SocArXiv

SocArXiv Preprint : 2026-04-12 10.31235/osf.io/6q8yg_v2

LLM Tool: A Hybrid Pipeline for Automated High-Throughput Text Annotation Using Local Language Models and BERT Classifiers

Antoine Lemor
Université de Sherbrooke

Shannon Dinan
Université Laval

Jérémy Gilbert
Western University

Abstract

Large language models now routinely annotate text in computational social science, but they do not hold up at corpus sizes of several million sentences. Proprietary LLMs are financially prohibitive, local open-weight LLMs take days or weeks of computation, and both remain opaque and hard to reproduce. Using LLMs to train dedicated classifiers solves these problems, yet the approach itself remains sparsely tested in the social sciences and largely inaccessible to researchers without engineering support. We present LLM Tool, an open-source Python package that runs the full hybrid workflow from a command line. On a bilingual corpus of 38,451 Canadian parliamentary debates and news media texts coded across four dimensions, classifiers trained on the best LLM labels reach a mean Micro F1 of 68.9%. Open-weight models such as GPT-OSS match one of the best proprietary models available, GPT-5, and deliver a 109–395× inference speedup over direct LLM annotation on standard workstations.

Keywords: large language models, text annotation, computational social science, transformer classifiers, methodological transparency.

Funding: This research was made possible by a *Programme d'appui à l'innovation pédagogique* grant from the Faculté des Sciences Sociales of Université Laval.

Introduction

Social scientists face a growing tension between the volume of available textual data and their capacity to analyze it rigorously. Parliamentary records, news archives, and social media streams generate millions of documents, yet the annotation methods that ensure valid

measurement remain slow, expensive, and hard to extend to new corpora (Grimmer, Roberts, and Stewart 2022; Krippendorff 2018). Large language models (LLMs) have emerged as a promising response, and many projects now rely on zero- or few-shot applications for text classification and annotation (Ornstein, Blasingame, and Truscott 2025; Chae and Davidson 2025). But the rapid adoption of these models has outpaced the development of methodological safeguards. Proprietary APIs limit reproducibility, commercial pricing restricts access, and the opacity of models undermines the transparency on which cumulative social science depends (Ollion, Shen, Macanovic, and Chatelain 2024; Weber and Reichardt 2023).

Paradoxically, each available approach solves one problem at the price of another. Proprietary LLMs are efficient but expensive, local open-weight LLMs are affordable but slow, and both leave the researcher choosing between money and time before they even reach the question of annotation quality. Concretely, annotating tens of thousands of sentences on local hardware requires days to weeks of continuous computation, while commercial API costs grow linearly with every new dataset. Fine-tuning smaller models improves efficiency but still requires labeled data that must come from somewhere (Chae and Davidson 2025). Human annotation remains the methodological standard, yet producing consistent labels across complex coding schemes demands extensive effort and yields only modest agreement even among trained experts (Artstein and Poesio 2008; Do, Ollion, and Shen 2024). What has been missing is infrastructure that combines the ability of LLMs to annotate directly from a codebook with the efficiency of task-specific classifiers in a workflow that any researcher can execute locally, reproduce entirely, and audit at every step.

The logic behind such pipeline is relatively straightforward: a LLM produce annotations that are then used to train a downstream classifier that reproduce the labeling logic produced by the LLM through what the literature calls "distillation" (Hinton, Vinyals, and Dean 2015; Di Palo, Singhi, and Fadlallah 2024; Tan, Li, Wang, Beigi, Jiang, Bhattacharjee, Karami, Li, Cheng, and Liu 2024). However, for social science research, the harder question is whether such a design can actually democratize LLM-based annotation by making large-scale annotation more accessible while maintaining the quality necessary for conducting rigorous research. This paper then asks: *can an encoder–decoder pipeline, executable locally on commodity hardware, produce annotations of a quality close enough to human coders to be trusted in downstream social-science analyses, and does the choice of LLMs as annotator matter once its outputs are distilled into a task-specific classifier?*

This article introduces LLM Tool, an open-source Python package designed to answer precisely this question. LLM Tool implements a full hybrid workflow through a command-line interface that any researcher can run on a local machine without writing any code. The logic behind it is as follows: decoder-based LLMs generate structured annotations, encoder-based transformer classifiers are then fine-tuned to reproduce the labeling logic with far greater efficiency, and every stage is automatically documented to ensure full traceability. We validate such pipeline

and answer our question by deploying LLM Tool on a bilingual Canadian corpus of 38,451 parliamentary debates and news media texts, annotated across four dimensions by five LLMs (27B–120B parameters, open-weight and proprietary) and evaluated against 1,549 sentences labeled by four human experts. The 20 resulting XLM-RoBERTa classifiers split into two clearly separated tiers: GPT-5 and GPT-OSS significantly outperform Gemma, Mixtral, and Nemotron ($p < 0.0001$), while the hybrid architecture delivers a 109–395× inference speedup over direct LLM annotation. The pipeline therefore brings high-throughput, near-human annotation within reach of any research team, but the choice of annotator remains a first-order methodological decision.

1. Previous Work

1.1. LLMs in Social Science: Promise, Limits, and Alternatives

Large language models have inspired considerable optimism among computational social scientists for their apparent ability to offer a solution to one of the field's most persistent challenges: annotating text corpora large enough to sustain quantitative analysis (Grimmer et al. 2022; Ziems, Held, Shaikh, Chen, Zhang, and Yang 2024; Tan et al. 2024). The production of labeled data has typically required extensive human effort, clear coding guidelines, and costly quality control (Krippendorff 2018; Artstein and Poesio 2008). LLMs, in contrast, appear to provide an efficient solution because they draw on vast pre-trained knowledge to perform complex classification and extraction tasks with minimal supervision. The prospect of moving quickly from a codebook to hundreds of thousands of coded sentences has made them attractive for research programs that would otherwise demand prohibitive amounts of time and human work.

Evidence, nevertheless, tempers this promise. While LLMs can achieve high accuracy on straightforward and unambiguous labeling tasks such as political affiliation detection, sometimes even outperforming supervised models (Törnberg 2025) and crowd-sourced annotators (Gilardi, Alizadeh, and Kubli 2023), they perform far less reliably on the complex and multidimensional coding schemes common in social science research (Halterman and Keith 2026; Alizadeh, Kubli, Samei, Dehghani, Zahedivafa, Bermeo, Korobeynikova, and Gilardi 2024). Open-source models, in particular, show a limited capacity to follow detailed codebooks, even when these provide explicit definitions and examples (Halterman and Keith 2026). A systematic comparison across training regimes (zero-shot, few-shot, fine-tuning, and instruction-tuning) confirms that the largest models offer the best zero-shot performance but that fine-tuning smaller models remains competitive due to lower cost and comparable accuracy (Chae and Davidson 2025).

The performance gap between proprietary models like GPT-4 and open alternatives remains substantial on political text classification benchmarks (Alizadeh et al. 2024), and even the

most advanced systems struggle when annotation requires sustained attention to nuanced and context-dependent distinctions. Moreover, multilingual settings tend to add further complexity. GPT performs reliably across 12 languages for psychological constructs (Rathje, Mirea, Sucholutsky, Marjieh, Robertson, and Van Bavel 2024), yet off-the-shelf multilingual models show uneven performance across languages and domains for more complex tasks (Bhat and Varma 2023; Mets, Karjus, Ibrus, and Schich 2024). Ultimately, it appears that task complexity, rather than model architecture or raw capacity, determines whether LLMs fulfill their promise or expose their current limits.

A second set of concerns involves the conditions under which many LLMs are available to researchers. Commercial API-based models operate as black boxes that limit validity checks and reproducibility (Ollion et al. 2024), and because these systems are continuously updated, deprecated, or replaced, results obtained at one point in time may not be replicable later (Weber and Reichardt 2023). Per-token pricing grows with data volume in ways that create barriers for unfunded teams and new researchers. For corpora of millions of sentences, end-to-end annotation with decoder LLMs becomes practically infeasible. Local models would require months or years of computing for inference alone, and commercial endpoints incur costs that exceed typical research budgets.

These technical constraints also produce broader concerns. Data privacy restricts the use of external services when corpora contain sensitive material, and political and demographic biases in training data raise important ethical issues (McGee 2023; Rozado 2023; Dong, Wang, Yu, and Caverlee 2023). In a field already confronting a reproducibility crisis, where independent teams analyzing identical data reach widely different conclusions depending on analytical choices (Breznau, Rinke, Wuttke, et al. 2022), the opacity of LLM-based methodologies is particularly concerning and underscores the need for transparent, auditable, and locally executable alternatives.

An alternative to zero-shot LLMs is the use of fine-tuned models, which show clear improvements in complex text annotation tasks (Alizadeh et al. 2024; Halterman and Keith 2026). For political text classification, fine-tuned models outperform few-shot prompting strategies when provided with even modest amounts of training data (Alizadeh et al. 2024; Wang, Qu, and Ye 2024b), though the optimal choice between fine-tuning and prompting depends on task complexity and dataset size (Chae and Davidson 2025). However, fine-tuning still requires labeled data. In that case, low-resource scenarios may match traditional approaches only when training sets reach several hundred or thousand examples (Wang et al. 2024b). Fine-tuning therefore shifts rather than solves the problem.

A further alternative is to train task-specific transformer models, specifically encoder-based architectures such as BERT (Devlin, Chang, Lee, and Toutanova 2019) and its multilingual derivatives (notably XLM-RoBERTa, a multilingual variant pre-trained on 100 languages; Conneau, Khandelwal, Goyal, Chaudhary, Wenzek, Guzmán, Grave, Ott, Zettlemoyer, and

Stoyanov 2020), through transfer learning. This approach leverages knowledge acquired during pre-training on very large text collections to adapt general language understanding to new tasks with relatively small amounts of labeled data (Laurer, van Atteveldt, Casas, and Welbers 2024; Wankmüller 2024). Pre-trained BERT models can match the performance of classical supervised models while requiring roughly a tenfold reduction in training data (Laurer et al. 2024). Domain-specific applications confirm the practical value of this approach (Dreier, Serrano, Gade, and Smith 2025; Matsuo and Fukumoto 2025; Bonikowski, Luo, and Stuhler 2022).

Yet transfer learning's efficiency depends on the quality of the supervision it receives during fine-tuning. Human annotators show variable consistency even on well-defined tasks, and general reviews of inter-annotator agreement emphasize that moderate agreement is common for semantically rich coding schemes (Artstein and Poesio 2008). The cost and difficulty of producing consistent annotations across tens of thousands of documents further exacerbate these issues (Do et al. 2024). A tension therefore persists. Transfer learning models require training data that are both large and consistent, yet available methods for generating such data either require considerable human effort or apply only to narrow task structures. A hybrid approach offers a potential solution: LLMs can serve not as direct classifiers but as generators of labeled data that efficiently train transformer-based models for downstream tasks.

1.2. Hybrid Architectures and Knowledge Distillation

The idea of using a larger model to generate training data for a smaller, more efficient one has a well-established lineage in natural language processing. Knowledge distillation, the process of training a smaller, efficient model to replicate the outputs of a larger one (Hinton et al. 2015), was originally proposed for compressing neural networks and has since been extended to settings where LLMs produce annotations that train compact classifiers (Di Palo et al. 2024; Tan et al. 2024). In this paradigm, a decoder-based LLM (a generative model that produces text) interprets annotation guidelines and produces structured labels, while an encoder-based transformer (a model optimized for classification, such as BERT or XLM-RoBERTa) learns to reproduce the labeling logic with far greater inference efficiency. The result is a system that merges the efficacy and interpretive flexibility of generative models with the throughput and efficiency of discriminative ones.

The underlying principle is straightforward (see Figure 1). The LLM produces contextual annotations, humans audit and correct its outputs, and a transformer model is trained from these refined labels to apply them consistently to the full corpus. Recent work has explored this architecture in several forms. Farr, Manzonelli, Cruickshank, and West (2025) propose RED-CT, a systems design methodology that uses LLM-labeled data with confidence-based sampling

to train edge classifiers for computational social science tasks. Di Palo et al. (2024) introduce Performance-Guided Knowledge Distillation (PGKD), an active learning loop between an LLM teacher and a student classifier that achieves up to 130× inference speedup. In political science, LLM annotations can approach the quality of human expert coding (Heseltine and Clemm von Hohenberg 2024), and transformer classifiers trained on LLM-generated data achieve performance close to that of models trained on human annotations (Pangakis and Wolken 2024). Multi-stage designs in which LLMs generate preliminary labels, automated filters estimate confidence, and human coders adjudicate ambiguous cases have also shown promise (Wang, Kim, Rahman, Mitra, and Miao 2024a).

A critical concern with any LLM-to-classifier pipeline is the propagation of label noise. Egami, Hinck, Stewart, and Wei (2023) demonstrate that even when LLM annotations reach 80–90% accuracy, naive use of these predicted labels in downstream analyses can introduce substantial bias and invalid confidence intervals. Consequently, they created a Design-Based Supervised Learning framework that combines high-volume LLM labels with a small set of expert annotations to produce statistically valid estimates. This principled correction shows the importance of human validation within hybrid workflows and reinforces the view that LLM-generated labels should not be treated as ground truth but as an approximation that benefits from systematic quality control.

Despite these advances, existing hybrid systems remain fragmented. They generally rely on proprietary APIs, target narrow tasks or domains, and rarely provide reproducible infrastructures that other researchers can reuse. The opacity of LLM mechanisms further complicates validation, and as Törnberg (2024) and Atreja, Ashkinaze, Li, Mendelsohn, and Hemphill (2024) have noted, this has created what resembles a methodological "Wild West," where prompt design and implementation details can decisively influence results. Three gaps therefore continue to limit cumulative progress. First, validation practices remain inconsistent and few systematic comparisons exist between LLM-generated and human-generated training data under controlled conditions (Chae and Davidson 2025; Park and Montgomery 2025; Bernhard-Harrer, Ashour, Eberl, Tolochko, and Boomgaarden 2025; Baden, Pipal, Schoonvelde, and van der Velden 2022). Second, even well-documented pipelines frequently fail to reproduce because of undocumented preprocessing steps, missing code, or unstable model versions (Weber and Reichardt 2023; Ollion et al. 2024). Third, while individual components exist (LLM

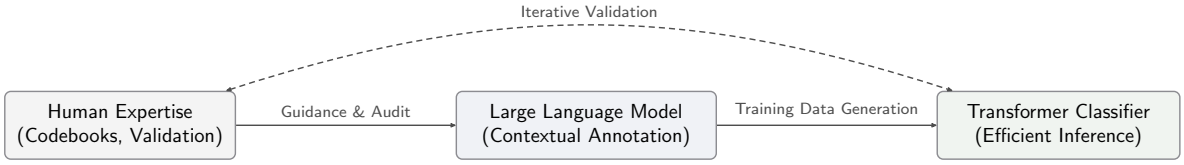

Figure 1: Conceptual structure of a hybrid annotation architecture integrating human, LLM, and transformer components.

annotation libraries, transformer training frameworks, evaluation suites), the field lacks an integrated open-source tool that deploys a complete hybrid architecture locally while ensuring full reproducibility, transparency, and accessibility.

Several open-source tools address parts of this gap. LLM annotation libraries facilitate label generation for very large corpora, and transformer training frameworks simplify fine-tuning. Yet none of these tools, to our knowledge, provides a fully local, end-to-end hybrid pipeline (from structured LLM annotation through transformer fine-tuning and with high-throughput inference on full corpus) with automatic documentation of all parameters, environment configurations, and model provenance for verifiable replication. Nor does any existing framework offer an accessible command-line interface that enables researchers to execute the entire process without programming expertise. Iterative approaches that combine machine learning with qualitative interpretation (Nelson 2020) have demonstrated the value of integrating computational and human expertise; what remains missing is the infrastructure to operationalize such integration across the full range of corpora that social scientists now study. The promise of hybrid architectures is well established; realizing it requires open, systematic, and locally executable pipelines that align scientific rigor with equitable access.

2. The LLM Tool Pipeline

LLM Tool was developed to address the methodological and infrastructural limitations identified above. It provides an open-source implementation of a fully local hybrid pipeline that unites decoder-based and encoder-based language models in a single reproducible workflow. While existing tools address individual components of this pipeline (LLM annotation, Tan et al. 2024; knowledge distillation, Di Palo et al. 2024; systems design methodologies, Farr et al. 2025), LLM Tool integrates the entire process from annotation through fine-tuning to high-throughput inference on the full corpus in a single locally executable package. The system enables any researcher to annotate large text corpora without reliance on commercial APIs or advanced programming skills.[1] It integrates transparency, accessibility, and empirical validation into a unified framework.

The pipeline combines two complementary components. Decoder-based LLMs interpret annotation guidelines and produce structured contextual labels, while transformer-based models generalize these annotations across entire corpora. Local execution through open-source LLMs removes dependency on external services and ensures the confidentiality of sensitive data. This configuration also eliminates recurring API costs that make annotation of very large corpora prohibitively expensive for most institutions. Once the LLM has produced an annotated dataset, the pipeline trains encoder-based models such as BERT and its derivatives (50+ models are available depending on languages of text-length) to reproduce the labeling

[1]The package also supports proprietary API providers; see Section B.

logic with far greater efficiency. This division of tasks allows the system to handle corpora of millions of sentences within days instead of months on a personal computer, a throughput impossible to reach through direct LLM inference alone.

The workflow operates entirely on local machines and can be executed through a command-line interface without needing any coding. Each stage of the workflow is automatically documented, including model parameters, prompts, software versions, and environment settings, which guarantees full traceability and replicability. The package integrates extensive technical infrastructure: several LLM provider architectures with fully local execution options, over 50 transformer models with automatic selection capabilities, comprehensive data characterization across 75+ languages, integrated quality assurance through multiple validation mechanisms, and multiple command-line interface modes for annotation, training, and validation workflows. Complete technical specifications are documented in Appendix B.

The LLM Tool pipeline follows four main phases (Figure 2). The first phase, *Input*, processes textual data from multiple formats and defines annotation tasks through structured prompts

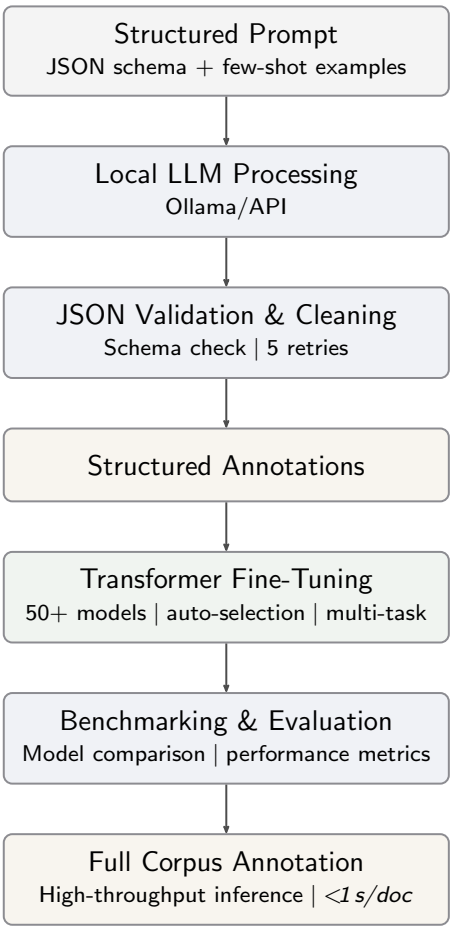

Figure 2: The LLM Tool pipeline, from structured prompts through annotation, training, and high-throughput inference on the full corpus.

with dynamic JSON schemas. Each prompt is validated automatically through on-the-fly data validation schemas (Pydantic models) to ensure that LLM outputs conform to the expected structure.

The second phase, *Quality Control*, verifies and cleans the raw LLM outputs. Each response is validated through successive checks, with up to five retries. The tool applies JSON repairs, regular expression cleaning, and schema validation to correct structural inconsistencies. All steps generate detailed logs. These procedures make the annotation process transparent and reproducible, which addresses the reproducibility problems observed in existing studies (Weber and Reichardt 2023; Ollion et al. 2024).

The third phase, *Model Training*, converts LLM-generated annotations into efficient transformer classifiers. The system provides access to more than 40 encoder architectures, including multilingual backbones (XLM-RoBERTa, mDeBERTa-v3), language-specific families (CamemBERT, FlauBERT, BARThez for French; DeBERTa-v3, RoBERTa, ELECTRA, Longformer, BigBird for English; and BERT variants for German, Spanish, Italian, Portuguese, Chinese, Arabic, Russian, and Hindi), and domain-specialized variants. During training, the tool also picks the best checkpoint and model's epoch automatically. At every epoch it computes a combined validation score (0.6 × macro F1 + 0.4 × micro F1 for multi-class and multi-label tasks, 0.7× F1 of the positive class + 0.3× macro F1 for binary ones), saves the checkpoint whenever this score improves, and discards the previous one (which save memory). The final model used downstream is therefore the best epoch of the run rather than the last one, without any manual intervention.

The final phase, *Output*, deploys the trained model for corpus-wide annotation and evaluation and produces outputs in multiple formats (whith compute parallelization included for maximizing on local machines large-scale deployment). The hybrid approach converts a one-time annotation investment into a reusable classifier whose cost is amortized across every subsequent run.

3. Methods and Data

3.1. Data Collection and Corpus Composition

Table 1: Corpus composition.

		Language		Source	
Dataset	**N**	**EN**	**FR**	**Parl.**	**Media**
Test set	1,549	791 (51%)	758 (49%)	798 (52%)	751 (48%)
Training (per LLM)	38,451	19,209 (50%)	19,242 (50%)	19,202 (50%)	19,249 (50%)

To validate the pipeline, we use two large databases of Canadian political and media texts.

The first is Agora+ (CAPP 2025a), which contains parliamentary debates from the federal and provincial (Quebec) legislatures. The second is Radar+ (CAPP 2025b), which aggregates media coverage from thirteen anglophone and francophone front pages. From the combined pool of the two databases, we draw two disjoint random samples. A training sample, used for LLM-based annotation, contains 38,451 sentences that each LLM annotates independently. A separate test sample, used for evaluation, contains 1,549 sentences that four researchers with expertise in Canadian politics manually annotated to serve as the ground truth. Because the two samples are drawn independently and share no sentences, evaluation is free of contamination. Since the corpus includes both English and French texts, multilingual classification capabilities can also be tested.

The two components of the dataset place very different demands on a classifier. Parliamentary debates include legislative discussions, question periods, and members' statements, where speakers often rely on procedural conventions, party-line framings, and technical policy vocabulary. The media subcorpus, drawn from thirteen anglophone and francophone front pages articles, speaks in a looser register shaped by editorial voic and the narrative conventions of political journalism. Evaluating the hybrid pipeline jointly on both registers therefore probes whether the same model can follow the codebook across formal and informal political discourse, rather than overfitting to the idiosyncrasies of a single genre.

The corpus was deliberately balanced along the two axes that most directly stress a bilingual classifier. English and French sentences are represented in near-equal proportions (19,209 and 19,242, respectively), and parliamentary and media texts contribute almost identically (19,202 and 19,249). The manually annotated test set was also structured to preserve this balance, with 51% English and 52% parliamentary sentences, so that aggregate performance figures cannot be inflated by over-representing the easier side of either axis.

Table 2: Annotation category distribution

Dimension	**Categories**	**Description**
Policy Themes	21	health, environment, economy, etc.
Specific Themes	2	welfare state, childcare
Political Parties	10	LPC, CPC, BQ, NDP, CAQ, etc.
Sentiment	3	positive, neutral, negative
Total	**36**	

3.2. Annotation Schema

The annotation schema Was created along a research project conducted in the Center for Public Policy Analysis (CAPP) and identifies four dimensions: policy themes, specific themes, political parties, and sentiment. Textual data are classified into one of 21 major public policy themes. These were designed to align with the Comparative Agendas Project (CAP) policy

categories, a validated international typology used by more than 27 countries and 38 affiliated institutions (Jones, Baumgartner, Theriault, Epp, Eissler, Lee, and Sullivan 2023). In addition, specific themes capture Canadian policy areas of interest to the research team: welfare state provisions and childcare. The political party dimension covers ten Canadian federal and Quebec provincial parties. The sentiment dimension distinguishes between positive, neutral, and negative framings.

3.3. Manual Annotation and Ground Truth

The ground-truth dataset comes from four annotators with expertise in Canadian politics (three M.A. students, one professor). The annotators were trained using the CAP codebook and provided with Canadian-specific guidelines. The Doccano platform was used as the annotation interface to ensure unbiased coding and to facilitate quality monitoring. The team began by coding 100 sentences; the research leader and an assistant then reviewed annotations and compared them to resolve conflicts. Each text received annotations from all four annotators; a consensus label was determined by majority vote for evaluation.

Figure 3 presents the per-dimension agreement scores and the class distributions across the annotation dimensions. Agreement metrics are computed per dimension from the four-annotator ratings on the 1,549 test sentences. For the three multi-label dimensions we report Krippendorff's α with the MASI distance (Passonneau 2006), which is sensitive to partial set overlap, and the mean of the per-label binary Fleiss' κ. For the single-label sentiment dimension we use the standard nominal α and Fleiss' κ. Political parties reach substantial agreement ($\alpha = 0.84$ [0.81, 0.86], $\kappa = 0.76$ [0.67, 0.79]), because party mentions are typically explicit and unambiguous. Policy themes and sentiment land in the moderate range ($\alpha = 0.53$ / $\kappa = 0.59$ and $\alpha = 0.49$ / $\kappa = 0.49$, respectively), which reflects the sometimes ambiguous CAP categories and the well-known subjectivity of sentiment judgements. Specific themes, the hardest dimension, show the lowest reliability ($\alpha = 0.32$ [0.27, 0.38], $\kappa = 0.48$ [0.40, 0.56]) due to the very narrow definitions and the sparsity of the two classes. Averaged across dimensions, $\bar{\alpha} = 0.54$ and $\bar{\kappa} = 0.58$. These levels are typical for fine-grained political coding tasks (Artstein and Poesio 2008; Krippendorff 2018) and justify the use of a majority-vote consensus rather than any single annotator's labels.

3.4. Experimental Design

Figure 4 summarizes the validation design. We selected five LLM configurations to span a range of architectures, sizes, and providers. Four models ran locally through Ollama, while one proprietary model (OpenAI GPT-5) served as an external benchmark. Four of the five models (Gemma, Mixtral, GPT-OSS, GPT-5) processed the full 38,451-sentence training pool with a 100% JSON validation success rate and zero structural parsing errors. Nemotron, by contrast,

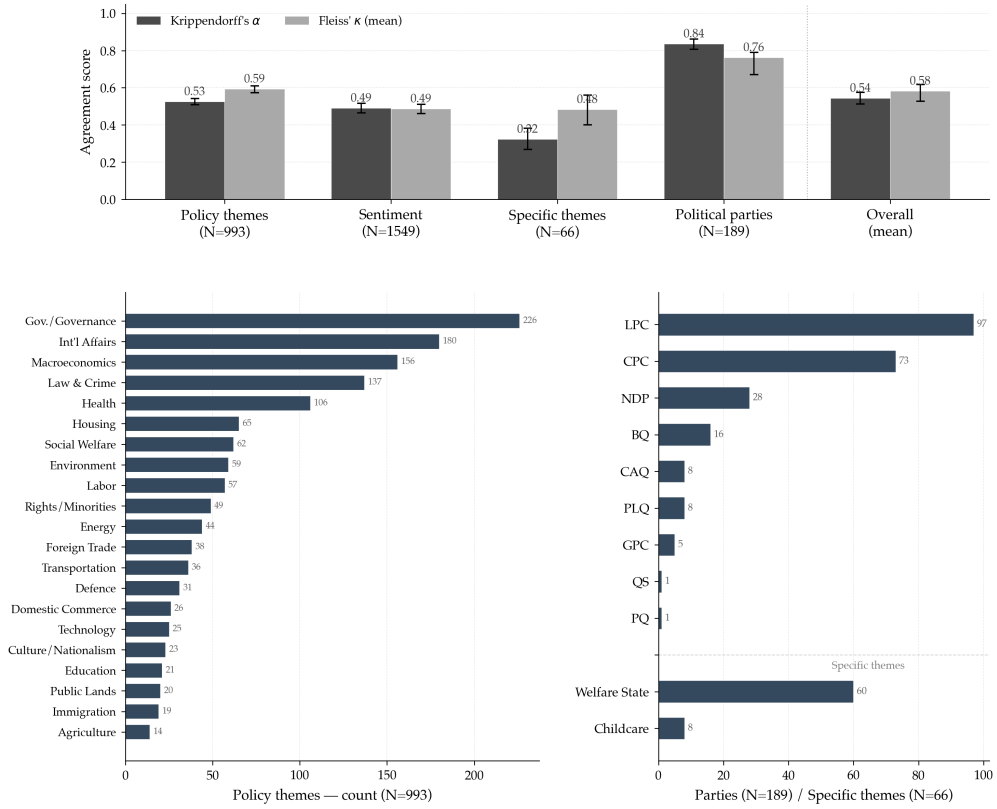

Figure 3: Ground-truth quality and category distributions (test set, N = 1,549). Top: per-dimension Krippendorff's α (MASI distance for multi-label dimensions, nominal for sentiment) and Fleiss' κ (mean over per-label binary κ for multi-label dimensions), with 95% bootstrap confidence intervals. Bottom left: distribution of the 21 CAP policy themes (N = 993 texts). Bottom right: political parties (N = 189 texts) and specific themes (N = 66 texts).

ran into persistent compliance failures: malformed JSON outputs and schema violations on French parliamentary inputs forced repeated re-prompting (58,672 API calls produced only 31,430 unique annotated sentences) and left 18.3% of the corpus uncovered. We accounted for this partial coverage in all subsequent analyses.

Table 3: LLM configurations used for annotation.

Model	**Parameters**	**Disk Size**	**Quant.**	**Context**	**Provider**	**Time/sent**
Gemma3	27B	17 GB	Full	128k	Ollama	19.7s
Nemotron	49B	42 GB	Full	4k	Ollama	8.5s
Mixtral	8×22B	80 GB	Full	64k	Ollama	5.5s
GPT-OSS	120B	65 GB	Q8	128k	Ollama	7.7s
GPT-5	N/A	N/A	N/A	128k	OpenAI	API

Per-sentence inference times ranged from 5.5 s for Mixtral up to 19.7 s for Gemma, with Nemotron and GPT-OSS sitting in between at 8.5 s and 7.7 s respectively. At that pace, annotating the full pool of 38,451 sentences took anywhere from about 2.4 days with Mixtral

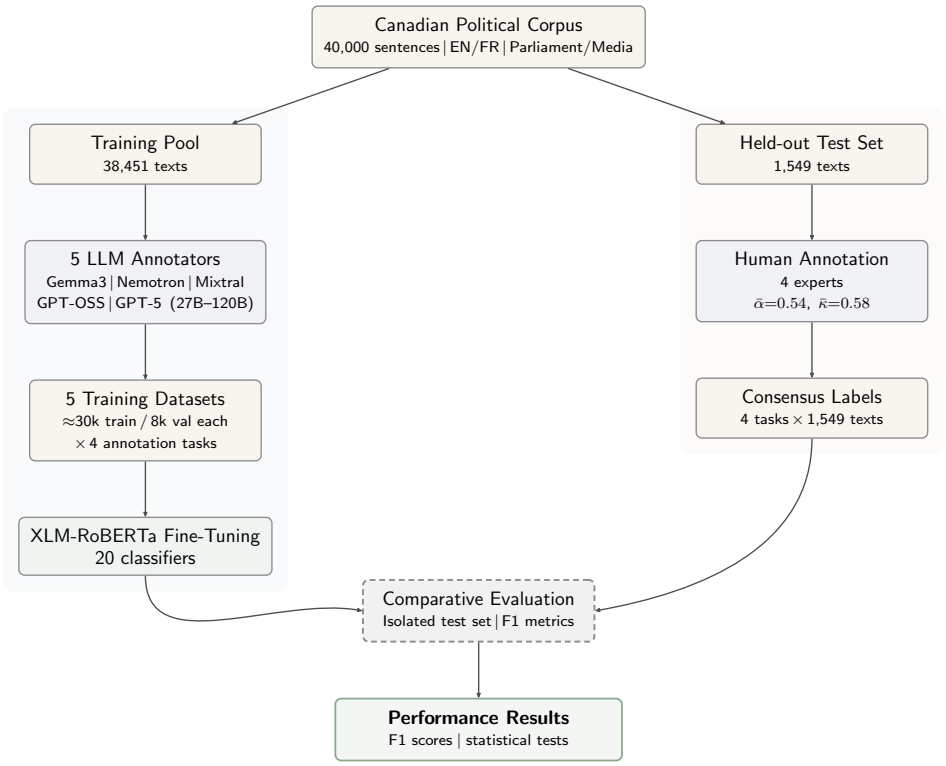

Figure 4: Validation pipeline. Five LLMs independently annotate 38,451 training texts across four tasks. Each annotation set trains an XLM-RoBERTa classifier (four task heads, 20 classifiers total). All classifiers are evaluated against consensus ground truth from four human annotators on an isolated test set of 1,549 texts. Training and test sentences are drawn from disjoint stratified samples of the corpus to ensure zero contamination.

to close to 8.8 days with Gemma, and roughly 3.4 days with GPT-OSS. Nemotron was a special case. Each sentence had to be re-prompted 1.87 times on average before returning a parsable output, which pushed its total compute time to 5.8 days (138.7 hours) and still left 18.3% of the corpus unannotated because of the compliance failures discussed above.

All five LLMs received the same prompt, which included system instructions defining an annotator role and context, the full list of category names with definitions, output specifications enforcing JSON formatting, and positive and negative few-shot examples. The annotation pipeline processed documents through several stages: preprocessing, batch construction, response parsing, and validation through a dynamically generated Pydantic schema with up to five retries and JSON repair for structural inconsistencies. The full English prompt used in the benchmark is reproduced in Appendix C.

According to the LLM Tool pipeline architecture (Section 2), each LLM-annotated corpus was then used to train a separate XLM-RoBERTa-base classifier. We trained one classifier per (LLM × task) pair: five LLM-annotated training sets (Gemma, Mixtral, Nemotron, GPT-OSS, GPT-5) crossed with four annotation tasks (policy themes, specific themes, political parties, sentiment), for a total of 20 classifiers. Each classifier was a separate XLM-RoBERTa-base fine-

tune with its own task-specific head. All five training runs shared identical hyperparameters (learning rate 2×10^{-5}, batch size 256, max 25 epochs, linear warmup) and the same random seed to ensure comparability. Within each run, the best checkpoint was selected automatically by the pipeline: the combined validation score described in Section 2 was recomputed at every epoch, the checkpoint was overwritten whenever that score improved, and training stopped after three epochs without improvement. The best checkpoint of each run was then used for the final evaluation on the held-out test set. Full hyperparameters and per-task training validation metrics are reported in Appendix A. All annotation and training was performed on consumer Apple Silicon hardware: a Mac Studio (M2 Ultra, 24 CPU cores, 60 GPU cores, 128 GB unified memory) and two MacBook Pro laptops (M4 Max, 16 CPU cores, 40 GPU cores, 128 GB unified memory each). LLM inference ran through Ollama; classifier training used Apple MPS GPU acceleration. Total training time ranged from 9.7 hours (Gemma) to 20.8 hours (GPT-5).

We report micro-averaged, macro-averaged, and weighted F1 scores. Micro F1 aggregated predictions across all classes and weighted each instance equally. Macro F1 averaged per-class scores and gave equal weight to each category regardless of prevalence. Weighted F1 adjusted class contributions by their support in the test set. Statistical significance was assessed through the Friedman test with Kendall's W, the Kruskal-Wallis H-test with η^2, post-hoc pairwise Wilcoxon signed-rank tests with Bonferroni correction, and bootstrap confidence intervals (10,000 resamples, 95% level). The unit of analysis for these tests was the per-class F1 score, with each class label treated as a block and each model as a treatment. Within-task tests used as many paired observations as the task contained classes, and the global Friedman test reported in the Results section pools per-class F1 scores across all four tasks (35 aligned observations per model, $\chi^2 = 43.38$, $W = 0.310$).

4. Results

This section takes up the question posed in the introduction: can a locally executable encoder–decoder pipeline reach near-human annotation quality, and does the choice of LLM annotator meaningfully change the downstream classifier? We approach the question from five complementary angles. *Overall performance* asks whether the five classifiers land in the same ballpark or split into distinguishable tiers on aggregate F1. *Category-specific performance* looks inside that aggregate to see whether some classes are systematically well or poorly handled across all classifiers. *Computational efficiency* quantifies what is gained, in wall-clock time, by replacing direct LLM inference with a distilled classifier on a large corpus. *Category prevalence* isolates the effect of test-set support on per-class F1, to distinguish genuine annotation failures from small-sample artefacts. Finally, *cross-lingual and cross-domain robustness* checks whether performance holds across English and French, and across parliamentary debates and news

media, as a pipeline that succeeds only on one language or one register would be of limited use for bilingual social-science research.

All five classifiers share the same XLM-RoBERTa-base backbone and the same hyperparameters; the only experimental variable is the LLM that produced their training labels. To keep the exposition readable, we write "GPT-5 classifier" as a shorthand for "the XLM-RoBERTa-base classifier trained on GPT-5 annotations," and likewise for the four other systems (GPT-OSS, Mixtral, Nemotron, and Gemma). Whenever we refer to an LLM as an annotator (e.g., its per-sentence latency, its JSON compliance rate), we name it without the "classifier" suffix; whenever we refer to downstream evaluation metrics, we are evaluating the corresponding XLM-RoBERTa classifier, not the LLM itself.

4.1. Overall Performance

Figure 5 presents Micro F1 scores for each classifier across the four annotation tasks. Two results stand out. First, the classifiers separate into two tiers: the GPT-5 classifier (mean Micro F1 = 0.689) and the GPT-OSS classifier (0.674) form a high-performing group; the Mixtral (0.518), Nemotron (0.517), and Gemma (0.510) classifiers cluster roughly 16 percentage points below. Second, task complexity shapes performance more than annotator size. Sentiment

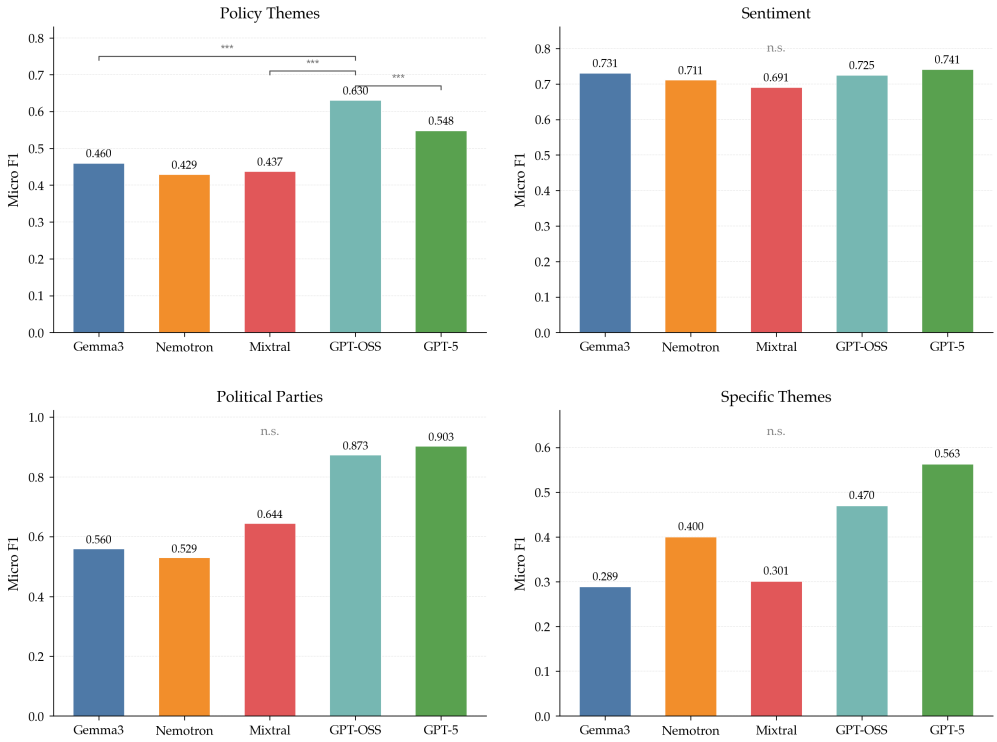

Figure 5: Micro F1 by annotation task. Brackets indicate significant pairwise differences (Wilcoxon signed-rank, Bonferroni-corrected; $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$). Sentiment is the easiest task; specific themes the hardest.

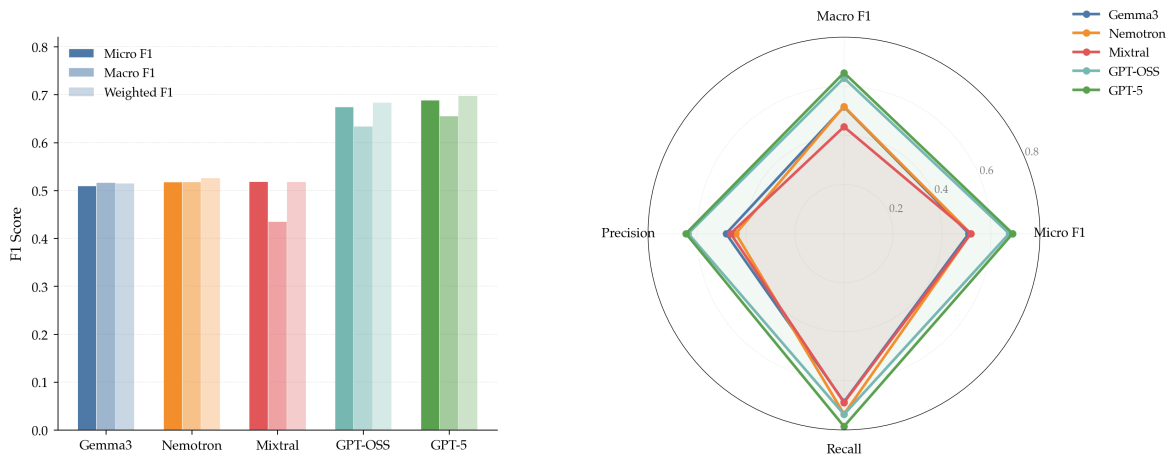

Figure 6: (a) Aggregate F1 metrics per classifier. (b) Multi-metric radar profiles. The GPT-5 and GPT-OSS classifiers dominate on every axis; the gap is widest on recall.

is the easiest task, with all five classifiers exceeding 0.69 and a five-point gap between the best and worst. Political party identification shows the sharpest differentiation (the GPT-5 classifier reaches 0.903, the Gemma classifier only 0.560). Policy themes, with 21 labels, range from 0.429 to 0.630 across classifiers. Specific themes prove the most difficult, and even the GPT-5 classifier reaches only 0.563, owing to severe class imbalance (welfare state has 60 test instances, childcare only 8).

The Friedman test confirms that these differences are significant at the global level ($\chi^2 = 43.38$, $p < 0.0001$, Kendall's $W = 0.310$). The Kruskal-Wallis test corroborates this finding ($H = 27.80$, $p < 0.0001$, $\eta^2 = 0.132$). Six of the ten pairwise Wilcoxon comparisons reach significance after Bonferroni correction: the GPT-OSS classifier is significantly above the Gemma, Nemotron, and Mixtral classifiers, and the GPT-5 classifier is significantly above the Gemma, Nemotron, and Mixtral classifiers. No significant difference exists between the GPT-5 and GPT-OSS classifiers ($p_{\text{corrected}} = 0.327$), nor within the Gemma/Nemotron/Mixtral classifier group. Figure 6 shows the aggregate metrics and multi-metric profiles across all tasks.

4.2. Category-Specific Performance

Figure 7 displays per-class F1 scores for every class of the four annotation tasks, with models as columns. Starting with the policy-themes block, well-defined domains achieve the strongest results when averaged across the five classifiers: foreign trade (mean F1 = 0.724), international affairs (0.704), health (0.678). Semantically broad categories prove harder; governments/governance (0.382) is overrepresented in training data yet dilutes classifier precision because it overlaps with nearly every other theme. Domestic commerce (0.184) lacks sufficient training signal.

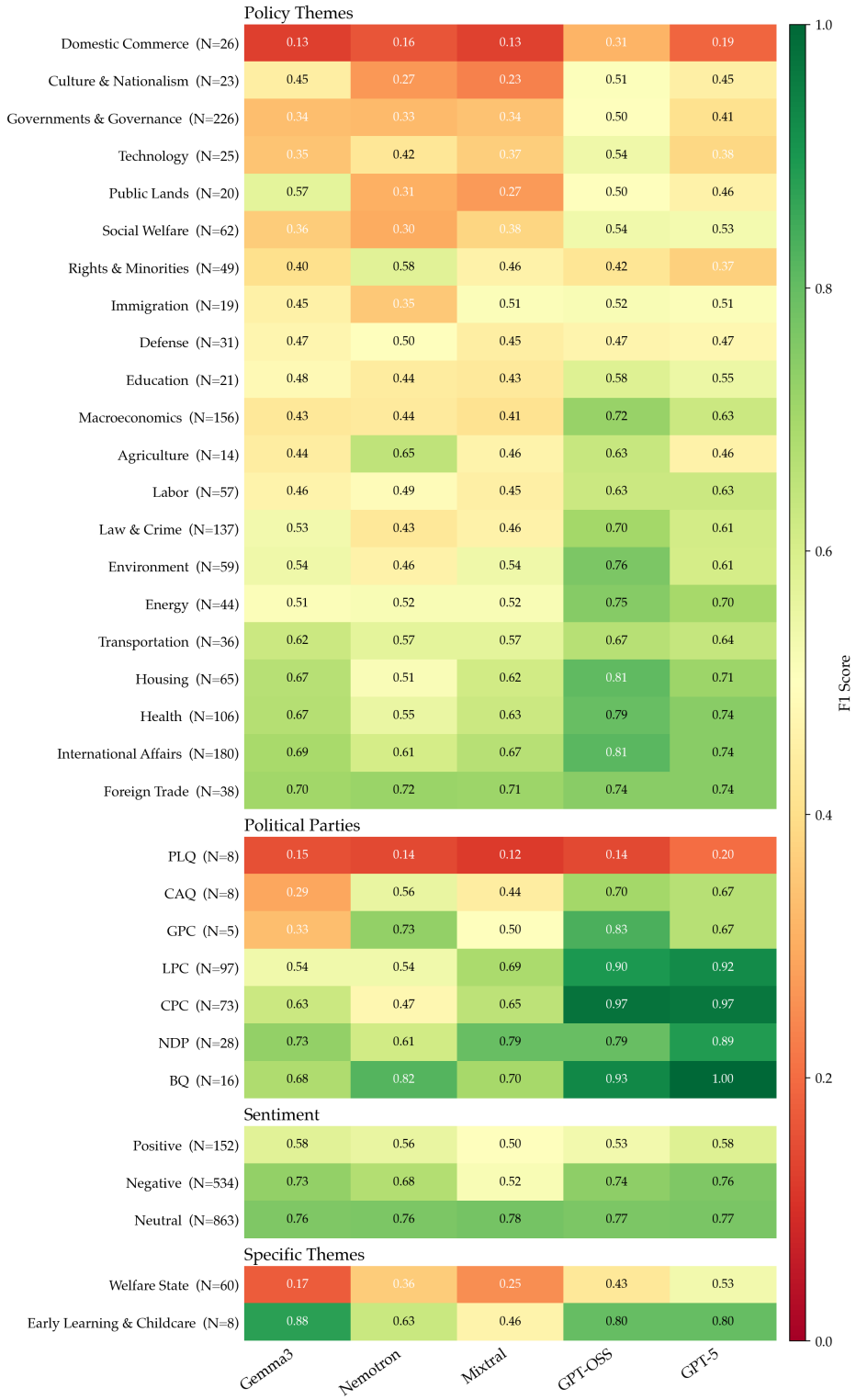

Figure 7: Per-class F1 scores across all four annotation tasks (policy themes, political parties, sentiment, specific themes). Within each task, rows are sorted by mean F1 (ascending) and test-set support (N) is shown for each class. The GPT-OSS classifier dominates the 21 policy themes, while GPT-5 takes over on the majority of the smaller label sets.

Interestingly, the GPT-OSS classifier is the strongest annotator on policy themes, where it outperforms the much larger GPT-5 model on 20 of the 21 classes. GPT-5, in turn, takes over on the smaller label sets: it wins 4 of the 10 political parties and 2 of the 3 sentiment categories in Figure 7. On the two-class specific-themes dimension, Gemma3 unexpectedly leads on childcare (F1 = 0.875) while GPT-5 takes welfare state (0.533), so the two top systems still effectively split the fine-grained work between them while sitting comfortably above the other three classifiers. First, this clearly shows that democratizing large-scale annotation while maintaining quality is achievable with the right model design. Second, this suggests that annotation quality depends more on the LLM's alignment with the task and its ability to follow instructions than on raw parameter count. The Mixtral classifier, despite being a smaller model, outperforms the Nemotron and Gemma classifiers on several categories, which may reflect differences in training data or architecture that make it better suited for structured annotation tasks.

For political parties, major federal parties are identified reliably (BQ: 0.826; NDP: 0.764; CPC: 0.739; LPC: 0.716); provincial parties with small test support prove harder (PLQ: 0.151; QS: 0.000). The per-model picture is even sharper than for policy themes. Task-level Micro F1 scores in Figure 5 already show the GPT-5 (0.903) and GPT-OSS (0.873) classifiers roughly 25–30 points above the Mixtral (0.644), Gemma (0.560), and Nemotron (0.529) classifiers on this dimension. The per-class breakdown in Figure 7 shows more precise results: GPT-5 and GPT-OSS dominate every federal party (BQ: 1.00 / 0.93; CPC: 0.97 / 0.97; LPC: 0.92 / 0.90; NDP: 0.89 / 0.79), while the three weaker systems rarely exceed 0.70 even on the largest classes. The two top classifiers also split the small provincial labels between them, with GPT-OSS leading on CAQ (0.70) and GPC (0.83) and GPT-5 edging ahead on PLQ (0.20). Sentiment follows the expected pattern. Neutral (0.767) outperforms negative (0.684) and positive (0.549), with the positive class suffering from lower prevalence and greater ambiguity.

4.3. Computational Efficiency

The hybrid architecture produces its clearest advantage on efficiency. Direct local LLM annotation requires between 5.5 s (Mixtral) and 19.7 s (Gemma) per sentence for the annotators that completed the full corpus. Annotating all 38,451 sentences consumed between 2.4 days (Mixtral) and 8.8 days (Gemma) of continuous local computation; GPT-OSS fell at 3.4 days. Nemotron, despite a fast per-call speed (8.5 s), accumulated 138.7 hours of computation due to compliance-related retries while annotating only 81.7% of the corpus. Once trained, the XLM-RoBERTa classifiers process the same corpus in approximately 32 minutes (≈50 ms per sentence), a speedup of 109× to 395× over the LLMs that produced their training labels.

This gap widens with corpus size (Figure 8b). For a corpus of 500,000 sentences, Mixtral would require roughly one month of direct LLM inference, GPT-OSS about 45 days, and

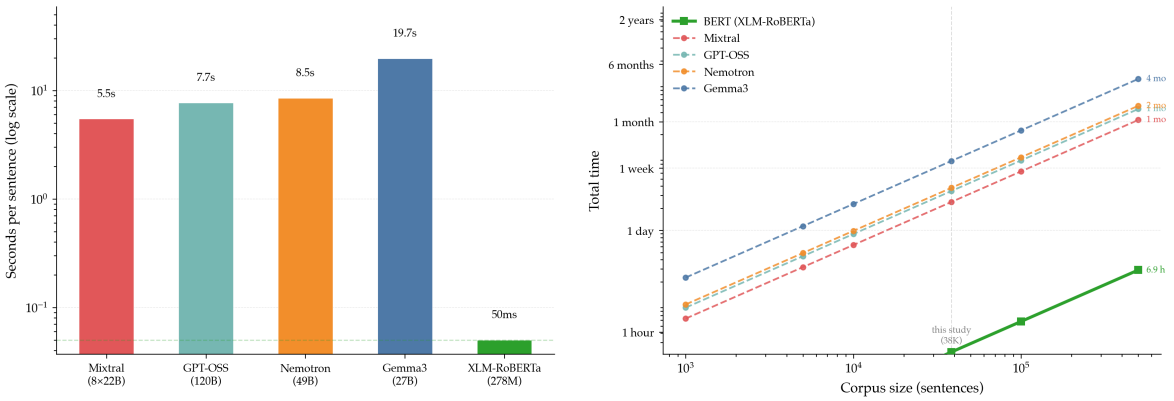

Figure 8: Computational efficiency. (a) Per-sentence inference time (log scale). (b) Projected total time by corpus size. BERT remains below one day even at 500 K sentences.

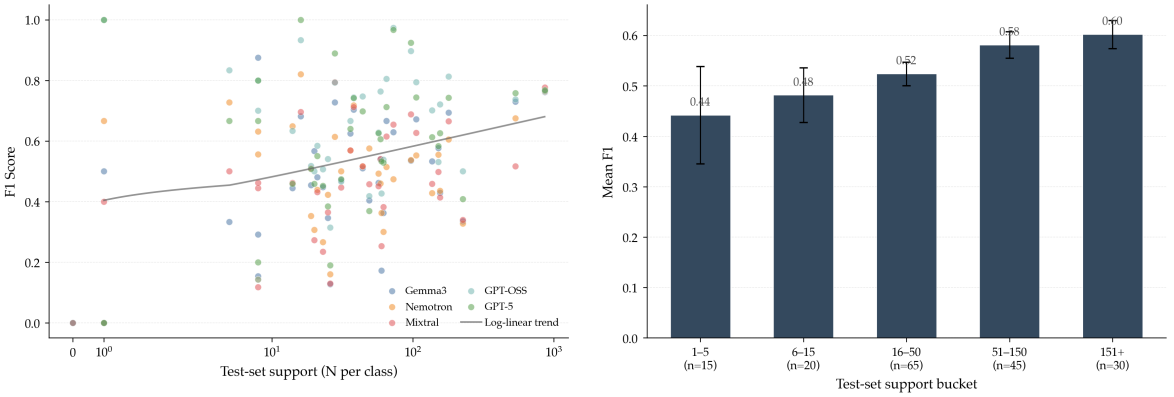

Figure 9: Effect of category prevalence on classification performance. (a) Per-class F1 vs. test-set support for all model–class pairs; the dashed line shows a log-linear trend. (b) Mean F1 by support bucket with standard errors.

Gemma nearly four months of continuous computation. The corresponding XLM-RoBERTa classifier processes the same corpus in under seven hours. The hybrid approach thus amortizes the initial annotation cost, as once the classifier is trained, it can serve indefinitely without further LLM computation.

4.4. Category Prevalence and Classification Performance

Beyond model choice, per-class performance tends to depend on category prevalence in the test set. Figure 9 quantifies this relationship. A clear log-linear trend emerges which shows that classes with fewer than 15 test instances rarely exceed F1 = 0.5, regardless of which LLM generated the training annotations. Categories with 151+ instances achieve a mean F1 of 0.60, while those with 1–5 instances average 0.44.

This finding has implications for the validation of the pipeline. The low performance on specific themes (welfare state: N = 60, childcare: N = 8) and rare parties (QS: N = 1, PQ: N = 1) is,

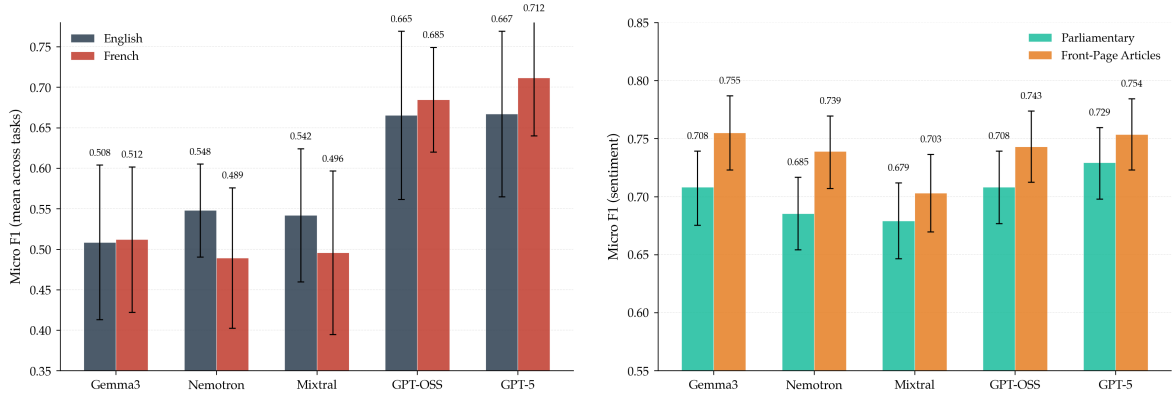

Figure 10: (a) Performance by language. The GPT-5 and GPT-OSS classifiers perform slightly better on French. (b) Performance by source type (sentiment task). Front-page articles yield higher F1 than parliamentary debates for all five classifiers.

at least partly, explained more by test-set sparsity than by annotation quality. The confidence scores of the XLM-RoBERTa classifiers confirm this pattern. The GPT-5 classifier yields the highest mean confidence on sentiment (0.884), the GPT-OSS classifier follows at 0.841, and the Gemma classifier shows wider distributions with a lower mean (0.777). Classifiers trained on cleaner LLM outputs are both more accurate and more certain in their predictions.

4.5. Cross-Lingual and Cross-Domain Robustness

The corpus comprises two languages (English, French) and two sources (parliamentary debates, front-page articles). Figure 10 examines whether the pipeline performs consistently across these dimensions.

The GPT-5 and GPT-OSS classifiers perform slightly better on French (GPT-5 classifier: 0.712 vs. 0.667). This may reflect the XLM-RoBERTa encoder's strength on Romance languages. The Mixtral (surprisingly) and Nemotron classifiers favour English (Mixtral classifier: 0.542 vs. 0.496). For source type, front-page articles yield higher sentiment F1 than parliamentary debates for all five classifiers (e.g., GPT-5 classifier: 0.754 vs. 0.729). Because the backbone and training parameters are held constant, these asymmetries are driven by the quality of the LLM annotations themselves rather than by any property of the downstream classifier.

5. Discussion

The central finding of this study is that the raw parameter count of the annotating LLM is a poor predictor of downstream classifier performance. All five XLM-RoBERTa classifiers share the same architecture, hyperparameters, and training procedure, so the only experimental variable is the LLM that produced their training labels. Performance splits sharply into two tiers. The classifiers trained on GPT-5 and GPT-OSS annotations sit roughly 16 percentage

points ahead of those trained on Gemma, Nemotron, or Mixtral annotations, and the ranking does not follow the annotator's parameter count. Mixtral, despite its 8×22B parameter count, generated 98 unique theme labels instead of the 21 prescribed categories and 57 party labels instead of 10. Nemotron produced persistent failures to output valid JSON, which forced repeated re-prompting (58,672 API calls produced only 31,430 unique annotated sentences) and left 18.3% of the corpus uncovered. The cleaner annotations produced by GPT-5 and GPT-OSS, especially their stricter adherence to the prescribed label set, are consistent with the hypothesis that instruction-following and schema compliance mediate downstream accuracy more directly than capacity (Chae and Davidson 2025). We cannot fully separate this fidelity channel from other latent model properties (training data, alignment procedures, tokenization), so the finding should be read as an observed association rather than a causal association.

However, this result carries a strong implication for researchers choosing between LLM sources. *The largest or most expensive model is not necessarily the best annotator*; what matters is whether the model can follow a codebook consistently. The GPT-OSS classifier, whose training labels came from a fully open-source LLM running locally, achieved performance statistically indistinguishable from the GPT-5 classifier, whose training labels came from the proprietary OpenAI API. High-quality annotation no longer requires dependence on commercial APIs. For teams concerned with data sovereignty, cost control, or long-term reproducibility, this finding is directly actionable. The LLM Tool pipeline can now be used with confidence even running with open-source models, with classifier performance that matches or exceeds that of the best proprietary APIs. The democratization of LLM-based and large-scale annotation is thus achievable without sacrificing quality, and with our own personal computers.

Nonetheless, propagation of label noise through hybrid pipelines deserves careful attention. Egami et al. (2023) already demonstrated that even moderate labeling error rates can introduce substantial bias when predicted labels are used in downstream analyses. Our design mitigates this risk by evaluating classifiers against a human-validated test set rather than treating LLM annotations as ground truth. Nevertheless, the study does not directly compare classifiers trained on LLM-generated labels with those trained on human annotations. The pipeline demonstrates that LLM-annotated data can produce effective classifiers, but the question of how these compare to classifiers trained on expert labels remains here unaddressed.

The stance of the authors, however, is that classifiers *should always be evaluated against human-validated ground truth.* Social science research is a conceptual and interpretive endeavour, which means that annotating and validating data against human judgment is not just a methodological choice but a substantive commitment to the interpretive nature of the phenomena under study. The LLM Tool pipeline does not seek to replace human judgment with machine-generated labels; rather, it offers a way to leverage the strengths of LLMs in generating training data while still anchoring evaluation in human-validated ground truth.

This means that, concretely, when humans disagree while annotating and evaluating a model,

they tend to refine their codebook and annotation criteria, and improve their research. Without this, social science research loses its own reflexivity and epistemological foundations. What the LLM Tool pipeline offers, however, is a way to alleviate the bottleneck of human annotation while still maintaining a human-validated benchmark for evaluation. Instead of annotating thousands of sentences to train a classifier, researchers can use LLMs to generate a large training set, and then evaluate the resulting classifier against a smaller, human-annotated test set, and re-iterate until satisfaction. This approach allows for scalability without sacrificing the interpretive rigor that is central to social science research.

A quantitative detail reinforces this argument. The inter-annotator agreement among our own human experts (mean $\bar{\alpha} = 0.54$ across the four dimensions, rising to 0.62 when the narrowly defined specific-themes dimension is excluded) sits within the moderate range that methodological reviews report for semantically rich coding tasks (Artstein and Poesio 2008). In other words, the phenomena we ask the classifiers to recover are themselves contested at the human level. Trained coders disagree, negotiate, and ultimately converge. It is exactly this contested, interpretive zone that the two weakest LLM annotators failed to navigate. Read this way, the 16-percentage-point gap between the two tiers of classifiers is not a verdict on raw model capacity but a measurement of how faithfully each annotator respected the interpretive frame set by human coders, which is consistent with iterative frameworks that integrate computational methods with qualitative judgment (Nelson 2020).

The fragmentation and opacity that characterize much of the existing literature on LLM-based annotation (Weber and Reichardt 2023; Ollion et al. 2024) motivated a design principle throughout this work. Every component of the pipeline should be transparent, reproducible, and executable without specialized infrastructure. Individual components have been validated in prior work (LLM-based annotation, Ornstein et al. 2025; Rathje et al. 2024; knowledge distillation to compact classifiers, Di Palo et al. 2024; Farr et al. 2025; systematic evaluation frameworks, Park and Montgomery 2025), yet the integration of these into a single locally executable tool with automatic documentation addresses a gap that has limited cumulative progress in the field. The systematic comparison of five LLM sources under identical conditions further contributes a benchmark that quantifies the cost-quality tradeoff and provides a template for the kind of rigorous model evaluation that computational social science need (Bernhard-Harrer et al. 2025; Park and Montgomery 2025).

Several limitations qualify these findings. The evaluation relies on a two corpus of Canadian political and mediatic texts in two languages; generalization to other domains, languages, and annotation schemas requires further validation. The test set ($N = 1{,}549$) provides adequate statistical power for sentiment and policy themes, but specific themes ($N = 66$) and rare political parties ($N < 5$) lack sufficient support for reliable per-class evaluation. The pipeline uses a single annotation pass per LLM; ensemble strategies, active learning methods (Di Palo et al. 2024), or bias-correction techniques (Egami et al. 2023) could improve training data

quality. Computational costs remain substantial, as annotating the corpus required between 2.4 and 8.8 days per model for full coverage, and Nemotron's compliance failures illustrate that not all local models can reliably complete an annotation campaign. The 18.3% of texts that Nemotron failed to annotate consist predominantly of French parliamentary inputs, so its training set is not a random subsample of the corpus and its reported performance should be interpreted as conditional on a narrower, English-leaning distribution rather than as a fair comparison with the models that achieved full coverage. While the trained classifiers amortize this investment over every subsequent use, the initial annotation phase may be prohibitive for resource-constrained teams. Finally, the rapid pace of LLM development means that the specific model comparisons presented here will age; the methodological framework and open-source infrastructure are designed to outlast any particular generation of models by adding quite simply new LLMs to the annotation stage.

Conclusion

The question that opened this article asked whether an encoder–decoder pipeline, executable locally on commodity hardware, could produce annotations of a quality close enough to human coders to be trusted in downstream social-science analyses, and whether the choice of LLM annotator still mattered once its outputs had been distilled into a task-specific classifier. The twenty classifiers evaluated here answer both halves of the question at once. The best distilled system reaches a mean Micro F1 of 0.667 against a human-validated test set on four genuinely difficult coding tasks, processes 38,451 sentences in roughly half an hour on a single workstation, and was trained from labels that an open-weight 120B model produced locally without any call to a commercial API. At the same time, the 16-percentage-point gap that separates the two tiers of classifiers confirms that the identity of the annotating LLM is a first-order methodological decision rather than an implementation detail, and it is a decision that cannot be read off the model's parameter count.

The ranking of our five annotators is not predicted by their size but by their willingness to stay inside the codebook their users designed. Mixtral invented 98 theme labels and 57 party labels for an eleven-class scheme, Nemotron left almost a fifth of the corpus untouched through repeated schema violations, and it is precisely those two annotators whose distilled classifiers sit at the bottom of the ranking. Instruction-following and schema compliance, in other words, are not stylistic features of an annotator but the mechanism through which a coder's interpretive frame survives the transition to machine labels. This finding should inform how researchers select an LLM annotator and, more importantly, what they monitor during annotation: the first quality signal a team should watch is not raw accuracy against a held-out set but adherence to the codebook that human experts negotiated in the first place.

Two other results carry implications beyond the specific Canadian corpus analysed here. The

first is that the GPT-OSS classifier, trained on labels produced entirely by an open-weight model running on local hardware, is statistically indistinguishable from the GPT-5 classifier, whose labels came from a proprietary API. High-quality hybrid annotation no longer requires a cloud subscription, and the entire pipeline described in this paper runs on the kind of workstation that most research groups already own. The second is that the two-orders-of-magnitude speedup distillation delivers is what turns this infrastructure into a practical tool for corpora whose size neither manual coding nor direct LLM inference could realistically handle. Together, these two results dissolve a practical barrier that has long constrained LLM-based annotation along institutional and financial lines.

None of this dissolves the harder methodological questions that LLM-based annotation raises for the social sciences, and the contribution of this paper is deliberately framed to leave room for them. The conditions under which machine-generated labels can substitute for human judgment, the biases that training data may encode and propagate, and the standards of validation appropriate for this new class of tools all require sustained, discipline-level investigation. Our own position is that such investigation should keep human-validated ground truth at the evaluative centre of any claim made with these tools, not because human coders are infallible but because the negotiation through which they converge on a coding scheme is precisely what makes the resulting measurements interpretable as social-science data. What the hybrid approach offers is an open, reproducible, and locally executable environment in which that negotiation can continue. The complete LLM Tool codebase, the five annotation runs analysed in this paper, the twenty distilled classifiers, and every script required to regenerate the figures and tables of the manuscript are publicly released to support replication, extension, and critical scrutiny.

A. From Raw Annotations to Final Benchmark: Protocol, Hyperparameters, and Full Results

This appendix documents the experimental protocol, hyperparameters, and complete benchmark results used throughout the paper. Every number reported below is reproduced by the scripts in `paper/scripts/` from the raw logs bundled in the repository (see `paper/README.md` for the reproduction pipeline).

A.1. Corpus and evaluation protocol

The evaluation set is a stratified 1,549-sentence sample of the Canadian political corpus balanced across language (English/French) and source (parliamentary speeches from Agora+, media front pages from Radar+). A disjoint 38,451-sentence sample forms the training pool from which each LLM's annotations are drawn (Table 1 in the main text). Each sentence carries four annotation tasks: *sentiment* (3 classes), *political party* (10 classes), *policy themes* (21 multi-label classes), and *specific themes* (2 multi-label classes). The test set is held out from all training procedures and used only for the final benchmark reported in Table 7.

A.2. LLM annotator configurations

Table 3 in the main text lists the five LLMs used as annotators. Four of them (Gemma 3, Nemotron, Mixtral, GPT-OSS) are executed locally through Ollama on a workstation GPU; GPT-5 is queried through the OpenAI API. All local models were run at their published precision or Q8 quantization; context windows match the model's native configuration. Per-sentence runtime is measured as the mean wall-clock time per successful annotation call, excluding retries.

Nemotron completed only 81.7% of the corpus (31,430 unique sentences out of 38,451): repeated JSON-schema violations on French parliamentary inputs exhausted the retry budget and blocked the remainder. The cleaned annotation log contains 58,672 successful API calls, meaning that each annotated sentence required on average 1.87 retries before producing a parsable output. This compliance gap, rather than hardware performance, explains both the partial coverage and the inflated wall-clock time reported in the efficiency statistics below.

A.3. BERT classifier: hyperparameters

All classifiers are fine-tuned from `xlm-roberta-base` (278M parameters) with the hyperparameters reported in Table 4. The same configuration is applied to every (LLM × task) pair, so that downstream differences reflect only differences in the training labels produced by each LLM. Hyperparameter values were chosen to saturate the available GPU memory (batch size 256 with gradient accumulation) and to give each task enough epochs (25) to converge

before early stopping intervenes (patience of 3). The random seed is fixed to 42 and the train/validation/test split is identical across all runs.

Table 4: BERT fine-tuning hyperparameters.

Parameter	**Value**
Base model	xlm-roberta-base
Learning rate	2e-05
Batch size	256
Max epochs	25
Warmup ratio	0.1
Weight decay	0.01
Optimizer	AdamW
Scheduler	Linear
Max sequence length	512
Early stopping patience	3
Random seed	42

A.4. Training dynamics

Table 5 reports validation performance for each (LLM × task) pair at the best epoch selected by early stopping. Three patterns are visible. First, GPT-5 and GPT-OSS classifiers dominate on every task, often by 10–20 points of macro F1. Second, best-epoch values are bimodal: sentiment saturates very early (5–8 epochs) while the harder multi-label tasks (political parties, specific themes, policy themes) continue to improve up to epoch 20–25, which justifies the `max_epochs=25` setting. Third, the macro F1 collapse for Mixtral on political parties (0.074) and policy themes (0.119) is a direct consequence of label-space pollution in the training annotations: Mixtral emitted 57 distinct party strings instead of the 10 prescribed by the codebook, and 98 theme strings instead of 21. The classifier inherits this noise and is unable to recover a clean decision boundary.

A.5. Inference efficiency

Table 6 compares the per-sentence and corpus-level inference cost of the five LLM annotators against the fine-tuned XLM-RoBERTa classifier. The BERT stage runs the full 38,451-sentence corpus in roughly 32 minutes on a single consumer GPU, which corresponds to a 109× to 395× speedup over direct LLM annotation depending on the model. This speedup is the practical motivation for the hybrid design: LLM inference is used once to generate training labels, after which the distilled classifier handles all subsequent inference at a fraction of the cost.

A.6. Main benchmark results

Table 5: BERT training validation performance per LLM source and task.

LLM Source	Task	F1 Macro	F1 Micro	Accuracy	Best Epoch
Gemma3	Political Parties	0.3246	0.3975	0.6528	25
Gemma3	Sentiment	0.4793	0.6881	0.6876	6
Gemma3	Specific Themes	0.3725	0.2379	0.7350	9
Gemma3	Policy Themes	0.3887	0.5488	0.2235	18
Nemotron	Political Parties	0.4058	0.3541	0.7269	13
Nemotron	Sentiment	0.5732	0.6311	0.6306	5
Nemotron	Specific Themes	0.2793	0.2520	0.9488	21
Nemotron	Policy Themes	0.4239	0.4822	0.1459	13
Mixtral	Political Parties	0.0738	0.4212	0.7491	16
Mixtral	Sentiment	0.3521	0.7695	0.7743	6
Mixtral	Specific Themes	0.1066	0.2598	0.8960	6
Mixtral	Policy Themes	0.1186	0.5066	0.1716	15
GPT-OSS	Political Parties	0.6222	0.8707	0.9661	24
GPT-OSS	Sentiment	0.7528	0.7748	0.7753	5
GPT-OSS	Specific Themes	0.4954	0.6006	0.9653	21
GPT-OSS	Policy Themes	0.6193	0.7193	0.5168	24
GPT-5	Political Parties	0.6273	0.9340	0.9844	23
GPT-5	Sentiment	0.7741	0.7887	0.7882	8
GPT-5	Specific Themes	0.7176	0.6846	0.9757	20
GPT-5	Policy Themes	0.6404	0.7218	0.3948	21

Table 6: Inference time comparison: LLM annotation vs. BERT classification.

Model	Type	Parameters	Time/sent	Total (38k)	Speedup
Gemma3	LLM	27B	19.7s	210.7h	395×
Nemotron	LLM	49B	8.5s	90.9h	170×
Mixtral	LLM	8×22B	5.5s	58.4h	109×
GPT-OSS	LLM	120B	7.7s	82.3h	154×
GPT-5	LLM	Proprietary	API	—	—
XLM-RoBERTa	BERT	278M	50ms	32min	1×

Table 7 ranks the five XLM-RoBERTa classifiers by micro F1 on the held-out 1,549-sentence test set. The ranking is consistent across all four metrics (micro F1, macro F1, weighted F1, precision). The gap between the top two classifiers (GPT-5 and GPT-OSS classifiers) and the remaining three is substantial and statistically significant (Section A.8).

A.7. Per-task breakdown

Table 8 decomposes the main ranking across the four annotation tasks. The GPT-5/GPT-OSS classifier advantage is widest on the label-rich tasks (policy themes and specific themes), where

Table 7: Overall model performance on the benchmark test set (N=1,549). Models ranked by mean Micro F1 across all annotation tasks.

Rank	Model	Micro F1	Macro F1	Weighted F1	Precision
1	GPT-5	**0.6887**	0.6551	0.6977	0.6454
2	GPT-OSS (120B)	0.6745	0.6334	0.6832	0.6360
3	Mixtral (8×22B)	0.5182	0.4349	0.5178	0.4630
4	Nemotron (49B)	0.5175	0.5180	0.5256	0.4415
5	Gemma 3 (27B)	0.5097	0.5164	0.5148	0.4814

codebook adherence during annotation matters most. On the three-class sentiment task, differences narrow and all five classifiers operate within a few F1 points of each other.

Table 8: Model performance by annotation task (Micro F1).

Model	Policy Themes	Sentiment	Political Parties	Specific Themes	Mean
Gemma3	0.4597	0.7308	0.5598	0.2887	0.5097
Nemotron	0.4290	0.7114	0.5295	0.4000	0.5175
Mixtral	0.4370	0.6908	0.6442	0.3009	0.5182
GPT-OSS	0.6300	0.7250	0.8732	0.4697	0.6745
GPT-5	0.5476	0.7411	0.9031	0.5630	0.6887

A.8. Statistical tests

We test whether the cross-classifier differences in Table 7 are likely to arise from chance. A Friedman test on the per-task F1 scores rejects the null hypothesis of equal ranks ($\chi^2 = 43.38$, $p < 10^{-4}$, Kendall's $W = 0.310$); a Kruskal–Wallis test on the pooled scores gives $H = 27.80$, $p < 10^{-4}$, $\eta^2 = 0.132$. Pairwise Wilcoxon signed-rank tests with Bonferroni correction confirm that the GPT-OSS classifier is significantly better than the three lower-ranked classifiers (Gemma 3, Nemotron, Mixtral) at $\alpha = 0.05$, and that the GPT-5 classifier is significantly better than the Gemma 3, Nemotron, and Mixtral classifiers. The GPT-5-vs-GPT-OSS classifier difference is not significant ($p_{\text{corr}} = 0.327$). The full set of comparisons is reported in Table 9.

A.9. Category-level performance

Table 10 aggregates per-class F1 scores within each task so that we can identify which label groups drive the overall ranking. Category-level inspection confirms that the top two systems are not merely better on average but dominate on almost every individual class, including low-frequency ones.

Table 9: Statistical tests for model comparison.

Test	**Statistic**	**p-value**	**Effect Size**
Friedman test	$\chi^2 = 43.38$	< 0.0001	Kendall's W = 0.31
Kruskal-Wallis test	H = 27.8	< 0.0001	$\eta^2 = 0.132$
Significant pairwise comparisons (Wilcoxon with Bonferroni correction)			
Gemma3 vs. GPT-OSS	—	0.0076	$r = 0.561$
Gemma3 vs. GPT-5	—	0.0036	$r = 0.604$
Nemotron vs. GPT-OSS	—	< 0.001	$r = 0.758$
Nemotron vs. GPT-5	—	0.0051	$r = 0.587$
Mixtral vs. GPT-OSS	—	0.0071	$r = 0.564$
Mixtral vs. GPT-5	—	0.0103	$r = 0.555$

Table 10: Best performing models by annotation dimension.

Dimension	**Best Model**	**Micro F1**	**Macro F1**	**Weighted F1**
Policy Themes	GPT-OSS	0.6300	0.6147	0.6552
Sentiment	GPT-5	0.7411	0.7030	0.7463
Political Parties	GPT-5	0.9031	0.7014	0.8960
Specific Themes	GPT-5	0.5630	0.6667	0.5647

B. Inside the Package: Architecture, Back-ends, and the Interactive Command-Line Interface

This appendix documents the components of the `llm_tool` package that implement the hybrid pipeline evaluated in the paper. Every claim below has been verified against the current source tree; where a feature differs from the experimental configuration used for the benchmark, we flag the difference explicitly. The package is organised around four subsystems: (i) an LLM annotation layer, (ii) a BERT-family training and inference layer, (iii) a schema- and language-aware validation layer, and (iv) an interactive command-line interface that coordinates the three.

B.1. LLM annotation layer

The annotation layer currently supports three provider back-ends with a common `generate()` interface, so that models can be swapped through configuration without code changes:

- **Commercial API.** OpenAI (`openai` client), with a unified error-handling layer that covers retry, exponential backoff, and typed exception remapping. OpenAI is the only commercial API provider wired end-to-end in the current release; scaffolding for additional providers exists in the source tree but is not exposed to the annotation pipeline and is not used in the experiments reported in this paper.
- **Local execution.** Ollama (for managed quantised checkpoints) and `llama-cpp-python` (for raw GGUF files). Both are optional dependencies that the package imports lazily, so that users who rely exclusively on the OpenAI API, or conversely on local models, are not forced to install unused back-ends.

The benchmark reported in this paper used Ollama for Gemma 3, Nemotron, Mixtral, and GPT-OSS, and the OpenAI back-end for GPT-5. No data from the local runs left the workstation on which the experiments were executed.

Prompts are produced by a prompt-engineering module that formats the task codebook, few-shot examples (if supplied), and the target text into a provider-specific template. Model output is then passed through a dynamic schema validator (Section B.3). When validation fails, the call is retried up to `max_retries_per_row` times (default 3); the benchmark reported here uses the package default.

B.2. BERT training and inference layer

The training layer provides two wrappers over Hugging Face Transformers. The first, `llm_tool/trainers/models.py`, exposes 13 canonical multilingual and language-specific families (BERT, XLM-RoBERTa, CamemBERT, and ten language-specific BERT variants). The

second, `llm_tool/trainers/sota_models.py`, extends this registry with 54 additional state-of-the-art encoder classes covering DeBERTa v3, RoBERTa, ELECTRA, ALBERT, BigBird, Longformer, LED, and French-specialised variants including CamemBERT v2, FlauBERT, and BARThez. In total the registry exposes 67 model wrappers; the paper uses only `xlm-roberta-base` so that every benchmark run differs only in the training labels supplied by each LLM.

Model selection is rule-based: a heuristic recommender scores candidate architectures against the detected languages, sequence-length distribution, class imbalance, and available hardware, and returns a ranked short-list. The user may either accept the recommendation or override it interactively.

The inference layer parallelises batch classification across CPU and GPU workers and preserves row-level metadata (input identifiers, detected language, model confidence) alongside the predicted labels. Batch sizes are configurable; the package default is 16, while the benchmark reported here uses 256 to saturate the training GPU (Table 4).

B.3. Schema validation and language detection

Each annotation task is declared as a JSON codebook. At run time, the package constructs a matching Pydantic model with `build_dynamic_schema()`, which validates every LLM response against the declared label space, enforces required fields, and coerces multi-label outputs into canonical form. Invalid responses are retried and, if still invalid after the retry budget, logged as failures.

Language detection relies on the `lingua-language-detector` library as the primary back-end, with heuristic fallbacks for edge cases. The library supports up to 75 languages; the package default configuration activates a curated subset of 30 languages that covers the European, East Asian, and South Asian language families most frequently encountered in social-science corpora. Detection scores drive the automatic selection of language-specific classifiers in the BERT layer and are also recorded as per-sentence metadata.

Cost control is limited to *estimation*: before dispatching an API run, the package computes a token-based cost projection from the current provider price list and prints it for user confirmation. There is no automatic spend cap; budget enforcement remains the user's responsibility.

B.4. Interactive command-line interface

The package exposes an interactive menu through `llm_tool/cli/advanced_cli.py` with the following workflow modes:

1. *The Annotator* — zero-shot LLM annotation of a raw text file with export to Label

Studio or Doccano.

2. *The Annotator Factory* — end-to-end pipeline that chains LLM annotation, training-data preparation, and BERT fine-tuning in a single session.
3. *Training Arena* — dedicated BERT training and benchmarking workspace. It runs an 18-step wizard (steps 5 through 18 in the package source) that walks through dataset inspection, language strategy, model-family selection, hyperparameter configuration, training, and evaluation.
4. *BERT Annotation Studio* — high-throughput inference over a trained classifier with parallel GPU/CPU execution and resumable sessions.
5. *Validation Lab* — quality-control workspace that ingests multiple annotator exports and computes inter-annotator agreement (pairwise Cohen's κ and Krippendorff's α on nominal data). This mode is under active development and is marked as such in the CLI.
6. *Resume Center* — session registry that lists partially completed annotation and training runs so that they can be resumed from the last checkpoint.
7. An *exit* option.

Every mode persists its configuration, progress, and outputs to a session directory, which supports both resumability and post-hoc reproducibility: re-opening a session recovers the exact parameters that produced the associated artefacts. These session directories, together with the deterministic training scripts and the frozen test set, are what allows the benchmark in Section A.6 to be re-run independently from the raw logs shipped with this replication package.

C. The Full English Annotation Prompt Passed to Every LLM

The following is the single English prompt passed to all five LLMs in the benchmark. It is reproduced verbatim from `paper/prompts/prompt_test_llm_tool.txt`. The French variant uses an equivalent translation with the same structure.

```
You are a text annotator specializing in Canadian and Quebec public policy. Analyze and
summarize the main Canadian public policy themes addressed in these one-sentence excerpts
from media articles or parliamentary debate transcripts, using the following categories to
structure the output in JSON format. You must write exclusively in JSON without any
explanatory text.

The categories must be clear, and appropriate values must be used:

Expected keys:
- "themes_long": "macroeconomics" if the sentence relates to a macroeconomics theme such as
interest rates, unemployment, monetary policy, the national budget, budget cuts, deficit,
fiscal policy, taxation, industrial policy, price control, or government spending;
"rights_liberties_minorities_discrimination" if the sentence relates to a theme of rights and
liberties, discrimination of groups such as minorities or by gender, age, people with
disabilities, freedom of expression, the right to vote, the right to freedom, Indigenous
affairs, or anti-government activities;
"health" if the sentence relates to a theme of health such as the healthcare system or
medical coverage, health insurance, the pharmaceutical industry, healthcare workers, mental
health, the cost of healthcare, and disease prevention or related topics;
"agriculture" if the sentence relates to a theme of agriculture, food standards, fishing,
subsidies to farmers, farm animals or related topics;
"labor" if the sentence relates to a theme of labor, unions, pensions, workers, employment
conditions or related topics;
"education" if the sentence relates to a theme of education such as elementary, secondary or
higher education, vocational education and training, special-needs, underprivileged students;
"environment" if the sentence relates to an environmental theme, including air pollution,
recycling, species and forests, climate change policy, and land conservation (but NOT water
management which is public_lands);
"energy" if the sentence relates to an energy theme, including nuclear energy, electric, gas,
coal, oil, petroleum, renewable energy, energy conservation, carbon tax, and carbon pricing;
"immigration" if the sentence relates to a theme of immigration, refugees, and citizenship;
"transportation" if the sentence relates to a theme of transportation or infrastructure such
as highways, railways, maritime routes, or air transport;
"law_and_crime" if the sentence relates to a theme of DOMESTIC law, crime, or family issues
such as white collar crime, juvenile crime, illegal drugs, prisons, police, and court
proceedings (but NOT international conflicts or wars);
"social_welfare" if the sentence relates to a theme of social protection such as low-income
assistance, elderly assistance, disabled assistance, volunteer associations and child care;
```

```
"housing" if the sentence relates to a theme of housing, urban affairs, neighborhood
development, community revitalization, urban and rural development, housing assistance, and
housing access for at-risk groups;
"domestic_commerce" if the sentence relates to a theme of domestic commerce, including banks,
securities, goods, consumer finance, insurance regulation, disaster relief, tourism, consumer
protection regulation, and sports regulation;
"defense" if the sentence relates to a theme of defense, including alliances, intelligence,
nuclear weapons, military aid, military procurement, military installations, military
operations, and military personnel;
"technology" if the sentence relates to a theme of space, science, technology, or
communications, weather forecasting or computers;
"foreign_trade" if the sentence relates to a theme of foreign trade such as trade agreements,
exports, private investments, competitiveness, tariffs, imports, and exchange rates;
"international_affairs" if the sentence relates to a theme of international affairs such as
international aid, international conflicts, wars, foreign governments, foreign leaders,
international resource exploitation, international finance, developing countries, or any
public policy from outside Canada;
"governments_governance" if the sentence relates to a theme of government operations,
intergovernmental relations, political affairs, elections, political campaigns, party
leadership, prime minister or minister resignations, political appointments, political
nominations, political scandals, parliamentary procedures, public service, or postal service
(but NOT budget/fiscal policy which is macroeconomics);
"public_lands" if the sentence relates to a theme of public lands, water management, water
restrictions, drinking water, territorial issues, national parks, and issues related to
territorial dependencies, territories, and devolution of powers;
"culture_nationalism" if the sentence relates to a theme of culture or nationalism;
"null" if the sentence explicitly does not relate to any of these themes.
- "political_parties_long": "LPC" if the sentence mentions or refers to the Liberal Party of
Canada, its MPs, or candidates;
"CPC" if the sentence mentions or refers to the Conservative Party of Canada, its MPs, or
candidates;
"BQ" if the sentence mentions or refers to the Bloc Québécois, its MPs, or candidates;
"NDP" if the sentence mentions or refers to the New Democratic Party, its MPs, or candidates;
"GPC" if the sentence mentions or refers to the Green Party of Canada, its MPs, or
candidates;
"CAQ" if the sentence mentions or refers to the Coalition Avenir Québec, its MPs, or
candidates;
"PLQ" if the sentence mentions or refers to the Liberal Party of Quebec, its MPs, or
candidates;
"PQ" if the sentence mentions or refers to the Parti Québécois, its MPs, or candidates;
"QS" if the sentence mentions or refers to Québec solidaire, its MPs, or candidates;
"PCQ" if the sentence mentions or refers to the Conservative Party of Quebec, its MPs, or
candidates;
"null" if the sentence explicitly does not relate to any of these parties.
- "specific_themes_long": "welfare_state" if the sentence relates to a theme of the welfare
state;
"early_learning_childcare" if the sentence relates to a theme of early education or
childcare;
```

```
"null" if the sentence explicitly does not relate to any of these themes.
- "sentiment_long": "positive" if the sentence has a positive tone in relation to one of the
21 "themes_long", "political_parties_long" or "specific_themes_long";
"neutral" if the sentence has a neutral tone in relation to one of the 21 "themes_long",
"political_parties_long" or "specific_themes_long";
"negative" if the sentence has a negative tone in relation to one of the 21 "themes_long",
"political_parties_long" or "specific_themes_long".

**Instructions**
- Strictly follow the structure of the keys defined above. Verify that the output matches
these keys (for example, the categories of "specific_themes_long" should not be placed in
"themes_long" and vice versa).
- Ensure that all keys are present in the JSON, using `null` when necessary.
- Do not include keys that are not defined in the expected keys above.
- Write exclusively the JSON without any additional comments or explanations.
- Include a "themes_long" if the sentence relates to public policy OR to political affairs
(elections, campaigns, government actions, political scandals, resignations, appointments).
- Identify the "themes_long" and "specific_themes_long" from a Canadian perspective.
- Indicate multiple "themes_long" if multiple "themes_long" are present.
- Indicate multiple "political_parties_long" if multiple "political_parties_long" are
present.
- Indicate only one "sentiment_long" for each sentence.

**CRITICAL RULES - Follow these precisely:**
1. Budget cuts, deficit, fiscal policy, government spending → use "macroeconomics" (NOT
"governments_governance")
2. Foreign governments, foreign leaders, US policy, international conflicts, wars → use
"international_affairs" (NOT "law_and_crime")
3. Water management, water restrictions, drinking water → use "public_lands" (NOT
"environment")
4. Carbon tax, carbon pricing, carbon rebate → use BOTH "environment" AND "energy"
5. Political campaigns, elections, party leadership, resignations → use
"governments_governance"
6. When a sentence discusses a foreign country's policy (e.g., US bitcoin reserve, UK
election), use "international_affairs"
7. International armed conflicts and terrorism are "international_affairs", NOT
"law_and_crime"
8. "law_and_crime" is ONLY for domestic Canadian crime, police, courts, prisons
9. When the government mentions "cuts" or "spending" in a fiscal context, the PRIMARY theme
is "macroeconomics"

**Example of an annotation:**

Pierre Poilievre a vivement dénoncé la nouvelle mesure du gouvernement imposant des normes
environnementales strictes sur l'industrie pétrolière.

**Example of JSON:**
```

```
{
  "themes_long": ["environment", "energy", "labor", "macroeconomics"],
  "political_parties_long": ["CPC"],
  "specific_themes_long": null,
  "sentiment_long": "negative"
}

**Example: Budget cuts**

Va-t-il annoncer qu'il annule ces coupes?

{
  "themes_long": ["macroeconomics"],
  "political_parties_long": null,
  "specific_themes_long": null,
  "sentiment_long": "neutral"
}

**Example: US policy (from Canadian perspective)**

Cynthia Lummis of Wyoming would require all bitcoin held by any federal agency be transferred
to the Treasury to be held in a strategic bitcoin reserve.

{
  "themes_long": ["international_affairs"],
  "political_parties_long": null,
  "specific_themes_long": null,
  "sentiment_long": "neutral"
}

**Example: International conflict**

The maelstrom of killing that left almost 1,200 Israelis and foreigners dead led inevitably
to more violence and death.

{
  "themes_long": ["international_affairs"],
  "political_parties_long": null,
  "specific_themes_long": null,
  "sentiment_long": "negative"
}

**Example: Water restrictions**

Officials are asking Calgarians to limit their non-essential water use.

{
  "themes_long": ["public_lands"],
```

```
  "political_parties_long": null,
  "specific_themes_long": null,
  "sentiment_long": "neutral"
}

**Example: Carbon rebate**

Canadians who filed their taxes will receive the new Canada carbon rebate amounts.

{
  "themes_long": ["macroeconomics", "environment", "energy"],
  "political_parties_long": null,
  "specific_themes_long": null,
  "sentiment_long": "neutral"
}

**Example: Parliamentary affairs**

Julian accused the Official Opposition of disrespecting the institution of Parliament.

{
  "themes_long": ["governments_governance"],
  "political_parties_long": null,
  "specific_themes_long": null,
  "sentiment_long": "negative"
}

**Example: Multiple political parties**

The NDP and the Liberals have agreed on a confidence-and-supply agreement that will keep the
government in power until 2025.

{
  "themes_long": ["governments_governance"],
  "political_parties_long": ["NDP", "LPC"],
  "specific_themes_long": null,
  "sentiment_long": "neutral"
}

**Example: Bloc Québécois**

Le Bloc québécois a exigé des mesures pour protéger la langue française dans les entreprises
fédérales au Québec.

{
  "themes_long": ["culture_nationalism", "labor"],
  "political_parties_long": ["BQ"],
  "specific_themes_long": null,
```

```
  "sentiment_long": "neutral"
}

**Example: Provincial party (CAQ)**

François Legault's CAQ government announced new immigration targets for Quebec, reducing the
number of temporary workers allowed in the province.

{
  "themes_long": ["immigration", "governments_governance"],
  "political_parties_long": ["CAQ"],
  "specific_themes_long": null,
  "sentiment_long": "neutral"
}

**Example: Welfare state**

Le gouvernement conservateur propose de couper dans les programmes d'aide sociale destinés aux
familles à faible revenu.

{
  "themes_long": ["social_welfare", "macroeconomics"],
  "political_parties_long": ["CPC"],
  "specific_themes_long": ["welfare_state"],
  "sentiment_long": "negative"
}

**Example: Early learning and childcare**

The federal childcare program has reduced fees to an average of $10 a day, making early
childhood education more accessible to working families.

{
  "themes_long": ["social_welfare", "education"],
  "political_parties_long": null,
  "specific_themes_long": ["early_learning_childcare"],
  "sentiment_long": "positive"
}

**Example: Multiple specific themes with political party**

Les libéraux ont promis d'investir massivement dans les garderies à 10$ et de renforcer le
filet social pour les aînés.

{
  "themes_long": ["social_welfare", "macroeconomics"],
  "political_parties_long": ["LPC"],
  "specific_themes_long": ["early_learning_childcare", "welfare_state"],
```

```
  "sentiment_long": "positive"
}

**Example: Parti Québécois**

Le Parti Québécois a dénoncé l'insuffisance des transferts fédéraux en santé, affirmant que le
Québec mérite sa juste part.

{
  "themes_long": ["health", "macroeconomics", "governments_governance"],
  "political_parties_long": ["PQ"],
  "specific_themes_long": null,
  "sentiment_long": "negative"
}

**Example: Green Party**

Elizabeth May and the Green Party are calling for an immediate halt to all new fossil fuel
projects in Canada.

{
  "themes_long": ["environment", "energy"],
  "political_parties_long": ["GPC"],
  "specific_themes_long": null,
  "sentiment_long": "negative"
}

Follow this structure for each sentence analyzed. No other comments or additional details
beyond the requested JSON structure and the specified categories should be added.

**Expected JSON Keys**
{
  "themes_long": "",
  "political_parties_long": "",
  "specific_themes_long": "",
  "sentiment_long": ""
}
```

References

Alizadeh, Meysam, Kubli, Maël, Samei, Zeynab, Dehghani, Shirin, Zahedivafa, Mohammadmasiha, Bermeo, Juan D., Korobeynikova, Maria, and Gilardi, Fabrizio. Open-source LLMs for text annotation: a practical guide for model setting and fine-tuning. *Journal of Computational Social Science*, 8(1):17, December 2024. ISSN 2432-2725. doi:10.1007/s42001-024-00345-9. URL https://doi.org/10.1007/s42001-024-00345-9.

Artstein, Ron and Poesio, Massimo. Survey Article: Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596, December 2008. doi:10.1162/coli.07-034-R2. URL https://aclanthology.org/J08-4004/.

Atreja, Shubham, Ashkinaze, Joshua, Li, Lingyao, Mendelsohn, Julia, and Hemphill, Libby. Prompt Design Matters for Computational Social Science Tasks but in Unpredictable Ways, June 2024. doi:10.48550/arXiv.2406.11980. URL http://arxiv.org/abs/2406.11980. arXiv:2406.11980 [cs].

Baden, Christian, Pipal, Christian, Schoonvelde, Martijn, and van der Velden, Mariken A. C. G. Three Gaps in Computational Text Analysis Methods for Social Sciences: A Research Agenda. *Communication Methods and Measures*, 16(1):1–18, 2022. doi:10.1080/19312458.2021.2015574.

Bernhard-Harrer, Jana, Ashour, Randa, Eberl, Jakob-Moritz, Tolochko, Petro, and Boomgaarden, Hajo. Beyond standardization: a comprehensive review of topic modeling validation methods for computational social science research. *Political Science Research and Methods*, pages 1–19, June 2025. ISSN 2049-8470, 2049-8489. doi:10.1017/psrm.2025.10008. URL https://www.cambridge.org/core/product/identifier/S2049847025100083/type/journal_article.

Bhat, Savita and Varma, Vasudeva. Large Language Models As Annotators: A Preliminary Evaluation For Annotating Low-Resource Language Content. In *Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems*, pages 100–107, Bali, Indonesia, 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.eval4nlp-1.8. URL https://aclanthology.org/2023.eval4nlp-1.8.

Bonikowski, Bart, Luo, Yuchen, and Stuhler, Oscar. Politics as Usual? Measuring Populism, Nationalism, and Authoritarianism in U.S. Presidential Campaigns (1952–2020) with Neural Language Models. *Sociological Methods & Research*, 51(4):1721–1787, November 2022. ISSN 0049-1241. doi:10.1177/00491241221122317.

Breznau, Nate, Rinke, Eike Mark, Wuttke, Alexander, et al. Observing Many Researchers Using the Same Data and Hypothesis Reveals a Hidden Universe of Uncertainty. *Proceedings of the National Academy of Sciences*, 119(44):e2203150119, 2022. doi:10.1073/pnas.2203150119.

CAPP. Agora+: Canadian parliamentary debates dataset. Centre d'analyse des politiques publiques, Université Laval, 2025a. URL to be provided by authors.

CAPP. Radar+: Canadian news coverage dataset. Centre d'analyse des politiques publiques, Université Laval, 2025b. URL to be provided by authors.

Chae, Youngjin and Davidson, Thomas. Large Language Models for Text Classification: From Zero-Shot Learning to Instruction-Tuning. *Sociological Methods & Research*, 2025. doi:10.1177/00491241251325243. Online First.

Conneau, Alexis, Khandelwal, Kartikay, Goyal, Naman, Chaudhary, Vishrav, Wenzek, Guillaume, Guzmán, Francisco, Grave, Edouard, Ott, Myle, Zettlemoyer, Luke, and Stoyanov, Veselin. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747/.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. 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*, pages 4171–4186. Association for Computational Linguistics, 2019. doi:10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Di Palo, Flavio, Singhi, Prateek, and Fadlallah, Bilal H. Performance-Guided LLM Knowledge Distillation for Efficient Text Classification at Scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3675–3687. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.emnlp-main.215. URL https://aclanthology.org/2024.emnlp-main.215/.

Do, Salomé, Ollion, Étienne, and Shen, Rubing. The Augmented Social Scientist: Using Sequential Transfer Learning to Annotate Millions of Texts with Human-Level Accuracy. *Sociological Methods & Research*, 53(3):1167–1200, August 2024. ISSN 0049-1241, 1552-8294. doi:10.1177/00491241221134526. URL https://journals.sagepub.com/doi/10.1177/00491241221134526.

Dong, Xiangjue, Wang, Yibo, Yu, Philip S., and Caverlee, James. Probing Explicit and Implicit Gender Bias through LLM Conditional Text Generation, 2023. doi:10.48550/arXiv.2311.00306. URL http://arxiv.org/abs/2311.00306. arXiv:2311.00306 [cs].

Dreier, Sarah K., Serrano, Sofia, Gade, Emily K., and Smith, Noah A. Troubles in Text: Using Natural Language Processing to Recognize Government Rationalizations for Rights Abuses. *The Journal of Politics*, 87(3):1221–1226, July 2025. ISSN 0022-3816, 1468-2508. doi:10.1086/733001. URL https://www.journals.uchicago.edu/doi/10.1086/733001.

Egami, Naoki, Hinck, Musashi, Stewart, Brandon M., and Wei, Hanying. Using Imperfect Surrogates for Downstream Inference: Design-based Supervised Learning for Social Science Applications of Large Language Models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d862f7f5445255090de13b825b880d59-Abstract-Conference.html.

Farr, David, Manzonelli, Nico, Cruickshank, Iain, and West, Jevin. RED-CT: A Systems Design Methodology for Using LLM-labeled Data to Train and Deploy Edge Linguistic Classifiers. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 58–67. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-industry.5/.

Gilardi, Fabrizio, Alizadeh, Meysam, and Kubli, Maël. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30): e2305016120, 2023. ISSN 0027-8424, 1091-6490. doi:10.1073/pnas.2305016120. URL https://pnas.org/doi/10.1073/pnas.2305016120.

Grimmer, Justin, Roberts, Margaret E., and Stewart, Brandon M. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton University Press, 2022.

Halterman, Andrew and Keith, Katherine A. Codebook LLMs: Evaluating LLMs as Measurement Tools for Political Science Concepts. *Political Analysis*, 34(2):188–204, April 2026. ISSN 1047-1987, 1476-4989. doi:10.1017/pan.2025.10017. URL http://arxiv.org/abs/2407.10747. arXiv:2407.10747 [cs].

Heseltine, Michael and Clemm von Hohenberg, Bernhard. Large language models as a substitute for human experts in annotating political text. *Research & Politics*, 11(1): 20531680241236239, January 2024. ISSN 2053-1680. doi:10.1177/20531680241236239.

Hinton, Geoffrey, Vinyals, Oriol, and Dean, Jeff. Distilling the Knowledge in a Neural Network, 2015. URL https://arxiv.org/abs/1503.02531. arXiv:1503.02531 [cs, stat].

Jones, Bryan D., Baumgartner, Frank R., Theriault, Sean M., Epp, Derek A., Eissler, Rebecca, Lee, Cheyenne, and Sullivan, Miranda E. Policy Agendas Project: Topics Codebook. Comparative Agendas Project, 2023. URL https://www.comparativeagendas.net/pages/master-codebook.

Krippendorff, Klaus. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, Thousand Oaks, CA, 4th edition, 2018. ISBN 9781506395661.

Laurer, Moritz, van Atteveldt, Wouter, Casas, Andreu, and Welbers, Kasper. Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning

with Deep Transfer Learning and BERT-NLI. *Political Analysis*, 32(1):84–100, 2024. doi:10.1017/pan.2023.20.

Matsuo, Akitaka and Fukumoto, Kentaro. Legislators' sentiment analysis supervised by legislators. *Political Science Research and Methods*, pages 1–17, September 2025. ISSN 2049-8470, 2049-8489. doi:10.1017/psrm.2025.10048. URL https://www.cambridge.org/core/product/identifier/S2049847025100484/type/journal_article.

McGee, Robert W. Is Chat Gpt Biased Against Conservatives? An Empirical Study, 2023. doi:10.2139/ssrn.4359405. URL https://papers.ssrn.com/abstract=4359405. 51 citations (Crossref/DOI) [2024-10-14].

Mets, Mark, Karjus, Andres, Ibrus, Indrek, and Schich, Maximilian. Automated stance detection in complex topics and small languages: The challenging case of immigration in polarizing news media. *PLOS ONE*, 19(4):e0302380, 2024. ISSN 1932-6203. doi:10.1371/journal.pone.0302380. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0302380. Publisher: Public Library of Science.

Nelson, Laura K. Computational Grounded Theory: A Methodological Framework. *Sociological Methods & Research*, 49(1):3–42, 2020. doi:10.1177/0049124117729703.

Ollion, Étienne, Shen, Rubing, Macanovic, Ana, and Chatelain, Arnault. The dangers of using proprietary LLMs for research. *Nature Machine Intelligence*, 6(1):4–5, January 2024. doi:10.1038/s42256-023-00783-6. URL https://hal.science/hal-04825849. Publisher: Nature Research.

Ornstein, Joseph T., Blasingame, Elise N., and Truscott, Jake S. How to Train Your Stochastic Parrot: Large Language Models for Political Texts. *Political Science Research and Methods*, 13(2):264–281, 2025. doi:10.1017/psrm.2024.64.

Pangakis, Nicholas and Wolken, Sam. Knowledge Distillation in Automated Annotation: Supervised Text Classification with LLM-Generated Training Labels. In *Proceedings of the Sixth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS 2024)*, pages 113–131, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.nlpcss-1.9.

Park, Ju Yeon and Montgomery, Jacob M. Toward a framework for creating trustworthy measures with supervised machine learning for text. *Political Science Research and Methods*, pages 1–17, September 2025. ISSN 2049-8470, 2049-8489. doi:10.1017/psrm.2025.10042. URL https://www.cambridge.org/core/product/identifier/S2049847025100423/type/journal_article.

Passonneau, Rebecca. Measuring Agreement on Set-valued Items (MASI) for Semantic and Pragmatic Annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL https://aclanthology.org/L06-1392/.

Rathje, Steve, Mirea, Dan-Mircea, Sucholutsky, Ilia, Marjieh, Raja, Robertson, Claire E., and Van Bavel, Jay J. GPT is an Effective Tool for Multilingual Psychological Text Analysis. *Proceedings of the National Academy of Sciences*, 121(34):e2308950121, 2024. doi:10.1073/pnas.2308950121.

Rozado, David. The Political Biases of ChatGPT. *Social Sciences*, 12(3):148, 2023. doi:10.3390/socsci12030148.

Tan, Zhen, Li, Dawei, Wang, Song, Beigi, Alimohammad, Jiang, Bohan, Bhattacharjee, Amrita, Karami, Mansooreh, Li, Jundong, Cheng, Lu, and Liu, Huan. Large Language Models for Data Annotation and Synthesis: A Survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 930–957. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.emnlp-main.54. URL https://aclanthology.org/2024.emnlp-main.54/.

Törnberg, Petter. Best Practices for Text Annotation with Large Language Models. *Sociologica*, 18(2):67–85, October 2024. ISSN 1971-8853. doi:10.6092/issn.1971-8853/19461. URL https://sociologica.unibo.it/article/view/19461. Number: 2.

Törnberg, Petter. Large Language Models Outperform Expert Coders and Supervised Classifiers at Annotating Political Social Media Messages. *Social Science Computer Review*, 43(6):1181–1195, 2025. ISSN 0894-4393. doi:10.1177/08944393241286471. URL https://doi.org/10.1177/08944393241286471. Publisher: SAGE Publications Inc.

Wang, Xinru, Kim, Hannah, Rahman, Sajjadur, Mitra, Kushan, and Miao, Zhengjie. Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–21, Honolulu, HI, USA, May 2024a. doi:10.1145/3613904.3641960.

Wang, Yu, Qu, Wen, and Ye, Xin. Selecting Between BERT and GPT for Text Classification in Political Science Research, November 2024b. doi:10.48550/arXiv.2411.05050. URL http://arxiv.org/abs/2411.05050. arXiv:2411.05050 [cs].

Wankmüller, Sandra. Introduction to Neural Transfer Learning With Transformers for Social Science Text Analysis. *Sociological Methods & Research*, 53(4):1676–1752, November 2024. ISSN 0049-1241. doi:10.1177/00491241221134527. URL https://doi.org/10.1177/00491241221134527. Publisher: SAGE Publications Inc.

Weber, Maximilian and Reichardt, Merle. Evaluation is all you need. Prompting Generative Large Language Models for Annotation Tasks in the Social Sciences. A Primer using Open Models, December 2023. doi:10.48550/arXiv.2401.00284. URL http://arxiv.org/abs/2401.00284. arXiv:2401.00284 [cs].

Ziems, Caleb, Held, William, Shaikh, Omar, Chen, Jiaao, Zhang, Zhehao, and Yang, Diyi. Can Large Language Models Transform Computational Social Science? *Computational Linguistics*, 50(1):237–291, March 2024. ISSN 0891-2017. doi:10.1162/coli_a_00502. URL https://doi.org/10.1162/coli_a_00502.

Affiliation:

Antoine Lemor
Université de Sherbrooke
Sherbrooke, QC, Canada
E-mail: antoine.lemor@usherbrooke.ca

Shannon Dinan
Université Laval
Québec, QC, Canada

Jérémy Gilbert
Western University
London, ON, Canada

SocArXiv Website https://socopen.org/
SocArXiv Preprints https://osf.io/preprints/socarxiv

SocArXiv Preprint 2026 *Submitted:* 2026-04-12
10.31235/osf.io/6q8yg_v2 *Accepted:*