Model weights and intermediate latent representations of state-of-the-art LMs trained by private entities may further be unavailable to end-users and researchers, making their analysis difficult and limiting their trustworthiness. While prompts may reveal some linguistic information present in these models, they can only be consolidated into the proxy measure of performance. Furthermore, prompting suffers from high output variability itself, and is difficult to reproduce for research purposes (Salinas and Morstatter, 2024). Despite these limitations, establishing model trustworthiness is crucial, and as such, we see targeted diagnostic benchmarks as an essential method to better discerning knowledge of a model's functional capacity (Litschko et al., 2023), and to understand which linguistic capabilities the model is actually employing (Schlangen, 2021).

Causal Interventions An issue in the aforementioned approaches is their reliance on correlations between model representation and linguistic properties. While methods, such as information-theoretic probing, are able to better distinguish between random and efficient representations, interventions in the form of removing this information would be necessary to confirm a causal link. Probing work has begun to use interventions to verify correlatory findings by, e.g., masking tokens relevant to a linguistic property (Lasri et al., 2022; Hanna et al., 2023), or preventing attention heads from learning connections akin to syntactic dependencies (Chen et al., 2024), however there is no consensus yet as for which methods to use across all types of linguistic properties. Furthermore, interventions often require re-training, which is prohibitively expensive or impossible for large and/or closed-source LMs. For subspace-based probing approaches, we nonetheless see opportunities for leveraging extracted variety spaces as interventions on an LM during inference. In combination with diagnostic benchmarks, they could help verify whether a linguistic skill necessary to solving a diagnostic task (e.g., syntactic understanding) is reduced if the relevant subspace has been nulled out (Ravfogel et al., 2020).

Cross-model Consistency Beyond, architecture-specific considerations, enabling trustworthiness of LMs across language varieties and tasks will require a better general understanding of their knowledge of origin, i.e., a consistent explanation for how they acquire their capabilities (Litschko et al., 2023). This lies at the core of trust in statistical ML models for language, considering the fact that almost the entirety of contemporary NLP is based on the distributional hypothesis (Harris, 1954; Firth, 1957). While LM capabilities continue improving across benchmarks and qualitatively in-practice, it remains unclear how modeling the statistical distribution of large datasets, i.e., the probability of the next token, leads to representations, which encode highly linguistically relevant information, and contribute to sophisticated model behaviors. This is highlighted even more by the fact that it is still not understood how training the same architecture on the same data, but with different random initializations, leads to wildly different model behaviors (Sellam et al., 2022; Hu et al., 2023; Chen et al., 2024), or, similarly, why slightly augmented prompts can lead to large performance differences (Leidinger et al., 2023; Salinas and Morstatter, 2024). In terms of probing, this issue connects to the need for the ability to compare how the same linguistic information is represented across LMs. The goal of future methods should therefore be to aim for methods which can extract highly specific linguistic information from complex model architectures, while remaining consistent and comparable with each other regardless of their host model.

Expansion to More Variety Dimensions Finally, it is important to consider variety dimensions complementary to those explored in this work. Our experiments have demonstrated the viability of probing for subspaces of formalized properties, such as syntax, as well as less-formalized ones like genre. With the application domains of LMs becoming increasingly broad and complex, we anticipate that future work will require a more human-centric focus on pragmatic information, in addition to semantic information grounded in different cultural contexts. A better understanding of these variety dimensions will be essential for establishing model trustworthiness in higher-stake, real-world scenarios. As most additional dimensions of variation will be even less clearly defined, it is important to establish qualitative definitions of each new dimension, which take human uncertainty and continuity into account, before designing quantitative methods to probe for these properties in LM latent spaces. This process is crucial for ensuring interpretability in a well-grounded context, that is aware of its limitations. In parallel, we see automatically learned representations as critical for scaling to finer-grained variety dimensions, and for modeling complex, high-dimensional interactions thereof. With traditional boundaries between NLP tasks breaking down, the field requires continuous *and* interpretable measures to maintain trustworthiness. Towards this purpose, we hope that our proposed framework of bridging qualitative and data-driven measures will contribute to a mutually beneficial cycle which bolsters our understanding of variation in language.

Backmatter

Bibliography

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Short papers*.

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6429–6438.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Nasir Ahmed, T. Natarajan, and Kamisetty Ramamohan Rao. 1974. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93.

Sohail Akhtar, Valerio Basile, and Viviana Patti. 2021. Whose opinions matter? perspective-aware models to identify opinions of hate speech victims in abusive language detection. *ArXiv preprint*, abs/2106.15896.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Maria Jesus Aranzabe, Aitziber Atutxa, Kepa Bengoetxea, Arantza Diaz de Ilarraza, Iakes Goenaga, Koldo Gojenola, and Larraitz Uria. 2015. Automatic conversion of the Basque dependency treebank to universal dependencies. In *Proceedings of the fourteenth international workshop on treebanks an linguistic theories (TLT14)*, pages 233–241.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Masayuki Asahara, Hiroshi Kanayama, Takaaki Tanaka, Yusuke Miyao, Sumire Uematsu, Shinsuke Mori, Yuji Matsumoto, Mai Omura, and Yugo Murawaki. 2018. Universal Dependencies version 2 for Japanese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Guy Aston and Lou Burnard. 1998. *The BNC Handbook: Exploring the British National Corpus with SARA*. Edinburgh textbooks in empirical linguistics. Edinburgh University Press.

Harald Baayen. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*, 1 edition. Cambridge University Press, Cambridge.

Maria Barrett, Max Müller-Eberstein, Elisa Bassignana, Amalie Brogaard Pauli, Mike Zhang, and Rob van der Goot. 2024. Can humans identify domains? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, Torino, Italy. European Language Resources Association.

Elisa Bassignana, Max Müller-Eberstein, Mike Zhang, and Barbara Plank. 2022. Evidence > intuition: Transferability estimation for encoder selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4218–4227, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Elisa Bassignana and Barbara Plank. 2022. CrossRE: A cross-domain dataset for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3592–3604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Emi Baylor, Esther Ploeger, and Johannes Bjerva. 2024. Multilingual gradient word-order typology from Universal Dependencies. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 42–49, St. Julian's, Malta. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *Computing Research Repository*, arxiv:2303.08112. Version 4.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal Dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin, Germany. Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wentau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava, and Dipti Sharma. 2018. Universal Dependency parsing for Hindi-English codeswitching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 987–998, New Orleans, Louisiana. Association for Computational Linguistics.

Douglas Biber. 1988. *Variation across speech and writing*. Cambridge University Press, Cambridge.

Douglas. Biber. 1995. *Dimensions of register variation : a crosslinguistic comparison*. Cambridge University Press, Cambridge.

Douglas Biber and Susan Conrad. 2009. *Register, Genre, and Style*, 1 edition. Cambridge Textbooks in Linguistics. Cambridge University Press, Cambridge.

Douglas Biber and Susan Conrad. 2019. *Register, genre, and style*. Cambridge University Press.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu

Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Computing Research Repository*, arxiv:2304.01373. Version 1.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan. The COLING 2016 Organizing Committee.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Pavel Blinov. 2021. RoBERTa-base Russian. `https://huggingface.co/blinoff/roberta-base-russian-v0`. Accessed 4th January, 2022.

Carlo Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Cristina Bosco, Felice Dell'Orletta, Simonetta Montemagni, Manuela Sanguinetti, and Maria Simi. 2014. The EVALITA 2014 dependency parsing task. In *EVALITA 2014 Evaluation of NLP and Speech Tools for Italian*, pages 1–8. Pisa University Press.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Anouck Braggaar and Rob van der Goot. 2021. Challenges in annotating and parsing spoken, code-switched, Frisian-Dutch data. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 50–58, Kyiv, Ukraine. Association for Computational Linguistics.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a german corpus. *Research on language and computation*, 2(4):597–620.

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

Flavio M Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. 2020. Udante: First steps towards the universal dependencies treebank of dante's latin works. In *Seventh Italian Conference on Computational Linguistics*, pages 1–7. CEUR-WS. org.

Özlem Çetinoğlu and Çağrı Çöltekin. 2019. Challenges of annotating a code-switching treebank. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 82–90, Paris, France. Association for Computational Linguistics.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. 2024. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. 2020. Pretrained language model embryology: The birth of ALBERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6813–6828, Online. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228.

Euisun Choi and Chulhee Lee. 2003. Feature extraction based on the Bhattacharyya distance. *Pattern Recognition*, 36:1703–1709.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. 2018. Building Universal Dependency treebanks in Korean. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pretrained language models. In *International Conference on Learning Representations*.

Alessandra Teresa Cignarella, Cristina Bosco, and Paolo Rosso. 2019. Presenting TWITTIRÒ-UD: An Italian Twitter treebank in Universal Dependencies. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 190–197, Paris, France. Association for Computational Linguistics.

Bernard Comrie. 1981. *Language universals and linguistic typology : syntax and morphology.* Basil Blackwell, Oxford.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018a. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018b. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for Chinese BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. Cost-effective selection of pretraining data: A case study of pretraining BERT on social media. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1675–1681, Online. Association for Computational Linguistics.

David Dale. 2021. RuBERT-tiny: A small and fast BERT for Russian. `https://habr.com/ru/post/562064/`. Accessed 4th January, 2022.

Hal Daumé III and Lyle Campbell. 2007. A Bayesian model for discovering typological implications. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 65–72, Prague, Czech Republic. Association for Computational Linguistics.

Joe Davison. 2020. Zero-Shot Learning in Modern NLP. Accessed December 4th, 2020.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland. European Language Resources Association (ELRA).

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Mathieu Dehouck and Pascal Denis. 2019. Phylogenic multi-lingual dependency parsing. 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 192–203, Minneapolis, Minnesota. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Melvil Dewey. 1952. *Dewey decimal classification & relative index*, 15. ed. edition. Forest Press, New York.

Elisa Di Nuovo, Cristina Bosco, Alessandro Mazzei, and Manuela Sanguinetti. 2019. Towards an italian learner treebank in universal dependencies. In *6th Italian Conference on Computational Linguistics, CLiC-it 2019*, volume 2481, pages 1–6. CEUR-WS.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. 2018. Data conversion and consistency of monolingual corpora: Russian UD treebanks. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 53–66.

Matthew S. Dryer. 1992. The Greenbergian word order correlations. *Language*, 68(1):81–138.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2018. Crowdsourcing semantic label propagation in relation classification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 16–21, Brussels, Belgium. Association for Computational Linguistics.

Puneet Dwivedi and Guha Easha. 2017. Universal Dependencies for Sanskrit. *International Journal of Advance Research, Ideas and Innovations in Technology*, 3(4).

Hanne Eckhoff, Kristin Bech, Gerlof Bouma, Kristine Eide, Dag Haug, Odd Einar Haugen, and Marius Jøhndal. 2018. The PROIEL treebank family: a standard for early attestations of Indo-European languages. *Language Resources and Evaluation*, 52(1):29–65.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards, B*, 71:233–240.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2014. Diffusion of lexical change in social media. *PloS one*, 9(11):e113114.

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. 2021. Not all models localize linguistic knowledge in the same place: A layer-wise probing on BERToids' representations. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 375–388, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mário Figueiredo. 2001. Adaptive sparseness using jeffreys prior. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

J. R. Firth. 1957. *Studies in linguistic analysis.* Special volume of the Philological Society. Basil Blackwell, Oxford.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5).

Lucie Flek. 2020. Returning the N to NLP: Towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7828–7838, Online. Association for Computational Linguistics.

Fonticons. 2021. Font Awesome Icons. CC-BY 4.0 License.

Mara Franzen. 2022. Alternatives to the dewey decimal system.

Marcos Garcia. 2016. Universal dependencies guidelines for the Galician-TreeGal treebank. Technical report, Technical Report, LyS Group, Universidade da Coruna.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Frances Gillis-Webber and Sabine Tittel. 2020. A framework for shared agreement of language tags beyond ISO 639. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3333–3339, Marseille, France. European Language Resources Association.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47, Portland, Oregon, USA. Association for Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Anna Gooding-Call. 2021. Racism in the dewey decimal system.

Joseph H. Greenberg. 1966. *Universals of language*, second edition. MIT Press, Cambridge.

Ralph Grishman and Richard Kittredge. 1986. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Taylor & Francis, New York.

Bruno Guillaume, Marie-Catherine de Marneffe, and Guy Perrier. 2019. Conversion et améliorations de corpus du Français annotés en Universal Dependencies. *Traitement Automatique des Langues*, 60(2):71–95.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational*

Linguistics, pages 8342–8360, Online. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Conference on Spoken Language Translation*, Seattle, Washington D.C. International Workshop on Spoken Language Translation.

Jan Hajič, Otakar Smrž, Petr Zemánek, Petr Pajas, Jan Šnaidauf, Emanuel Beška, Jakub Krácmar, and Kamila Hassanová. 2009. Prague Arabic dependency treebank 1.0.

Jihun Hamm and Daniel D. Lee. 2008. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 376–383, New York, NY, USA. Association for Computing Machinery.

Harald Hammarström. 2015. Glottolog: a free, online, comprehensive bibliography of the world's languages. In *3rd International Conference on Linguistic and Cultural Diversity in Cyberspace*, pages 183–188. UNESCO.

Michael Hanna, Roberto Zamparelli, and David Mareček. 2023. The functional relevance of probed information: A case study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 835–848, Dubrovnik, Croatia. Association for Computational Linguistics.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Dag TT Haug and Marius Jøhndal. 2008. Creating a parallel treebank of the old Indo-European Bible translations. In *Proceedings of the second workshop on language technology for cultural heritage data (LaTeCH 2008)*, pages 27–34.

John A. Hawkins. 1983. *Word order universals.* Quantitative Analyses of Linguistic Structure. Academic Press, San Diego, California.

David G. Hays. 1979. Applications. In *17th Annual Meeting of the Association for Computational Linguistics*, pages 89–89, La Jolla, California, USA. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Oliver Hellwig, Salvatore Scarlata, Elia Ackermann, and Paul Widmer. 2020. The treebank of vedic Sanskrit. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5137–5146, Marseille, France. European Language Resources Association.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Barbora Hladká, Jan Hajič, Jirka Hana, Jaroslava Hlaváčová, Jiří Mírovskỳ, and Jan Raab. 2008. The Czech academic corpus 2.0 guide. *The Prague Bulletin of Mathematical Linguistics*, 89(1):41–96.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.

Antti Honkela and Harri Valpola. 2004. Variational learning and bits-back coding: an information-theoretic view to bayesian learning. *IEEE transactions on Neural Networks*, 15(4):800–810.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 752–762, Beijing, China. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

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

Michael Y. Hu, Angelica Chen, Naomi Saphra, and Kyunghyun Cho. 2023. Latent state models of training dynamics. *Transactions on Machine Learning Research*.

Ziyang Hu. 2018. Discrete Cosine Transform for PyTorch. `https://github.com/zh217/torch-dct`.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Anton Karl Ingason, Eiríkur Rögnvaldsson, Einar Freyr Sigurosson, and Joel C. Wallenberg. 2020. The Faroese parsed historical corpus. CLARIN-IS, Stofnun Árna Magnússonar.

Vojtěch Jarník. 1930. O jistém problému minimálním.(z dopisu panu o. boruvkovi). *Práce moravské přírodovědecké společnosti*, 6(4):57–63.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Hadi S Jomaa, Lars Schmidt-Thieme, and Josif Grabocka. 2021. Dataset2vec: Learning dataset meta-features. *Data Mining and Knowledge Discovery*, 35(3):964–985.

Jussi Karlgren and Douglass Cutting. 1994. Recognizing text genres with simple metrics using discriminant analysis. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*.

Constantinos Karouzos, Georgios Paraskevopoulos, and Alexandros Potamianos. 2021. UDALM: Unsupervised domain adaptation through language modeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2579–2590, Online. Association for Computational Linguistics.

Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. 2020. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *ArXiv preprint*, abs/2009.10277.

Brett Kessler, Geoffrey Nunberg, and Hinrich Schutze. 1997. Automatic detection of text genre. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 32–38, Madrid, Spain. Association for Computational Linguistics.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Johannes Kiesel, Milad Alshomary, Nailia Mirzakhmedova, Maximilian Heinrich, Nicolas Handke, Henning Wachsmuth, and Benno Stein. 2023. SemEval-2023 task 4: ValueEval: Identification of human values behind arguments. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2287–2303, Toronto, Canada. Association for Computational Linguistics.

Kiyoung Kim. 2020. Pretrained language models for korean. `https://github.com/kiyoungkim1/LMkor`.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computing Research Repository*, arxiv:1412.6980. Version 9.

Durk P Kingma, Tim Salimans, and Max Welling. 2015. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Kia Kirstein Hansen, Maria Barrett, Max Müller-Eberstein, Cathrine Damgaard, Trine Eriksen, and Rob van der Goot. 2023. DanTok: Domain beats language for Danish social media POS tagging. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 271–279, Tórshavn, Faroe Islands. University of Tartu Library.

Richard Kittredge. 1982. Sublanguages. *American Journal of Computational Linguistics*, 8(2):79–84.

Richard Kittredge and Ralph Grisham. 1986. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum Associates.

Richard Kittredge and John Lehrberger. 1982. *Sublanguage: Studies of language in restricted semantic domains*. Walter de Gruyter.

Andrew V Knyazev and Merico E Argentati. 2002. Principal angles between subspaces in an a-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Greek-bert: The greeks visiting sesame street. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, page 110–117, New York, NY, USA. Association for Computing Machinery.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. 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 2755–2768, Hong Kong, China. Association for Computational Linguistics.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, Online. Association for Computational Linguistics.

Taja Kuzman, Peter Rupnik, and Nikola Ljubešić. 2022. The GINCO training dataset for web genre identification of documents out in the wild. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1584–1594, Marseille, France. European Language Resources Association.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

David Lee. 2002. Genres, registers, text types, domains and styles: clarifying the concepts and navigating a path through the bnc jungle. In *Teaching and Learning by Doing Corpus Analysis*, pages 245–292. Brill.

David Yong Wey Lee. 2001. Genres, registers, text types, domains and styles: Clarifying the concepts and navigating a path through the bnc jungle. *Language Learning & Technology*, 5:37–72.

John Lee, Herman Leung, and Keying Li. 2017. Towards Universal Dependencies for learner Chinese. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 67–71, Gothenburg, Sweden. Association for Computational Linguistics.

Yong-Bae Lee and Sung Hyon Myaeng. 2002. Text genre classification with genre-revealing and subject-revealing features. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 145–150, New York, NY, USA. Association for Computing Machinery.

Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. 2023. The language of prompting: What linguistic properties make a prompt successful? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9210–9232, Singapore. Association for Computational Linguistics.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig. 2015. Ethnologue: Languages of the world, eigh- teenth edition. SIL International, Dallas, Texas.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tomasz Limisiewicz and David Mareček. 2021. Introducing orthogonal constraint in structural probes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*

(Volume 1: Long Papers), pages 428–442, Online. Association for Computational Linguistics.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Jian Pei, and Liang Zhao. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ArXiv*.

Robert Litschko, Max Müller-Eberstein, Rob van der Goot, Leon Weber-Genzel, and Barbara Plank. 2023. Establishing trustworthiness: Rethinking tasks and model evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Singapore. Association for Computational Linguistics.

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, Valencia, Spain. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. 2018. Parsing tweets into Universal Dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long Papers), pages 965–975, New Orleans, Louisiana. 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. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021a. Probing across time: What does RoBERTa know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 820–842, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021b. Crossner: Evaluating cross-domain named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13452–13460.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Christos Louizos, Karen Ullrich, and Max Welling. 2017. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mikko Luukko, Aleksi Sahala, Sam Hardwick, and Krister Lindén. 2020. Akkadian treebank for early Neo-Assyrian royal inscriptions. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 124–134, Düsseldorf, Germany. Association for Computational Linguistics.

Olga Lyashevskaya, Angelika Peljak-Łapińska, and Daria Petrova. 2017. UD_Belarusian-HSE. `https://github.com/UniversalDependencies/UD_Belarusian-HSE`.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535, Copenhagen, Denmark. Association for Computational Linguistics.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden - making a swedish BERT. *CoRR*, abs/2007.01658.

Christopher D. Manning and Hinrich Schütze. 2003. *Foundations of statistical natural language processing*, sixth edition. MIT Press, Cambridge, Massachussets.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

David McClosky. 2010. *Any domain parsing: automatic domain adaptation for natural language parsing.* Ph.D. thesis, Brown University.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav

Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Ailsa Meechan-Maddon and Joakim Nivre. 2019. How to parse low-resource languages: Cross-lingual parsing, target language annotation, or both? In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 112–120, Paris, France. Association for Computational Linguistics.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Mistral. 2022. Mistral — Large Scale Language Modeling Made Easy. `https://nlp.stanford.edu/mistral/`.

Maria Mitrofan, Verginica Barbu Mititelu, and Grigorina Mitrofan. 2019. MoNERo: a biomedical gold standard corpus for the Romanian language. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 71–79, Florence, Italy. Association for Computational Linguistics.

Ines Montani, Matthew Honnibal, Matthew Honnibal, Sofie Van Landeghem, Adriane Boyd, Henning Peters, Paul O'Leary McCann, jim geovedi, Jim O'Regan, Maxim Samsonov, György Orosz, Daniël de Kok, Marcus Blättermann, Duygu Altinok, Raphael Mitsch, Madeesh Kannan, Søren Lind Kristiansen, Edward, Raphaël Bournhonesque, Lj Miranda, Peter Baumgartner, Richard Hudson, Explosion Bot, Roman, Leander Fiedler, Ryn Daniels, Wannaphong Phatthiyaphaibun, Grégory Howard, and Yohei Tamura. 2023. spaCy: v3.5.2.

Steven Moran, Daniel McCloy, and Richard Wright. 2014. PHOIBLE online. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Ari S. Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. *Computing Research Repository*, arxiv:1806.05759. Version 3.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Atreyee Mukherjee, Sandra Kübler, and Matthias Scheutz. 2017. Creating POS tagging and dependency parsing experts via topic modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 347–355, Valencia, Spain. Association for Computational Linguistics.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021a. Genre as weak supervision for cross-lingual dependency parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4786–4802, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021b. How universal is genre in Universal Dependencies? In *Proceedings of the 20th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2021)*, pages 69–85, Sofia, Bulgaria. Association for Computational Linguistics.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022a. Probing for labeled dependency trees. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7711–7726, Dublin, Ireland. Association for Computational Linguistics.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022b. Sort by structure: Language model ranking as dependency probing.

In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1307, Seattle, United States. Association for Computational Linguistics.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022c. Spectral probing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7730–7741, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208, Singapore. Association for Computational Linguistics.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. 2021. Understanding the failure modes of out-of-distribution generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Cuong Nguyen, Tal Hassner, Matthias W. Seeger, and Cédric Archambeau. 2020. LEEP: A new measure to evaluate transferability of learned representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7294–7305. PMLR.

Dong Nguyen, Laura Rosseel, and Jack Grieve. 2021. On learning and representing social meaning in NLP: a sociolinguistic perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 603–612, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9131–9143, Online. Association for Computational Linguistics.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Aljoscha Burchardt, Marie Candito, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Silvie Cinková, Çağrı Çöltekin, Miriam Connor, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Marhaba Eli, Ali Elkahky, Tomaž Erjavec, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Pinkey Nainwani, Anna Nedoluzhko, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Petya

Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Livy Real, Siva Reddy, Georg Rehm, Larissa Rinaldi, Laura Rituma, Rudolf Rosa, Davide Rovati, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Antonio Stella, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zhuoran Yu, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.0 – CoNLL 2017 shared task development and test data. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Nostalgebraist. 2020. Interpreting GPT: The logit lens. Accessed 18th May, 2024.

Mai Omura, Aya Wakasa, and Masayuki Asahara. 2023. Universal dependencies for japanese based on long-unit words by ninjal. *Journal of Natural Language Processing*, 30(1):4–29.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Robert Östling, Carl Börstell, Moa Gärdenfors, and Mats Wirén. 2017. Universal Dependencies for Swedish Sign Language. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 303–308, Gothenburg, Sweden. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649, Valencia, Spain. Association for Computational Linguistics.

Vishakh Padmakumar, Leonard Lausen, Miguel Ballesteros, Sheng Zha, He He, and George Karypis. 2022. Exploring the role of task transferability in large-scale multi-task learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2542–2550, Seattle, United States. Association for Computational Linguistics.

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting* *of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won-Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Tae Hwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Eunjeong Lucy Park, Alice Oh, Jung-Woo Ha, and Kyunghyun Cho. 2021. KLUE: korean language understanding evaluation. *CoRR*, abs/2105.09680.

Niko Partanen, Rogier Blokland, KyungTae Lim, Thierry Poibeau, and Michael Rießler. 2018. The first Komi-Zyrian Universal Dependencies treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 126–132, Brussels, Belgium. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Agnieszka Patejuk and Adam Przepiórkowski. 2018a. *From Lexical Functional Grammar to Enhanced Universal Dependencies: Linguistically informed treebanks of Polish.* Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Agnieszka Patejuk and Adam Przepiórkowski. 2018b. *From Lexical Functional Grammar to Enhanced Universal Dependencies: Linguistically informed treebanks of Polish.* Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Xingchao Peng, Yichen Li, and Kate Saenko. 2020. Domain2vec: Domain embedding for unsupervised domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 756–774. Springer.

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

Philipp Petrenz and Bonnie Webber. 2011. Squibs: Stable classification of text genres. *Computational Linguistics*, 37(2):385–393.

Philipp Petrenz and Bonnie Webber. 2012. Label propagation for fine-grained cross-lingual genre classification. In *xLite: Cross-Lingual Technologies (NIPS 2012 Workshop)*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Barbara Plank. 2011. Domain adaptation for parsing. University of Groningen.

Barbara Plank. 2016. What to do about non-standard (or non-canonical) language in nlp. In *KONVENS*, Bochum, Germany.

Barbara Plank. 2022. The “problem” of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland. Association for Computational Linguistics.

Nick Pogrebnyakov and Shohreh Shaghaghian. 2021. Predicting the success of domain adaptation in text similarity. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 206–212, Online. Association for Computational Linguistics.

Rhitabrat Pokharel and Ameeta Agrawal. 2023. Estimating semantic similarity between in-domain and out-of-domain samples. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 409–416, Toronto, Canada. Association for Computational Linguistics.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural*

Language Learning, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Robert Clay Prim. 1957. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401.

Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal Dependencies for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 163–172, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Investigating language universal and specific properties in word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Computing Research Repository*, arxiv:1706.05806. Version 2.

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

Taraka Rama and Sowmya Vajjala. 2017. A Telugu treebank based on a grammar book. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 119–128, Prague, Czech Republic.

Loganathan Ramasamy and Zdeněk Žabokrtský. 2012. Prague dependency style treebank for Tamil. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1888–1894, Istanbul, Turkey. European Language Resources Association (ELRA).

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. Domain divergences: A survey and empirical analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849, Online. Association for Computational Linguistics.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine learning*, 34:151–175.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Ines Rehbein and Felix Bildhauer. 2017. Data point selection for genre-aware parsing. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 95–105, Prague, Czech Republic.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Jan Rijkhoff. 2007. Linguistic typology: a short history and some current issues. *Tidsskrift for sprogforskning*, 5(1):1–18.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Rudolf Rosa. 2015. Parsing natural language sentences by semi-supervised methods. *CoRR*, abs/1506.04897.

R. Rosenfeld. 2000. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Sebastian Ruder, Jonas Pfeiffer, and Ivan Vulić. 2022. Modular and parameter-efficient fine-tuning for NLP models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 23–29, Abu Dubai, UAE. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Jack Rueter and Francis Tyers. 2018. Towards an open-source universal-dependency treebank for Erzya. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 106–118, Helsinki, Finland. Association for Computational Linguistics.

Pegah Safari, Mohammad Sadegh Rasooli, Amirsaeid Moloodi, and Alireza Nourian. 2022. The Persian dependency treebank made universal. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7078–7087, Marseille, France. European Language Resources Association.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Abel Salinas and Fred Morstatter. 2024. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance. *Computing Research Repository*, arxiv:2401.03729. Version 3.

Alessio Salomoni. 2019. UD_German-LIT. `https://github.com/UniversalDependencies/UD_German-LIT`.

Tanja Samardžić, Ximena Gutierrez-Vasques, Rob van der Goot, Max Müller-Eberstein, Olga Pelloni, and Barbara Plank. 2022. On language spaces, scales and cross-lingual transfer of UD parsers. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 266–281, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Stephanie Samson and Cagrı Cöltekin. 2020. UD_Tagalog-TRG. `https://github.com/UniversalDependencies/UD_Tagalog-TRG`.

Evan Sandhaus. 2008. The new york times annotated corpus.

Marta Sandri, Elisa Leonardelli, Sara Tonelli, and Elisabetta Jezek. 2023. Why don't you do it right? analysing annotators' disagreement in subjective tasks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2428–2441, Dubrovnik, Croatia. Association for Computational Linguistics.

Manuela Sanguinetti, Cristina Bosco, Lauren Cassidy, Özlem Çetinoğlu, Alessandra Teresa Cignarella, Teresa Lynn, Ines Rehbein, Josef Ruppenhofer, Djamé Seddah, and Amir Zeldes. 2020. Treebanking user-generated content: A proposal for a unified representation in Universal Dependencies. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5240–5250, Marseille, France. European Language Resources Association.

Manuela Sanguinetti, Cristina Bosco, Alberto Lavelli, Alessandro Mazzei, Oronzo Antonelli, and Fabio Tamburini. 2018. PoSTWITA-UD: an Italian Twitter treebank in Universal Dependencies. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. Adversarial training for cross-domain Universal Dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 71–79, Vancouver, Canada. Association for Computational Linguistics.

Danielle Saunders. 2022. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *Journal of Artificial Intelligence Research*, 75:351–424.

Sber Devices. 2021. ruRoBERTa-large. `https://huggingface.co/sberbank-ai/ruRoberta-large`. Accessed 4th January, 2022.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Computing Research Repository*, arxiv:2211.05100. Version 3.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? *Computing Research Repository*, arxiv:2304.15004. Version 1.

David Schlangen. 2021. Targeting the benchmark: On methodology in current natural language processing research. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 670–674, Online. Association for Computational Linguistics.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Shaked

Greenfeld, and Reut Tsarfaty. 2021. Alephbert: A hebrew large pre-trained language model to start-off your hebrew NLP application with. *CoRR*, abs/2104.04052.

Satoshi Sekine. 1997. The domain dependence of parsing. In *Fifth Conference on Applied Natural Language Processing*, pages 96–102, Washington, DC, USA. Association for Computational Linguistics.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. 2022. The multiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*.

Serge Sharoff. 2007. Classifying web corpora into domain and genre using automatic feature identification. In *Proceedings of the 3rd Web as Corpus Workshop*, pages 83–94.

Serge Sharoff. 2021. Genre annotation for the web: Text-external and text-internal perspectives. *Register Studies*, 3(1):1–32.

Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning:"Taiga" syntax tree corpus and parser. In *Proceedings of "CORPORA-2017" International Conference*, pages 78–84.

Mo Shen, Ryan McDonald, Daniel Zeman, and Peng Qi. 2016a. UD_Chinese-GSD. `https://github.com/UniversalDependencies/UD_Chinese-GSD`.

Mo Shen, Ryan McDonald, Daniel Zeman, and Peng Qi. 2016b. UD_Chinese-GSD. `https://github.com/UniversalDependencies/UD_Chinese-GSD`.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Pranaydeep Singh, Gorik Rutten, and Els Lefever. 2021. A pilot study for BERT language modelling and morphological analysis for ancient and medieval Greek. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 128–137, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, pages 1–9.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018a. 82 treebanks, 34 models: Universal Dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018b. An investigation of the interactions between pre-trained word embeddings, character models and POS tags in dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2711–2720, Brussels, Belgium. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 682–686, Portland, Oregon, USA. Association for Computational Linguistics.

Miloš Stanojević and Shay B. Cohen. 2021. A root of a problem: Optimizing single-root dependency parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10540–10557, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Achim Stein and Sophie Prévost. 2013. Syntactic annotation of medieval texts. *New methods in historical corpora*, 3:275.

Benno Stein and Sven Meyer Zu Eissen. 2006. Distinguishing topic from genre. In *Proceedings of the 6th International Conference on Knowledge Management (I-KNOW 06)*, pages 449–456. Citeseer.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Will Styler. 2011. The Enronsent Corpus. *Boulder: University of Colorado at Boulder Institute of Cognitive Science.*

Sara Stymne. 2020. Cross-lingual domain adaptation for dependency parsing. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 62–69, Düsseldorf, Germany. Association for Computational Linguistics.

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

Hiroya Takamura, Ryo Nagata, and Yoshifumi Kawasaki. 2016. Discriminative analysis of linguistic features for typological study. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 69–76, Portorož, Slovenia. European Language Resources Association (ELRA).

Alex Tamkin, Dan Jurafsky, and Noah Goodman. 2020. Language through a prism: A spectral approach for multiscale language representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 5492–5504. Curran Associates, Inc.

Ming Tan, Yang Yu, Haoyu Wang, Dakuo Wang, Saloni Potdar, Shiyu Chang, and Mo Yu. 2019. Out-of-domain detection for low-resource text classification tasks. 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 3566–3572, Hong Kong, China. Association for Computational Linguistics.

Ryan Teehan, Miruna Clinciu, Oleg Serikov, Eliza Szczechla, Natasha Seelam, Shachar Mirkin, and Aaron Gokaslan. 2022. Emergent structures and training dynamics in large language models. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 146–159, virtual+Dublin. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

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

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages

1357–1366, San Diego, California. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *Computing Research Repository*, arxiv:1908.08962. Version 2.

Francis Tyers and Karina Mishchenkova. 2020. Dependency annotation of noun incorporation in polysynthetic languages. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 195–204, Barcelona, Spain (Online). Association for Computational Linguistics.

Francis Tyers, Mariya Sheyanova, Aleksandra Martynova, Pavel Stepachev, and Konstantin Vinogorodskiy. 2018. Multi-source synthetic treebank creation for improved cross-lingual dependency parsing. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 144–150, Brussels, Belgium. Association for Computational Linguistics.

Dennis Ulmer, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank. 2022. Experimental standards for deep learning in natural language processing research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2673–2692, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

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.

Rob van der Goot. 2023. MaChAmp at SemEval-2023 tasks 2, 3, 4, 5, 7, 8, 9, 10, 11, and 12: On the effectiveness of intermediate training on an uncurated collection of datasets. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 230–245, Toronto, Canada. Association for Computational Linguistics.

Rob van der Goot, Zoey Liu, and Max Müller-Eberstein. 2024. Enough is enough! a case study on the effect of data size for evaluation using universal dependencies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, Torino, Italy. European Language Resources Association.

Rob van der Goot, Max Müller-Eberstein, and Barbara Plank. 2022. Frustratingly easy performance improvements for low-resource setups: A tale on BERT and segment embeddings. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1418–1427, Marseille, France. European Language Resources Association.

Rob van der Goot, Ahmet Üstün, and Barbara Plank. 2021a. On the effectiveness of dataset embeddings in mono-lingual,multi-lingual and zero-shot conditions. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 183–194, Kyiv, Ukraine. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021b. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Marlies van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. 2015. What's in a domain? analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 560–566, Beijing, China. Association for Computational Linguistics.

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. 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 1105–1116, Hong Kong, China. Association for Computational Linguistics.

Sappadla Prateek Veeranna, Jinseok Nam, Eneldo Loza Mencıa, and Johannes Fürnkranz. 2016. Using semantic similarity for multi-label zero-shot classification of text documents. In *Proceeding of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium: Elsevier*, pages 423–428.

Sebastiano Vigna. 2015. A weighted correlation index for rankings with ties. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1166–1176. Association for Computing Machinery.

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

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R.

Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022.

What language model architecture and pretraining objective works best for zero-shot generalization? In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22964–22984. PMLR.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Bonnie Webber. 2009. Genre distinctions for discourse in the Penn TreeBank. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 674–682, Suntec, Singapore. Association for Computational Linguistics.

Lucas Weber, Jaap Jumelet, Elia Bruni, and Dieuwke Hupkes. 2021. Language modelling as a multi-task problem. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2049–2060, Online. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Computing Research Repository*, arxiv:2206.07682. Version 2.

Orion Weller, Kevin Seppi, and Matt Gardner. 2022. When to use multi-task learning vs intermediate fine-tuning for pre-trained encoder transfer learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 272–282, Dublin, Ireland. Association for Computational Linguistics.

Jennifer C. White and Ryan Cotterell. 2021. Examining the inductive bias of neural language models with artificial languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 454–463, Online. Association for Computational Linguistics.

Wikimedia. 2022. Wikimedia downloads.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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.

Tak-sum Wong, Kim Gerdes, Herman Leung, and John Lee. 2017. Quantitative comparative syntax on the Cantonese-Mandarin parallel dependency treebank. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 266–275, Pisa, Italy. Linköping University Electronic Press.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *ArXiv*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *Computing Research Repository*, arXiv: 1609.08144. Version 2.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646, Online. Association for Computational Linguistics.

Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. 2021. Unsupervised out-of-domain detection via pre-trained transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1052–1061, Online. Association for Computational Linguistics.

Yumo Xu and Mirella Lapata. 2019. Weakly supervised domain detection. *Transactions of the Association for Computational Linguistics*, 7:581–596.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Takumi Yoshikoshi, Daisuke Kawahara, and Sadao Kurohashi. 2020. Multilingualization of a natural language inference dataset using machine translation. *Proceedings of the 244th Meeting of Natural Language Processing*, pages 1–8. (Translated from Japanese original).

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR.

Shorouq Zahra. 2020. Parsing low-resource Levantine Arabic: Annotation projection versus small-sized annotated data.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, H̑órunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Siratun

Jannat, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Lorena Martín-Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Steinhór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian

Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal dependencies 2.7/2.8/2.9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.