DECAF: A Dynamically Extensible Corpus Analysis Framework

Max Müller-Eberstein Rob van der Goot Anna Rogers
IT University of Copenhagen, Denmark
`{mamy, robv, arog}@itu.dk`

Abstract

The study of generalization in Language Models (LMs) requires controlled experiments that can precisely measure complex linguistic variations between training and testing datasets. We introduce DECAF, a framework that enables the analysis and filtering of linguistically-annotated datasets down to the character level. Rather than creating new resources for each experiment, DECAF starts from datasets with existing linguistic annotations, and leverages them to analyze, filter, and generate highly controlled and reproducible experimental settings targeting specific research questions. We demonstrate DECAF's functionality by adding 28 morphosyntactic annotation layers to the 115M-word BabyLM corpus and indexing the resulting 1.1B annotations to analyze its internal domain variance, and to create a controlled training data curriculum for a small-scale gender bias study. We release DECAF as an open-source Python library, along with the parsed and indexed version of BabyLM, as resources for future generalization research.

1 Introduction

The core methodological premise of Machine Learning necessitates the evaluation of model capabilities using non-overlapping train-test data splits. For Language Models (LMs), this fundamental assumption is increasingly violated due to issues such as the inaccessibility of pre-training data (Palmer et al., 2023), benchmark contamination (Deng et al., 2024; Dong et al., 2024), and hidden overlaps in train-test splits (Lewis et al., 2021; Kambhatla et al., 2023). Addressing these challenges requires more fine-grained knowledge and control over experimental data (Hupkes et al., 2023). Generalization research thus commonly relies on controlled training data interventions—deliberately removing examples with specific properties from training corpora to evaluate whether models can infer these properties from related structures (Patil et al., 2024). However, the community currently lacks uniform standards and toolkits for conducting such experiments due to the need to balance *complexity*, *specificity*, and *reproducibility*.

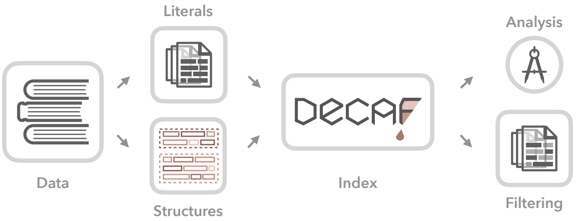


Figure 1: **DECAF** is a framework for large-scale corpus analysis and filtering, which maintains extensibility by constructing separate indices over raw text (`literals`) and annotations (`structures`).

To provide LM generalization researchers with a tool which balances all three desiderata, we introduce DECAF—a Dynamically Extensible Corpus Analysis Framework (illustrated in Fig. 1).

Complexity. Most work studying LM generalization through training data interventions relies on line-by-line filtering of text files, where each line is evaluated based on token-level attributes (Maudslay et al., 2019; Wei et al., 2021; Patil et al., 2024)—e.g., tokens + part-of-speech tags (Misra and Mahowald, 2024). In practice, annotations are concatenated to each token, and filters are defined using regular expressions. This approach has yielded many valuable findings, but it is difficult to extend to more complex filtering criteria. New annotation layers require adding an increasing number of specially formatted tags to each token. Capturing relations beyond the token-level requires formatting, such as bracketing, which makes filtering expressions more complex. Furthermore, queries need to be linearized, limiting the experiments that can be run in languages with freer word orders and more complex morphologies than English.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 351–362
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Specificity. Increasing the complexity of filter queries typically requires specialized tools. For instance, Tregex (Levy and Andrew, 2006) remains the state-of-the-art for filtering constituency-parsed data, and there are many other tools specialized to formats, such as the Universal Dependencies (e.g. Popel et al., 2017; Peng and Zeldes, 2018; Kalpakchi and Boye, 2020). These tools support complex queries, but often have steep learning curves, and as they are not designed to be extensible to annotations beyond their initial purpose, practitioners are limited in the types of research questions they can investigate.

Reproducibility. With larger datasets and more complex filtering criteria, reproducibility becomes increasingly difficult. This problem is especially prevalent for LM pre-training corpora, which stem from less-curated sources. While datasets and processing pipelines have become increasingly standardized and consolidated on centralized hubs (Honnibal and Montani, 2017; Lhoest et al., 2021), filtering often still uses custom scripts which—even if shared—depend on the dataset's original formatting. Working with new annotation layers thus requires changes to both the data formatting and the associated filtering code. Often, it therefore remains necessary to re-process the entire dataset to conduct new experiments.

By designing DECAF with flexibility at its core, we aim to support the next level in scale and complexity for filtered training corpus interventions. Specifically, we contribute:

- DECAF: an open-source framework for filtering corpora with respect to complex criteria across annotation layers (Section 2).
- A demonstration of DECAF, in which we parse and index the 115M-word BabyLM corpus to analyze the syntactic divergence between its sub-domains (Section 3).
- A case study, in which we use DECAF to generate training data interventions for investigating the effects of grammatical gender and data ordering on LM gender bias (Section 4).

We release DECAF as an MIT-licensed Python package, and further publish our parsed BabyLM corpus, with its associated DECAF index and filters, as resources for future work.[1]

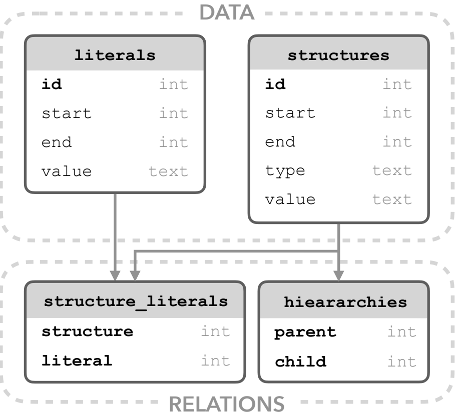


Figure 2: **Database Schema for DECAF**, containing raw text (`literals`), annotations (`structures`), links between the two, as well as hierarchical relationships.

2 DECAF

DECAF acts as a framework over raw data, annotations, and filters. It builds a unified index over existing annotated data, which can be filtered based on complex combinations of annotation layers.

2.1 Intended Use

The primary use case of DECAF is to facilitate experiments using filtered training corpus interventions (Patil et al., 2024). In addition to filtering, it can also be used to analyze existing corpora with respect to annotated properties, as well as to compare different corpora with each other. Even with limited annotations, such as for classification benchmarks, the framework can help identify spurious signals by, e.g., identifying tokens which co-occur frequently with a particular label. In cases without any pre-existing annotations, DECAF can help quantify character-level overlaps across training and evaluation data. As such, we believe that DECAF's corpus analysis features can also help reduce errors during the creation of new corpora, by continually identifying common error patterns.

2.2 Design Principles

To maintain extensibility across different types of annotations, DECAF breaks them down into their elemental components and constructs a unified database index. Fig. 2 illustrates the underlying schema, which encompasses the following.

`literals` as the atomic unit for raw text, including its value and position in the original corpus. They can correspond to different granularities, such as characters (e.g., for morphological analy-

[1] `https://mxij.me/x/decaf`

ses), tokens (e.g., for most word-level annotations), sentences (e.g., sentence-level classification), etc. While they are necessary for filtering by surface forms (e.g., `upos="DET" & literal="an"`), they can be omitted to save storage space or to prevent indexing of copyrighted materials.

`structures` correspond to linguistic units (e.g., boundaries of morphemes, tokens, documents), and annotations thereof. They are specified by their position, type (e.g., `"token"`, `"upos"`), and value. For linguistic units the value is traced back to the corresponding literal, while for annotations it corresponds to their labels (e.g., `"VERB"`, `"positive"`).

`structure_literals` links structures back to their literals and is used for analyzing annotations with respect to their surface form. While some structures directly correspond to their start and end range in the original corpus, this junction table allows for the extraction of non-linear structures, such as for graphs with intersecting edges.

`hierarchies` stores hierarchical relationships between structures. At a fundamental level this includes the relationships between, e.g., token-annotations → tokens → sentences → documents. This information is used to resolve filtering queries, which search for lower-level annotations contained in a specific higher-level structure. Additionally, this table links graphical structures, such as dependency trees, entity graphs, or cross-document links.

By importing common NLP annotations into this unified schema, we create one index across many different types of linguistic information, while preserving rich, cross-structural relationships not captured by linearized filtering systems. Despite the simplicity of this schema, extracting relevant information requires the construction of complex database queries. To make such queries accessible to users with different experience levels, DECAF provides a simplified Python API which translates and optimizes filtering criteria into the database language, and automatically manages other hyper-parameters for efficient processing.

2.3 Implementation

Backend. Among the database backends which could support the DECAF schema, we opt for the open-source SQLite engine.[2] Compared to more feature-rich backends, it offers a simple, server-less setup, which is essential for the ease-of-use by individual researchers. Furthermore, the resulting indices are self-contained in the respective SQLite files, making them easy to share. To ensure high read and write speeds, and scalability to larger annotated corpora, DECAF further implements sharding of larger datasets into sub-databases, while preserving hierarchical dependencies, such as document boundaries. Sharding happens transparently to the user, who can query the entire corpus as one.

Scalability. The core technologies of DECAF are highly scalable: the database backend plus sharding can be easily parallelized when additional compute is available. Even in terms of single-threaded performance, our experiments in Sections 3 and 4 exhibit linear scaling with respect to the number of tokens versus processing time.

Packaging. The Python API for database management and querying is implemented with a focus on limiting external dependencies to ensure future reproducibility. As such, filtering of existing indices requires only the Python standard libraries, while external libraries are primarily used to parse annotated data for index creation and for running more complex analyses on the resulting statistics.

Extensibility. DECAF is designed to be easily extensible to new annotation formats, as all queries are processed within the unified data schema. By uncoupling raw data and annotations, annotation layers can be continually added to existing indices without having to, e.g., modify text files and rewriting regular expression filters. Adding support for new annotation formats thus only requires contributors to supply an import script. While the default API aims to provide the most common querying functionalities, it can be extended to support more annotation-specific queries (e.g., dependency tree traversal). As filters further query the index, instead of the raw data, they can also be easily shared and applied to new datasets. We believe this dataset-agnostic framework allows for a more scalable, community-driven approach to conducting corpus analyses, and filtered training interventions.

2.4 Interface

Import. Data indexing is handled by dedicated scripts, which translate each annotation format into the unified schema. Out-of-the-box, DECAF provides an interface for importing CoNLL-U data—a popular format for linguistic annotation, used in

[2] https://sqlite.org

the Universal Dependencies project (Nivre et al., 2020). An index is constructed by running:

```
python scripts/import/ud.py
  --input /path/to/data.conllu
  --output /path/to/index
```

Filtering. Data retrieval and filtering from a DECAF index is specified through a Python API, in which the user defines a `Filter` containing one or more `Criteria`, each with one or more `Conditions`. For example, the syntactic generalization experiments of Misra and Mahowald (2024) rely on identifying all Article+Adjective+Numeral+Noun constructions (e.g., "a beautiful five days")—originally, using a 326-character regular expression. In DECAF, we would specify this intervention as the following filter:

```
Filter([
    Criterion([
        Condition(
            stype='upos',
            values=['DET'],
            literals=['a', 'an'])]),
    Criterion([
        Condition(
            stype='upos',
            values=['ADJ'])]),
    Criterion([
        Condition(
            stype='upos',
            values=['NUM'])]),
    Criterion([
        Condition(
            stype='upos',
            values=['NOUN']),
        Condition(
            stype='Number',
            values=['Plur'])],
        operation='AND')],
    sequential=True,
    hierarchy=['sentence', 'token']
)
```

The filter matches sentences within which all criteria occur in sequence at least once. Note that besides solely matching PoS-sequences, as in the original work, we can more specifically provide, e.g., the desired surface form ("a", "an"), and nouns in plural form. Finally, we supply a hierarchical constraint, which specifies that the conditions must be fulfilled for tokens within individual sentences (i.e., cannot cross sentence boundaries).

Export. With the filter in place, the relevant data can be extracted or masked from the index by applying it in a script following the example in:

```
python scripts/export/filtered.py
  --input /path/to/index
  --output /path/to/output.txt
```

DECAF can operate both at the level of parent structures (e.g., all sentences containing the matched structures), as well as at the sub-structure level to, e.g., remove all relative clauses from a corpus, while keeping the main clause intact.

3 Case Study: Analyzing BabyLM

To demonstrate the analysis functionality of DECAF, and to provide the community with a reusable resource, we create a morphosyntactically parsed and indexed version of the 115M-word BabyLM corpus (Warstadt et al., 2023). We then analyze the similarity of its sub-corpora with respect to the distributional divergence of their linguistic properties.

3.1 Parsing

Our annotation layers for BabyLM include the default Universal Dependencies (Nivre et al., 2020; UD) annotations for tokenization, universal parts-of-speech (UPoS), dependencies, as well as the extended XPoS, and 23 morphological layers, plus lemmatization. We train a multi-task model to perform all tasks simultaneously using the MaChAmp toolkit (van der Goot et al., 2021) v0.4.2, using default hyperparameters. As training data, we use the UD GUM-corpus (Zeldes, 2017), as it covers our target annotation set, is manually annotated, and contains a wide variety of domains, which we expect to lead to more robust transfer performance. To obtain accurate annotations, we compared the performance of four different LMs, and selected `DeBERTa-v3-large` (He et al., 2021) as our final model. More details on the parsing procedure and annotation layers can be found in Appendix A.

3.2 Analysis

After parsing the BabyLM corpus, we next index all sub-corpora using DECAF. Table 1 shows that our pipeline identified 115M words with 1.1B annotations, linked via 1.3B hierarchical relations. Indexing this corpus on an M3 MacBook Pro takes ∼1.5 hours. As indexing time scales linearly with corpus size, even on a local machine, this indicates a reasonable potential for scaling to larger corpora.

With the indices, we next demonstrate running a high-dimensional Exploratory Data Analysis (EDA) using DECAF. Specifically, we query the frequency distribution of each annotation layer and compute the pairwise Jensen-Shannon divergence (JSD; Wong and You, 1985) across all sub-corpora, taking the average JSD across annotation types to obtain the final divergence (details in Appendix B).

Subset	Sentences	Words	Literals	Structures	Hierarchies	Time
BNC	819,740	8,794,948	16,532,030	77,076,051	93,026,467	387s
CHILDES	5,809,876	30,811,091	54,254,290	277,728,676	327,731,106	1,901s
Gutenberg	1,640,286	31,980,830	58,341,144	274,224,492	334,905,580	1,372s
Subtitles	3,508,947	24,933,681	44,863,286	219,920,133	262,769,601	1,061s
Switchboard	164,993	1,785,749	3,125,325	15,019,774	18,261,286	73s
Wiki	1,116,999	17,023,435	31,338,669	143,048,435	174,861,307	697s
Total	13,060,841	115,329,734	208,454,744	1,007,017,561	1,211,555,347	5,491s

Table 1: **BabyLM Index Statistics per Subset**, showing the number of sentences, words, database entries for literals, structures, and hierarchies, as well as the runtime for importing each subset into a DECAF index.

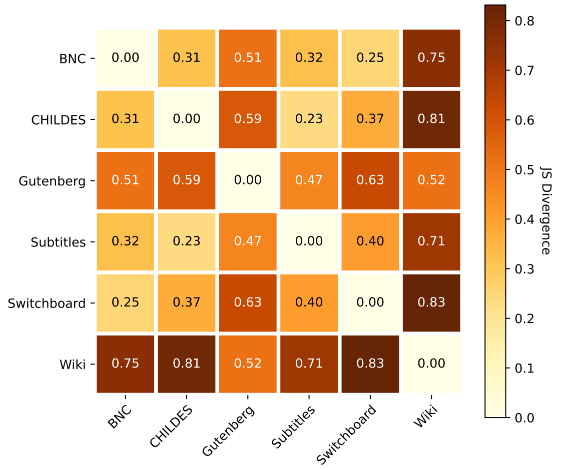


Figure 3: **Morphosyntactic Divergence of BabyLM Sub-corpora** as measured by the Jensen-Shannon divergence with respect to their annotation distributions.

Fig. 3 shows a clear split between written and spoken language, where Wiki and Gutenberg diverge up to 0.83 JSD from the other sub-corpora. Using DECAF to extract the overlaps across specific annotation layers, we find that these differences are driven by style differences (e.g., more vernacular, “kind of“, “like“ in spoken data), and domain-specific biases, such as Wiki almost exclusively using negative polarity indicators (e.g., “no”, “not”). Our analysis also identifies transcription differences, such as Wiki writing numbers as digits, while speech datasets write them out as words. All spoken datasets further share comparable distributions over the grammatical person used, while Wiki almost never uses the first person, and Gutenberg uses the third person 71% of the time. Finally, all corpora share similar skews in their gender pronoun distributions with an average of 17% female, 33% male, and 50% neutral pronouns.

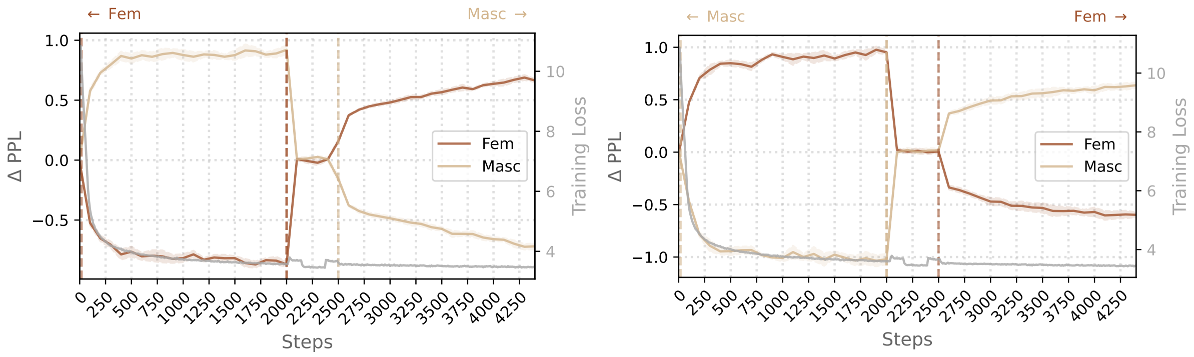
This EDA shows how DECAF can help identify domain characteristics, annotation mismatches, and biases that may be relevant during dataset creation, as well as for generating targeted training interventions.

4 Case Study: Training Interventions for Gender Bias Mitigation

To demonstrate DECAF’s ability to aid targeted training interventions, we next run a small-scale case study investigating: *What are the effects of training data order on occupational gender bias?* Specifically, the contrast between catastrophic forgetting (Kotha et al., 2024), which posits that later data are more likely to be retained, versus observations that data presented earlier are memorized better (Leybzon and Kervadec, 2024). Measuring how downstream model bias is affected by *when* minority group data are observed may be helpful for informing gender bias mitigation strategies.

Data Using DECAF, we construct a training data intervention as follows: First, we define 16 filters, which extract all BabyLM sentences containing pronouns of a specific gender (details in Appendix C). These sentences are then sorted by their specificity with respect to the research question, i.e., sentences containing the target gender + a target occupation (`Occ`) come first, while mixed-gender sentences, and sentences containing the non-target gender come later. Next, we balance the total number of pronouns in each specificity level to obtain exactly the same amount of sentences containing one gender versus the other. Finally, we interleave the gendered sentences with the remaining non-gendered BabyLM data at regular intervals, obtaining the training data schedule: `Fem+Occ` → `Fem` → `Fem+Masc` → `Masc` → `Masc+Occ` (and reverse). The final training data includes 12.5M total sentences, including 1.1M gendered sentences, interleaved every 11 steps.[3]

[3] Note that about 500k sentences with exclusively masculine pronouns are removed during data balancing.



(a) Fem → Masc

(b) Masc → Fem

Figure 4: **Training Dynamics on WinoBias**, training five seeds of Pythia-14M from scratch on data with a balanced number of FEM/MASC pronouns, for which the distribution shifts from one to the other across training (indicated by dashed vertical lines). ΔPPL shows the perplexity increase/decrease for when occupation-pronoun pairs are anti-stereotypical; Training Loss is measured by the cross-entropy loss on next-token prediction.

Evaluation The success of the intervention is evaluated using WinoBias (Zhao et al., 2018), a benchmark measuring a model's ability to link a binary gendered pronoun to one of 40 occupation in 1,584 sentences (e.g., "The developer argued with the designer because she did not like the design."). We report the change in perplexity when the encountered pronoun is anti-stereotypical (ΔPPL), i.e., how 'surprised' the model is by a gender and occupation co-occurring. For LM pre-training, we report the cross-entropy loss. In total, we evaluate the training dynamics of 1,380 model checkpoints.

Models We train five seeds of Pythia-14M (Biderman et al., 2023; van der Wal et al., 2025) from scratch on our modified training data. While small in scale, their pre-trained checkpoints already exhibited clear biases on WinoBias (Fig. 5), making them well suited for demonstrating the effect of this training intervention. We use the hyperparameters reported by Biderman et al. (2023) for one epoch, and track the model bias during training.

4.1 Results

Fig. 4 shows the training dynamics of the Fem → Masc, and Masc → Fem interventions. The general training loss follows a stable trajectory, which starts converging after around 1.5k steps. Meanwhile, ΔPPL flips throughout training in accordance with the data ordering. During early training when only one type of gendered pronoun has been observed (i.e., before 2.2k steps), the models unsurprisingly exhibit less perplexity when presented with pronouns of the observed type. At the half-way point, the models observe the first sentences containing pronouns of more than one gender. In this range, there is a brief period in which ΔPPL tends towards zero, before it flips in favor of the new pronoun, which remains as a final bias until the end of training. At both flips, we notice a small spike in the overall training loss, indicating that the model is adjusting to the data change. This pattern is mirrored for either data setup, with the final bias tending towards overconfidence for the most recently observed gender. For bias mitigation strategies, our results indicate that balanced training data alone is insufficient to reduce gender bias, and that recency bias must be taken into account. While this experiment should not be taken as a full study of bias mitigation, it demonstrates DECAF's ability to construct targeted training interventions for the study of LM training dynamics.

5 Conclusion

We introduced DECAF, a flexible framework for analyzing and filtering annotated datasets in order to facilitate targeted LM training corpus interventions. Using DECAF, we analyzed a parsed version of the 115M-word BabyLM corpus, containing 1.1B annotations in complex hierarchical relationships. Using the resulting index, we measured the distributional divergence of 24 morphosyntactic annotation layers across the sub-corpora of BabyLM. Finally, we conducted a case study on how the order of gendered pronouns in a balanced corpus affects LM performance on the WinoBias benchmark. The high level of control DECAF provides over the generated training data allowed us to observe clear shifts in bias throughout training despite an otherwise balanced corpus.

Limitations

Dependence on Existing Annotations. DECAF does not perform any annotation on-the-fly, it relies on annotations that are already available or performed by an external annotation tool. We believe this separation of text and annotations is crucial for future extensibility. In total absence of annotations, DECAF can still be used to compute character-level overlaps across indices, e.g., to compare training data with target benchmarks. Additionally, we share the parsing scripts, as well as the data and models used in our case studies.

Annotation Formats. Currently, DECAF supports indexing datasets in CoNLL-U format, enabling the import of popular linguistically annotated datasets, such as the Universal Dependencies, but limiting the scope of available annotations. As the underlying data schema is highly flexible, we anticipate that new annotation formats can be easily integrated by providing dedicated import scripts.

Filter Types. DECAF includes a Python API for constructing complex filters for the underlying data indices. For certain types of annotations, this interface may however not be able to handle all queries: e.g., traversing nested hierarchical structures in constituency parses. As the required information is nonetheless available in the underlying database schema, implementing these filters is a matter of augmenting the relevant SQL queries. Towards incorporating such specific features in the future, we build the filtering API with extensibility in mind by providing relevant pre-constructed SQL views, and allowing for the direct querying of the underlying databases, should users be proficient in SQL.

Case Study: BabyLM. To the best of our knowledge, we provide the most granular analysis of the morphosyntactic overlaps across the sub-corpora of BabyLM to date. While our analysis based on Jensen-Shannon divergence allows us to identify the root differences across domains (e.g., between written and spoken data), it is by no means comprehensive. We hope that future work can build on the annotations and indices, which we release, and develop new modes of analysis to provide an incrementally clearer picture of how these sub-corpora differ. Both the older methodologies from corpus linguistics (Kilgarriff, 2001; McEnery and Hardie, 2013) and the newer techniques developed for the analysis of NLP datasets, such as dataset cartography, or the annotation artifact identification (Gururangan et al., 2018; Swayamdipta et al., 2020), may provide inspiration for future linguistic criteria to be indexed and analyzed.

Case Study: WinoBias. The experiments in Section 4 are run at a smaller scale compared to LMs which are used in production, and are intended only as a demonstration of DECAF framework. However, overall there is currently much interest in research on smaller models in order to predict performance on larger models (Ivgi et al., 2022), and Pythia-14M's training dynamics have been shown to be indicative of its larger variants (van der Wal et al., 2025). The BabyLM corpus itself is frequently used to conduct similar training interventions, wherein LMs are trained from scratch for studying their generalization capabilities (e.g., Misra and Mahowald, 2024). Finally, while WinoBias covers binary gendered pronouns only, the filters applied in our experiments can easily be extended with additional genders, cases, etc., (including in other languages), given the relevant annotations. The fact that the indexing and filtering of 115M words can already be conducted on a local machine further gives us confidence in DECAF's ability to scale to larger corpora necessary for training modern LMs.

Broader Impact

DECAF supports basic research on generalization and robustness of Machine Learning solutions for Natural Language Processing. It aims to broaden the scope of experiments that are possible with training data interventions and highly-controlled train-test splits—making such research easier and more accessible. Towards this goal, we provide a unified indexing schema which can support a wide variety of annotations. To not compromise reproducibility through added complexity, we further separate the raw data, annotations, and filtering. This way, indices on pre-existing annotations can be shared and extended, while filters operate in a unified space, meaning that they are transferable across different datasets.

6 Acknowledgements

We thank the anonymous reviewers for their insightful feedback, and thank the IT University of Copenhagen's High-Performance Computing Cluster for supporting the case studies in this work. This work was funded by the Independent Research Fund Denmark (grant 3160-00022B).

References

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

BNC Consortium. 2007. British national corpus, XML edition. Literary and Linguistic Data Service.

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.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719, Mexico City, Mexico. Association for Computational Linguistics.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050, Bangkok, Thailand. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1).

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, et al. 2023. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5(10):1161–1174.

Maor Ivgi, Yair Carmon, and Jonathan Berant. 2022. Scaling laws under the microscope: Predicting transformer performance from small scale experiments. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7354–7371, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dmytro Kalpakchi and Johan Boye. 2020. UDon2: a library for manipulating Universal Dependencies trees. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 120–125, Barcelona, Spain (Online). Association for Computational Linguistics.

Gauri Kambhatla, Thuy Nguyen, and Eunsol Choi. 2023. Quantifying train-evaluation overlap with nearest neighbors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2905–2920, Toronto, Canada. Association for Computational Linguistics.

Adam Kilgarriff. 2001. Comparing Corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2024. Understanding catastrophic forgetting in language models via implicit inference. In *The Twelfth International Conference on Learning Representations*.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings* *of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Danny D. Leybzon and Corentin Kervadec. 2024. Learning, forgetting, remembering: Insights from tracking LLM memorization during training. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 43–57, Miami, Florida, US. Association for Computational Linguistics.

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.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. 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 5267–5275, Hong Kong, China. Association for Computational Linguistics.

Tony McEnery and Andrew Hardie. 2013. The History of Corpus Linguistics. In *The Oxford Handbook of the History of Linguistics*. Oxford University Press.

Kanishka Misra and Kyle Mahowald. 2024. Language models learn rare phenomena from less rare phenomena: The case of the missing AANNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 913–929, Miami, Florida, USA. Association for Computational Linguistics.

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.

Alexis Palmer, Noah A. Smith, and Arthur Spirling. 2023. Using proprietary language models in academic research requires explicit justification. *Nature Computational Science*, 4(1):2–3.

Abhinav Patil, Jaap Jumelet, Yu Ying Chiu, Andy Lapastora, Peter Shen, Lexie Wang, Clevis Willrich, and Shane Steinert-Threlkeld. 2024. Filtered corpus training (FiCT) shows that language models can generalize from indirect evidence. *Transactions of the Association for Computational Linguistics*, 12:1597–1615.

Siyao Peng and Amir Zeldes. 2018. All roads lead to UD: Converting Stanford and Penn parses to English Universal Dependencies with multilayer annotations. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 167–177, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka. 2022. mLUKE: The power of entity representations in multilingual pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7316–7330, Dublin, Ireland. Association for Computational Linguistics.

Beatrice Santorini. 1990. Part-of-speech tagging guidelines for the penn treebank project.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. 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.

Oskar van der Wal, Pietro Lesci, Max Müller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. 2025. Polypythias: Stability and outliers across fifty language model pre-training runs. In *The Thirteenth International Conference on Learning Representations*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wikimedia Foundation. 2022. Wikipedia (simple english). Dump from 1st December, 2022.

Andrew K. C. Wong and Manlai You. 1985. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5):599–609.

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.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Appendix

A BabyLM Parsing

Sub-corpora. The BabyLM corpus (Warstadt et al., 2023) is a collection of six sub-corpora, which aim to capture different facets of child-directed language. The size of the corpus is motivated by the number of words a child is typically exposed to before the age of 12. In our studies, we use the corresponding 100M-word version of the corpus, counting 115M syntactic words following our own tokenization pipeline. The six sub-corpora are divided as follows:

- BNC (BNC Consortium, 2007): 8.8M words of transcribed, spoken dialogue from the British National Corpus.
- CHILDES (MacWhinney, 2000): 30.8M words from the CHILDES project, which includes child-directed/produced speech, and situational descriptions (in square brackets). Each utterance starts with a speaker identifier (e.g., `CHI`, `MOT`), which we extract into a separate `speaker` metadata field.
- GUTENBERG (Gerlach and Font-Clos, 2020): 32.0M words from books in Project Gutenberg, from authors born after 1850.
- SUBTITLES (Lison and Tiedemann, 2016): 24.9M words from the OpenSubtitles project, which includes movie and TV subtitles covering spoken dialogue, as well as situational descriptions (in round brackets).
- SWITCHBOARD (Stolcke et al., 2000): 1.8M words from the Switchboard Dialogue Acts corpus of transcribed phone conversations.
- WIKI (Wikimedia Foundation, 2022): 17M words from the Simple English Wikipedia.

Note that the number of words in our parsed corpus is higher than reported in the original corpus, due to the fact that our tokenization identifies *syntactic words*, i.e., functional units in the Universal Dependencies schema (e.g., `It's` → `It 's`).

Annotation Layers. For the initial sentence segmentation of BabyLM, we use the NLTK segmenter (Bird et al., 2009). For parser training and inference, we used the default hyperparameters of MaChAmp, ignoring multi-word tokens according to the ud-conversion-tools[4]. We train a separate decoder head for each of the following tasks:

- word: word segmentation modeled as a binary subword level labeling task.
- UPoS: 17 PoS tags following the UD guidelines, predicted by a single feedforward layer.
- XPoS: language/corpus-specific PoS tags, which, in the case of GUM, follow the Penn Treebank guidelines (Santorini, 1990) and cover 45 finer-grained labels.
- lemma: the canonical or base form of the word. In MaChAmp this task is converted to a sequence labeling task, where a label describes character edits of the transformation of a word to its lemma.
- morphology: the labeling of 21 features (following GUM), each describing a morphological categorization. If a feature is present, it includes a label for the specific category (e.g., Number=Sing). For the purpose of DECAF, we separate each feature into a separate annotation layer.
- dependencies: syntactic dependency relations that hold between words. MaChAmp implements this task through a Deep Biaffine Parser (Dozat and Manning, 2017). Each word is labeled with a reference to its parent + the syntactic relation between them. There are 36 different relations in UD.

For selecting the base language model to parse BabyLM with, we first evaluated 4 LMs on the development data of the GUM corpus[5]: DeBERTa-v3-large (He et al., 2021), luke-large (Yamada et al., 2020), mluke-large (Ri et al., 2022), and xlm-roberta-large (Conneau et al., 2020). On the development data of GUM, the average performance of the best model over all 5 tasks was 98.0 F1. This was within 0.2% compared to the worst LM (97.7 F1). Hence, we opted for a qualitative comparison; an annotator with previous experience in UD annotation inspected the first 25 differences in predictions on our target data. Based on these observations, we selected DeBERTa-v3-large model as our final model.

B BabyLM Analysis

The annotation divergence analysis in Section 3 is based on the frequency distributions of all 'non-sparse' annotation layers (i.e., no tokens, or lemmas). This includes the morphological annotations, Abbr, Case, Definite, Degree, ExtPos, Foreign, Gender, Mood, NumForm, NumType, Number, Person, Polarity, Poss, PronType, Reflex, Style, Tense, Typo, VerbForm, Voice, as well as the syntactic annotations, deprel, upos, xpos. As some of these annotations are binary (e.g., abbreviations), we add an Other category to each of these, which covers all non-marked occurrences.

For measuring the distributional similarity, we chose the Jensen-Shannon divergence (JSD; Wong and You, 1985), which we compute for each annotation type $a \in \mathcal{A}$ across each sub-corpus pair i, j, before taking an overall average:

$$\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \mathrm{D}_{\mathrm{JS}}(p_{i,a} || p_{j,a}) \qquad (1)$$

C WinoBias Experiments

Filters. We define filters of increasing specificity to the WinoBias benchmark, to identify all gendered pronoun occurrences in the BabyLM corpus. In simplfied form, these include:

- any target-gender pronoun:
  - {upos=PRON & Gen=Fem}
- target-pronoun as subject:
  - {upos=PRON & Gen=Fem & dep=nsubj}
  - {upos=VERB|AUX}
- target-pronoun as subject of subordinate clause:
  - {upos=SCONJ & dep=mark}
  - {upos=PRON & Gen=Fem & dep=nsubj}
  - {upos=VERB|AUX}
- target-pronoun as oblique:
  - {upos=ADP}
  - {upos=PRON & Gen=Fem & dep=obl}

Additionally, we add filters, in which any of the above co-occur in a sentence with any of WinoBias' 40 occupational terms (Zhao et al., 2018). Together with filters targeting the opposite gender, we construct at a total of 16 DECAF filters.

[4] https://github.com/bplank/ud-conversion-tools

[5] https://robvanderg.github.io/evaluation/tune-lms/ informed our initial selection.

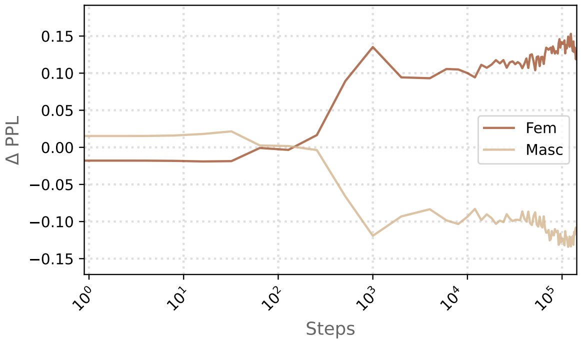


Figure 5: **Training Dynamics on WinoBias**, across the original pre-training of Pythia-14M (Biderman et al., 2023; van der Wal et al., 2025). ΔPPL shows the perplexity increase/decrease for when occupation-pronoun pairs are anti-stereotypical.

Pre-trained Models. Evaluating five seeds of the pre-trained Pythia-14M checkpoints (Biderman et al., 2023; van der Wal et al., 2025) throughout their original training on the Pile corpus (Gao et al., 2020), Fig. 5 shows perplexity that is biased against female pronouns. This divide manifests surprisingly quickly, after around 1k training steps, or 0.7% of full training, and remains until the end.

Custom Model Training. For training our own Pythia-14M models on the data interventions generated by DECAF, we train using the same hyperparameters as in (Biderman et al., 2023), on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. Training one model takes approximately one hour.