

CustomNLP4U 2026

**The Second Workshop on Customizable NLP: Progress and
Challenges in Customizing NLP for a Domain, Application,
Group, or Individual**

Proceedings of the Workshop

July 3, 2026

©2026 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-396-8

Introduction

Welcome to the Second Workshop on Customizable NLP (CustomNLP4U), co-located with the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026) in San Diego, California. CustomNLP4U brings together researchers and practitioners exploring how language models can be adapted to the specific needs of individual users, groups or organizations.

Most language modeling research today focuses on building generalist models through pretraining at scale and reinforcement learning-based post-training. The increased capabilities of LLMs have promised greater productivity and innovation, seeing broad consumption for personal and commercial use. However, users' expectations, values, and workflows vary significantly across domains, organizations, cultures, and individuals — factors that are often under-considered in existing pipelines. Generalist models have non-uniform performance, particularly for consumers in sensitive and specialized domains such as law, finance, or health, and for individuals or cultures less represented online. For language models to deliver on their promise, there is a need to develop models that can be tailored to different consumers, easily controlled by them, and able to reason about users' private knowledge and context to provide personalized responses. Alongside these methodological questions, model customization raises ethical and security concerns around copyrighted data, user privacy, and bias — especially in sensitive domains where customization can bring large benefits but also higher risks. To discuss the progress and challenges on these interdisciplinary questions, CustomNLP brings together researchers from diverse fields spanning Natural Language Processing (NLP), Machine Learning (ML), Human-Computer Interaction (HCI), and Fairness and AI Policy.

We received 47 submissions and accepted 22 papers for presentation at the workshop — an acceptance rate of 46.8%. Of these, 17 are archival contributions and appear in this volume; the remaining 5 are non-archival and were presented at the workshop only. Every paper received careful evaluation through a double-blind review process coordinated through OpenReview, with each submission seen by multiple members of our program committee.

We thank the authors for entrusting their work to CustomNLP4U, and we are deeply grateful to our program committee and area chairs for their thoughtful reviews under a tight schedule. We thank the members of our steering committee — Hamed Zamani, Dongyeop Kang, and Yulia Tsvetkov — for their guidance throughout the planning of the workshop. We also thank the ACL 2026 workshop chairs and publication chairs for their support, and the broader *CL community for continuing to make space for focused workshops alongside the main conference.

We hope the program sparks new collaborations and ideas, and that you enjoy the workshop.

The CustomNLP4U 2026 Organizers

Sheshera Mysore, Sachin Kumar, Vidhisha Balachandran, Shirley Anugrah Hayati, Faeze Brahman, Hanane Nour Moussa, and Alireza Salemi

Organizing Committee

Organizers

Sheshera Mysore, Microsoft Office of Applied Research, USA
Sachin Kumar, The Ohio State University and Allen Institute for AI, USA
Vidhisha Balachandran, Microsoft Research, USA
Shirley Anugrah Hayati, University of Minnesota, USA
Faeze Brahman, Allen Institute for AI, USA
Hanane Nour Moussa, The Ohio State University, USA
Alireza Salemi, University of Massachusetts Amherst, USA

Steering Committee

Hamed Zamani, University of Massachusetts Amherst, USA
Dongyeop Kang, University of Minnesota, USA
Yulia Tsvetkov, University of Washington, USA

Program Committee

Reviewers

Shardul Aggarwal, Arnav Arora, Anirudh Atmakuru, Katherine Atwell

Xiaoyan Bai, Ahsaas Bajaj, Nishant Balepur

Isabel Cachola, Yapei Chang, Daiwei Chen, Harshita Chopra

Debarati Das, Jane Arleth dela Cruz

Cristina Garbacea, Sayan Ghosh, Anmol Goel, Taicheng Guo, Yaoqi Guo

Jawad Hossain, Tiancheng Hu

Brihi Joshi

To Eun Kim, Yekyung Kim, Kemal Kirtac

Chen Cecilia Liu, Haokun Liu, Yuhan Liu

Stephen Meisenbacher

Krithika Ramesh, Jenna Russell

Amrita Singh

Yu-Min Tseng

Anvesh Rao Vijjini

Manya Wadhwa, Jiajing Wan

Wentao Zhang, Wangjie Zheng, Karen Zhou, Shengqi Zhu

Table of Contents

<i>BAID: A Benchmark for Bias Assessment of AI Detectors</i> Priyam Basu, Yunfeng Zhang and Vipul Raheja	1
<i>Small Language Models for the Democratization of Financial Literacy: Challenges and Opportunities</i> Tagore Rao Kosireddy, Jeffrey David Wall and Evan Lucas	11
<i>From Understanding to Engagement: Personalized pharmacy Video Clips via Vision Language Models (VLMs)</i> Suyash Mishra, Qiang Li, Anubhav Girdhar and Srikanth Patil	24
<i>Evaluating Customized vs. Generalist Transformer-based Models for Legal Contract Classification</i> Amrita Singh, H. Suhan Karaca, Aditya Joshi, Hye-young Paik and Jiaojiao Jiang	44
<i>Personalizing News Headlines with Retrieval-Augmented Generation</i> Jiajing Wan, Samia Touileb, Lubos Steskal and Lilja Øvrelid	55
<i>Building Multi-turn Intent Classification with LLM-based Labeling</i> Biancen Xie, Kaiqi Bian, Jai Ranjan Singh Gusain, Manikandarajan Ramanathan and Raj Maragoud	68
<i>Cross-Tokenizer LLM Distillation through a Byte-Level Interface</i> Avyav Kumar Singh, Yen-Chen Wu, Alexandru Cioba, Alberto Bernacchia and Davide Buffelli	84
<i>Fine-grained Readability Controlled Summarization of Scientific Documents via Control Vectors</i> Isabel Cachola, Kuleen Sasse and Mark Dredze	97
<i>Building a Custom Taxonomy of AI Skills and Tasks from the Ground Up with Job Postings</i> Stephen Meisenbacher and Peter Norlander	117
<i>Using Topological Data Analysis to Characterize the Layers of Language Models Before and After Word Substitution Attacks</i> Adam Tang, Catherine Liu, Kimberly Lopez, Shreya Subramanian, Leif Zinn-Brooks, Alexia E. Schulz and Adaku Uchendu	131
<i>Customizing ASR for Language Documentation and Resource Prioritization</i> Alexandra Fort and Shobhana Lakshmi Chelliah	149
<i>Improving Medical Hallucination Detection with System Combination and Rule-based Customization</i> Jonathan Lasko, Damianos Karakos and Francis Keith	160
<i>Asking the Right Questions: Can expert-prompted LLMs reformulate legal queries from non-experts?</i> Katherine Atwell, Morgan A. Gray, Jaromir Savelka, Len Rial, Sera Linardi and Malihe Alikhani	167
<i>When Valid Signals Fail: Regime Boundaries Between LLM Features and RL Trading Policies</i> Zhengzhe Yang	182
<i>Unintended Effects of Geographic Conditioning in Large Language Models</i> Naz Col and David M. Chan	191
<i>Efficiency vs. Verifiability in Evidence-Aware RAG: Does Prompt Compression Preserve Citation Grounding?</i> Aiyu Li, Qian Peng and Bin Chen	202
<i>When Gradients Collide: Failure Modes of Multi-Objective Prompt Optimization for LLM Judges</i> Parth Darshan and Abhishek Divekar	216

Program

Friday, July 3, 2026

09:00 - 09:15 *Opening Remarks*

09:15 - 10:45 *Poster Session*

BAID: A Benchmark for Bias Assessment of AI Detectors

Priyam Basu, Yunfeng Zhang and Vipul Raheja

Small Language Models for the Democratization of Financial Literacy: Challenges and Opportunities

Tagore Rao Kosireddy, Jeffrey David Wall and Evan Lucas

From Understanding to Engagement: Personalized pharmacy Video Clips via Vision Language Models (VLMs)

Suyash Mishra, Qiang Li, Anubhav Girdhar and Srikanth Patil

A Versatile Multi-Modal Agent for Rare Disease Diagnosis and Risk Gene Prioritization

Tianyu Liu, Wangjie Zheng, Weihao Xuan, Rui Yang, Botao Yu, Kexin Huang, Nan Liu and Hongyu Zhao

Evaluating Customized vs. Generalist Transformer-based Models for Legal Contract Classification

Amrita Singh, H. Suhan Karaca, Aditya Joshi, Hye-young Paik and Jiaojiao Jiang

Personalizing News Headlines with Retrieval-Augmented Generation

Jiajing Wan, Samia Touileb, Lubos Steskal and Lilja Øvreliid

FD-RAG: Federated Dual-System Retrieval-Augmented Generation

Tianhao Gao, Kai Yang and Yiyang Li

Building Multi-turn Intent Classification with LLM-based Labeling

Biancenc Xie, Kaiqi Bian, Jai Ranjan Singh Gusain, Manikandarajan Ramanathan and Raj Maragoud

Cross-Tokenizer LLM Distillation through a Byte-Level Interface

Avyav Kumar Singh, Yen-Chen Wu, Alexandru Cioba, Alberto Bernacchia and Davide Buffelli

Fine-grained Readability Controlled Summarization of Scientific Documents via Control Vectors

Isabel Cachola, Kuleen Sasse and Mark Dredze

Friday, July 3, 2026 (continued)

Building a Custom Taxonomy of AI Skills and Tasks from the Ground Up with Job Postings

Stephen Meisenbacher and Peter Norlander

10:45 - 11:00 *Coffee Break*

11:00 - 12:15 *Poster Session (cont.)*

Using Topological Data Analysis to Characterize the Layers of Language Models Before and After Word Substitution Attacks

Adam Tang, Catherine Liu, Kimberly Lopez, Shreya Subramanian, Leif Zinn-Brooks, Alexia E. Schulz and Adaku Uchendu

Customizing ASR for Language Documentation and Resource Prioritization

Alexandra Fort and Shobhana Lakshmi Chelliah

Improving Medical Hallucination Detection with System Combination and Rule-based Customization

Jonathan Lasko, Damianos Karakos and Francis Keith

Asking the Right Questions: Can expert-prompted LLMs reformulate legal queries from non-experts?

Katherine Atwell, Morgan A. Gray, Jaromir Savelka, Len Rial, Sera Linardi and Malihe Alikhani

Hint-Assisted Reasoning: Improving Mathematical Problem Solving in Small Language Models

Jawad Hossain, Xiangyu Guo, Jiawei Zhou and Chong Liu

When Valid Signals Fail: Regime Boundaries Between LLM Features and RL Trading Policies

Zhengzhe Yang

Unintended Effects of Geographic Conditioning in Large Language Models

Naz Col and David M. Chan

Efficiency vs. Verifiability in Evidence-Aware RAG: Does Prompt Compression Preserve Citation Grounding?

Aiyu Li, Qian Peng and Bin Chen

Who's Asking? Simulating Role-Based Questions for Conversational AI Evaluation

Navreet Kaur, Hoda Ayad, Hayoung Jung, Shravika Mittal, Munmun De Choudhury and Tanu Mitra

Friday, July 3, 2026 (continued)

When Gradients Collide: Failure Modes of Multi-Objective Prompt Optimization for LLM Judges

Parth Darshan and Abhishek Divekar

Low-resource Authorship Style Transfer via Dynamic Style Inference and Parameter Modulation

Jongkyung Shin, Minguk Jeon, Chan Woo Park and Chiehyeon Lim

12:15 - 12:30 *Closing Remarks*

BAID: A Benchmark for Bias Assessment of AI Detectors

Priyam Basu
Superhuman

priyam.basu@superhuman.com

Yunfeng Zhang
Superhuman

yunfeng.zhang@superhuman.com

Vipul Raheja
Superhuman

vipul.raheja@superhuman.com

Abstract

AI-generated text detectors gain adoption in educational and professional contexts, their fairness remains underexamined. While prior research has uncovered isolated cases of bias, particularly against English Language Learners (ELLs), there is a lack of systematic evaluation of such systems across broader sociolinguistic factors. In this work, we propose a comprehensive evaluation framework for AI detectors across various types of biases. As part of this framework, we introduce a suite of targeted datasets spanning 7 major categories: demographics, age, educational grade level, dialect, formality, political leaning, and topic. Using this, we evaluate four open-source state-of-the-art AI text detectors and find consistent disparities in detection performance, particularly low recall rates for texts from underrepresented groups. Our contributions provide a scalable, transparent approach for auditing AI detectors and emphasize the need for bias-aware evaluation before these tools are deployed for public use.

1 Introduction

As large language models (LLMs) such as GPT-4 (OpenAI, 2024) and LLaMA (Touvron et al., 2023) continue to improve, the line between machine-generated and human-written text is becoming increasingly difficult to draw. These models now produce writing that is not only grammatically correct but also stylistically sophisticated and contextually nuanced (Brown et al., 2020), while being indistinguishable to the amateur eye. Recent advancements have introduced new risks around the generation of deceptive content, raising serious concerns about their potential to mislead or manipulate public perception (Solaiman et al., 2019). These risks span a range of real-world applications, including the automated creation of fabricated news stories (Zellers et al., 2020), fake product reviews (Meng et al., 2025), inauthentic social media posts intended to

influence public opinion (Loth et al., 2024) as well as phishing attacks (Thapa et al., 2025). In parallel, educators have expressed growing unease over the use of generative tools in academic settings (Currie, 2023).

In response, researchers have proposed a variety of detection methods aimed at distinguishing machine-written text from human-written text. These efforts span a range of approaches, from leveraging statistical irregularities in generated outputs (Gehrmann et al., 2019) to training supervised classifiers on curated datasets (Mitchell et al., 2023). Most detectors operate under a binary assumption that a given input is either fully AI-generated or fully human-written, thus evaluating text at a paragraph or document level, while some work focusses on fine-grained detection, including phrase-level or even token-level classification (Teja et al., 2025).

Although significant progress has been made in developing and evaluating AI-generated text detectors, these models have not been tested for fairness and equity. In particular, research on bias in AI detectors remains sparse. (Liang et al., 2023) systematically investigated this issue, where they found that widely-used detectors disproportionately classify texts written by non-native English speakers as AI-generated due to their lower linguistic perplexity. This discovery underscores a troubling consequence: detectors may inadvertently penalize individuals based on their language background, even when their writing is entirely original. Motivated by this insight, our work extends the investigation of bias in AI detectors by evaluating their behavior across a broader and more diverse set of dimensions. Specifically, we examine seven types of bias - demographics, age, educational grade level, dialect, formality, political leaning and topic, to offer a more comprehensive assessment of how detection systems may fail across different groups. By doing so, we aim to highlight not only the technical

limitations of current detectors but also the social implications of deploying them at scale without rigorous fairness evaluations.

2 Related Works

Various methods have been developed for the detection of AI-generated text. Early approaches like (Gehrmann et al., 2019) used statistical cues and visualizations to exploit the fact that AI-generated text often relies on a narrower range of high-probability word patterns. Other methods like (Bao et al., 2024) provide zero-shot way solutions by analyzing outputs via perplexity or entropy differences. ZipPy (Thinkst Applied Research, 2023) foregoes heavy neural networks for speed by using compression ratios to measure textual novelty (an indirect perplexity metric) against a reference corpus of AI-generated text. However, more recent works like GPTZero (Mitchell et al., 2023), Desklib (Desklib, 2025) focus on this as a finetuning task on human vs ai-generated texts. Beyond purely AI- or human-authored texts, researchers have started examining hybrid human-AI texts. (Zeng et al., 2024) explore sentence-level detection in collaborative writing, highlighting that identifying AI-generated segments amid human revisions is extremely challenging. They found that when humans selectively edit or intermix AI-generated sentences, detectors struggle due to rapidly switching authorship and minimal stylistic cues in short segments. This has lead to some more works on sentence-level and phrase-level AI detection (Wang et al., 2023).

Recent benchmarking efforts have focused on systematically evaluating the quality and generalization of AI text detectors across domains, models, and use cases. (Pudasaini et al., 2025) highlight that detectors often fail under distribution shifts, paraphrasing, and newer model generations, underscoring the brittleness of current approaches. (Yu et al., 2025) examine the effectiveness of detectors in academic review scenarios and reveal substantial false-positive risks when evaluating legitimate human writing, especially in specialized or formal domains. (Tao et al., 2024) demonstrated that detection performance varies widely across languages and content genres. (Dugan et al., 2024), showed significant degradation under paraphrased or obfuscated text conditions.

With more widespread use of these detectors, concerns have risen around their reliability and bias. (Liang et al., 2023) in their paper ran an ex-

periment which showed how GPT-detectors were extremely biased against non-native English speakers, incorrectly classifying more than half of the TOEFL human-written essays by English Language Learner (ELL) students as LLM generated. This stems from the underlying low perplexity values (inverse of word sequence probability) of essays written by ELL students. (Chu et al., 2024) talk about how source heuristics such as nationality and content heuristics like linguistics attributes play a significant factor in authenticity detection. They further talk about about content from Asian and Hispanic writers are more likely to be judged as AI users when labeled as domestic students, suggesting interactions between racial stereotypes and AI detection, even when judged by humans. However, they do not run any experiments or show any empirical results to prove this hypothesis – a gap which we explore in this work.

3 Dataset

3.1 Overview

We introduce BAID, a benchmark designed to evaluate the fairness of AI-generated text detectors across diverse demographic and linguistic subgroups. While existing benchmarks primarily assess detector performance under standard or neutral conditions, our benchmark evaluates fairness in conditions where subgroup attributes may influence detector outputs.

To build the dataset, we collect multiple human-written documents spanning seven major bias groups, each containing multiple subgroups. For every human-written document, we generate a corresponding AI-written version using LLMs prompted with carefully crafted instructions that control for a human-like tone. This ensures that both human and AI texts share comparable semantic contexts while differing in authorship. The resulting dataset consists of three fields: `human_written_document`, `AI_generated_document`, and `subgroup_value`.

In contrast to existing datasets that evaluate models under typical or neutral inputs, BAID emphasizes fairness evaluation under bias-revealing conditions. Similar to prior works such as FLEX (Jung et al., 2025), which test language models in extreme fairness scenarios, BAID seeks to expose potential disparities in how AI detectors behave across population subgroups.

The design of BAID follows three guiding prin-

principles - (a) Fairness coverage: The dataset should include a wide range of demographic and linguistic variables that reflect real-world diversity. (b) Semantic control: Human and AI texts should express the same content and intent, ensuring that fairness differences arise from subgroup attributes rather than topical or stylistic drift. (c) Practical evaluation: BAID is intended as a diagnostic tool to evaluate model fairness across a variety of writing domains, from formal essays to conversational text.

3.2 Bias Dimensions

While earlier work (e.g., Stanford HAI (Liang et al., 2023)) focused on the disadvantage faced by English Language Learner (ELL) students, BAID broadens this scope to include demographic, social, and stylistic dimensions. Each bias type represents a fairness-relevant variable where detector disparities could translate into real-world harms.

- **Demographic bias.** We use the ASAP 2.0 dataset (scrosseye, 2020), which contains persuasive essays from standardized writing assessments. Metadata includes author race/ethnicity, gender, socioeconomic status, disability status, and ELL status.
- **Age bias.** Based on the Blog Authorship Corpus (Tatman, 2020), which includes 600K posts from roughly 19,000 bloggers aged 13-48, grouped into four ranges: teens, 20s, 30s, and 40s.
- **Grade-level bias.** The ASAP 2.0 corpus also includes grade-level information ranging from 8 to 12, allowing comparison of writing maturity effects on detector outcomes.
- **Dialect bias.** We examine three English varieties - African American Vernacular English (AAVE) (Blodgett et al., 2016), Singaporean English (Singlish) (Tatman, 2017), and Standard American English (SAE) (Groenwold et al., 2020) to measure robustness across dialectal variation.
- **Formality bias.** We use the GenZ vs. Standard English dataset (Seraaphonano, 2024), which contains 820 paired sentences contrasting informal GenZ phrasing with formal equivalents.

- **Topic bias.** Using the Blog Authorship Corpus, we select ten major topical categories (Arts, Communication/Media, Education, Engineering, Internet, Law, Non-profit, Student, Technology, and Unknown) to test whether detector fairness depends on subject matter.
- **Political ideology bias.** We adopt the dataset from (Baly et al., 2020), which includes articles annotated as left-leaning, neutral, or right-leaning, to evaluate ideological sensitivity.

3.3 Prompt Design

To generate "AI-authored" counterparts, we use a set of structured zero-shot prompts that simulate light-touch human revisions. For essays and articles, the model is prompted to act as an editor, rewriting overly formal or robotic phrases while preserving paragraph structure and meaning. Prompts explicitly discourage stereotypical AI markers such as "in this essay," "delve into," or "in conclusion," and instead promote natural discourse connectors like "so," "but," and "also."

For short-form or conversational inputs (e.g., tweets and messages), prompts are customized to match the linguistic features of each dialect. For example, AAVE samples emphasize authentic syntactic and lexical constructions, while Singlish prompts incorporate pragmatic particles and colloquial phrasing. In all cases, the models are instructed to maintain semantic similarity to the human-written text.

3.4 Generation Process

All AI-generated documents are produced using GPT-4.1 (OpenAI, 2024) and Claude Sonnet 3.7 (Anthropic, 2025). We use a multi-threaded generation pipeline with built-in retry mechanisms to handle rate-limit and timeout errors. Generated outputs are filtered for completeness and cleaned to remove hashtags, emojis and links. Each generation is paired with its corresponding human-written sample and subgroup label.

3.5 Data Quality and Filtering

To ensure quality and reliability, we apply a multi-stage validation process:

- **Automatic filtering:** We discard samples with token repetition and incomplete generations.

Bias Type	Subgroup	Dataset Citation	Count
Demographic	Race/ethnicity:	(scrosseye, 2020)	
	American Indian/Alaskan Native		184
	Asian/Pacific Islander		2988
	Black/African American		3800
	Hispanic/Latino		3192
	Two or more races/Other		2048
	White		4000
	ELL status:		
	Yes		3996
	No		4000
	Student disability status:		
	Yes		3996
	No		3400
	Socioeconomic status:		
Economically disadvantaged	3880		
Not economically disadvantaged	3996		
Gender:			
Female	4000		
Male	3996		
Grade level	Grade 8	(scrosseye, 2020)	1950
	Grade 9		52
	Grade 10		426
	Grade 11		2754
	Grade 12		2638
Age level	Teens	(Tatman, 2020)	4991
	20s		4996
	30s		4996
	40s		4985
Dialect	AAVE	(Blodgett et al., 2016)	19180
	Singlish	(Tatman, 2017)	10000
	SAE	(Groenwold et al., 2020)	8070
Formality bias	GenZ English	(Seraaphonano, 2024)	3280
	Standard English		3280
Topic level	Arts	(Tatman, 2020)	4995
	Communication/Media		4998
	Education		4995
	Engineering		4990
	Internet		4984
	Law		7487
	Non-profit		4995
	Student		4994
	Technology		4998
	Unknown		4154
Political leaning	Left	(Baly et al., 2020)	12800
	Neutral		11010
	Right		13722

Table 1: Dataset composition across bias dimensions, subgroups, and sample counts

- **Semantic alignment:** Using sentence-level embeddings, we compute cosine similarity between human and AI pairs to confirm that the generated text preserves core meaning with a threshold of 0.85.

This process ensures that fairness measurements reflect true subgroup differences rather than artifacts of poor generation quality. The final dataset contains 208166 document pairs distributed across seven bias types and 41 subgroups. Table 1 illustrates the different types of biases, subgroups and count.

4 Detectors

We evaluate two types of detectors: neural models and statistical models. In total, we apply four widely used AI-generated text detectors on the BAID benchmark:

- **Desklib** (Desklib, 2025) - A model developed by fine-tuning a deberta-v3-large (He et al.,

2023) model with adversarial attacks across different domains.

- **E5-small** (MayZhou, 2024) - A lightweight model built using LoRA (Hu et al., 2021) fine-tuning on the E5-small (Wang et al., 2024) encoder model.
- **Radar** (Hu et al., 2023) - A model jointly trained on a detector and a paraphraser task via adversarial learning to improve resilience against LLM-based paraphrasing and cross-model transfer.
- **ZipPy** (Thinkst Applied Research, 2023) - A fast detection model that correlates ease of compression to perplexity as a metric for authorship authenticity.

We run each detector over the entire dataset, individually for human-written and ai-generated text, and compare performance across different biases and subgroups. We use the default decision thresh-

olds for each detector, treating them as black-box systems.

5 Evaluation

Table 3 reports per-subgroup results for each bias. For a fair evaluation, we evaluate the detectors only on the human-written texts, as any observed bias would meaningfully originate from human authorship. In contrast, AI-generated texts merely simulate subgroup characteristics through prompting and do not truly represent the underlying demographic or linguistic identity, making them unsuitable for assessing fairness across real-world subgroups. The metrics we focus on are precision, recall and F1. Three findings are consistent:

Precision. Across all bias dimensions, Desklib demonstrated high precision (0.97-0.99) for demographic, grade-level, and political subgroups, confirming its strong reliability in identifying AI-generated text when present. However, its precision declined on dialectal and informal writing (e.g., 0.44 for Singlish and 0.16 for GenZ). E5 also achieves consistently high precision on demographics and grade levels (0.95-0.99), though its performance drops for dialectal text (0.28-0.50) and some topic categories (0.60-0.75), with notably poor precision for GenZ content (0.04). Radar maintains stable, mid-range precision values (0.55-0.76) across subgroups, suggesting moderate but consistent performance. In contrast, ZipPy records the lowest precision on demographic and grade-level data (0.19-0.31) but achieves moderately better precision (0.49-0.54) on dialect, formality, and topic biases. Overall, the results indicate that neural models such as Desklib and E5 handle precision more consistently, whereas compression-based systems like ZipPy exhibit higher sensitivity to writing style and input length.

Recall. Desklib achieves strong recall on demographic and grade-level dimensions (0.83-0.96), demonstrating robustness to linguistic diversity, though it declines on dialectal and informal text (0.12-0.35). E5 records very low recall for demographic and political groups (0.03-0.45) but substantially higher recall for dialects such as Singlish and Standard English (0.35-0.97 and 0.55, respectively). Radar remains relatively balanced (0.57-0.72) but shows reduced recall on dialectal and informal categories. ZipPy performs worst overall, with recall collapsing on demographics and grade-

level data (0.02-0.55) but reaching extremely high levels (0.95-0.99) for age, dialect, topic, and GenZ subgroups. These results highlight the importance of recall in fairness-sensitive applications: detectors that underperform in recall risk misclassifying human-written text as AI-generated, disproportionately penalizing certain underrepresented groups. Consequently, improving recall robustness across linguistic variation remains an essential goal for fair detection.

F1. Desklib performs the most consistently across all bias dimensions, with high F1 scores ranging from 0.89 to 0.96 on demographic and grade-level subgroups. Its performance decreases, however, on dialectal and informal registers (0.14-0.47). Radar follows with moderate F1 scores (0.60-0.75), though it also experiences noticeable drops on dialect and formality dimensions (0.21-0.33). Despite its high precision, E5 yields relatively low F1 values for demographics (0.30-0.45) and GenZ English (0.04), improving only for Standard English (0.66) and certain topical categories (0.44-0.55). ZipPy exhibits the weakest overall performance, with very low F1 on demographics and grade levels (0.03-0.27), but achieves higher values (0.65-0.70) on dialectal, formality, and topical text. These findings underscore that aggregate F1 averages can mask substantial subgroup disparities, reinforcing the need for disaggregated fairness evaluations.

5.1 Length Sensitivity in ZipPy

ZipPy relies on compression-based heuristics rather than supervised training. The detector "seeds" a compression stream with AI-generated text and then measures how efficiently new samples compress relative to that seed. Texts that share lexical or structural similarity with AI-generated ones achieve higher compression ratios, while human-written texts are typically more variable and compress less efficiently. ZipPy's effectiveness is highly sensitive to input length, shorter texts offer fewer repeating tokens and thus less reliable compression estimates. In future work, we would mitigate this limitation by chunking longer texts and aggregating compression scores across segments to improve stability.

6 Limitations

While we provide a benchmark across a variety of domains, our analysis is limited to a fixed set of

Dimension	Subgroup	Desklib			E5			Radar			ZipPy		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Gender	F	0.98	0.84	0.91	0.98	0.17	0.29	0.60	0.66	0.62	0.24	0.15	0.20
	M	0.99	0.85	0.92	0.99	0.25	0.40	0.61	0.64	0.62	0.23	0.16	0.19
Race	American Indian	0.97	0.65	0.78	0.89	0.17	0.30	0.54	0.61	0.57	0.19	0.13	0.15
	API	0.98	0.82	0.90	0.99	0.19	0.31	0.55	0.64	0.59	0.24	0.15	0.19
	African American	0.98	0.87	0.93	0.98	0.29	0.45	0.63	0.66	0.64	0.28	0.21	0.24
	Hispanic	0.99	0.85	0.91	0.99	0.23	0.37	0.61	0.65	0.63	0.25	0.18	0.21
	White	0.98	0.86	0.92	0.97	0.21	0.34	0.60	0.65	0.62	0.24	0.17	0.20
Two or more	0.99	0.84	0.91	0.95	0.21	0.34	0.60	0.70	0.65	0.26	0.21	0.23	
Economic Disadvantage	Y	0.98	0.84	0.90	0.97	0.27	0.43	0.60	0.65	0.62	0.26	0.20	0.23
	N	0.98	0.83	0.90	0.98	0.20	0.33	0.59	0.65	0.61	0.22	0.16	0.19
Disability Status	Y	0.98	0.81	0.89	0.98	0.19	0.32	0.62	0.65	0.63	0.29	0.26	0.27
	N	0.99	0.84	0.91	0.98	0.37	0.54	0.60	0.66	0.63	0.23	0.15	0.18
ELL Status	Y	0.97	0.77	0.86	0.99	0.20	0.32	0.61	0.63	0.62	0.23	0.17	0.20
	N	0.99	0.85	0.92	0.97	0.24	0.45	0.61	0.65	0.63	0.29	0.23	0.25
Grade Level	8	0.98	0.89	0.93	0.99	0.23	0.38	0.72	0.57	0.63	0.10	0.34	0.05
	9	0.99	0.99	0.99	0.99	0.27	0.43	0.76	0.73	0.75	0.58	0.55	0.56
	10	0.99	0.90	0.94	0.99	0.23	0.37	0.74	0.75	0.74	0.06	0.02	0.04
	11	0.99	0.92	0.95	0.99	0.18	0.30	0.72	0.67	0.70	0.08	0.03	0.04
	12	0.99	0.93	0.96	0.99	0.26	0.41	0.75	0.73	0.74	0.08	0.02	0.03
Age Level	Teens	0.64	0.92	0.76	0.58	0.55	0.57	0.27	0.31	0.29	0.49	0.95	0.65
	20s	0.65	0.88	0.75	0.63	0.39	0.48	0.23	0.22	0.23	0.50	0.97	0.66
	30s	0.68	0.86	0.76	0.65	0.32	0.43	0.28	0.26	0.27	0.50	0.96	0.66
	40s	0.68	0.80	0.74	0.66	0.28	0.39	0.30	0.26	0.28	0.50	0.96	0.66
Dialect Bias	Singlish	0.44	0.26	0.33	0.28	0.35	0.31	0.18	0.25	0.21	0.49	0.98	0.66
	AAVE	0.38	0.20	0.27	0.42	0.71	0.52	0.30	0.52	0.38	0.50	0.98	0.66
	SAE	0.74	0.35	0.47	0.50	0.97	0.66	0.32	0.72	0.44	0.50	0.99	0.67
Formality Bias	GenZ	0.16	0.12	0.14	0.04	0.04	0.04	0.01	0.04	0.02	0.50	0.99	0.67
	Standard	0.52	0.41	0.46	0.72	0.55	0.62	0.38	0.30	0.33	0.54	0.97	0.70
Topic Level	Arts	0.69	0.89	0.78	0.60	0.46	0.52	0.28	0.25	0.26	0.50	0.97	0.65
	Communication/Media	0.73	0.85	0.79	0.63	0.35	0.45	0.29	0.24	0.27	0.50	0.97	0.66
	Education	0.73	0.88	0.80	0.66	0.39	0.49	0.32	0.28	0.30	0.50	0.95	0.66
	Engineering	0.72	0.88	0.79	0.64	0.44	0.52	0.32	0.29	0.30	0.50	0.97	0.66
	Internet	0.68	0.86	0.76	0.62	0.34	0.43	0.29	0.24	0.26	0.50	0.96	0.66
	Law	0.79	0.89	0.83	0.66	0.32	0.43	0.29	0.24	0.26	0.51	0.98	0.67
	Non-profit	0.75	0.88	0.81	0.72	0.44	0.55	0.31	0.22	0.25	0.50	0.97	0.66
	Student	0.68	0.90	0.77	0.59	0.51	0.55	0.28	0.28	0.28	0.49	0.96	0.65
	Technology	0.71	0.86	0.78	0.64	0.38	0.48	0.27	0.23	0.25	0.50	0.97	0.66
	Unknown	0.70	0.87	0.78	0.66	0.41	0.51	0.30	0.26	0.28	0.49	0.96	0.65
Political Ideology	Left leaning	0.98	0.93	0.96	0.73	0.06	0.11	0.51	0.99	0.68	0.45	0.81	0.58
	Neutral	0.99	0.89	0.93	0.57	0.03	0.06	0.51	0.99	0.68	0.46	0.83	0.59
	Right leaning	0.99	0.95	0.97	0.78	0.08	0.14	0.51	0.99	0.68	0.45	0.82	0.58

Table 2: Performance by subgroup across all bias dimensions on human-written texts

only four detectors. Including larger commercial systems or emerging hybrid detectors (e.g. multi-modal or cross-lingual models) could provide additional insight into fairness trends. Another limitation is that even though it spans seven bias dimensions, the focus is on English text, which can be extended to other languages to enable multilingual evaluation. Finally, ZipPy and other statistical detectors are highly sensitive to input length and formatting, while neural detectors may be influenced by pretraining corpora biases. These architectural differences complicate direct comparisons, which call for including multi-language corpora, dynamic threshold calibration, and experiments with hybrid detection models to better understand fairness under broader real-world conditions.

7 Conclusion

In this work, we introduced a benchmark designed to systematically evaluate the fairness of AI-generated text detectors across diverse demographic and linguistic subgroups. We revealed consistent disparities in detection behavior, most notably

recall gaps that disproportionately penalize under-represented writing styles such as dialectal and informal English. While neural detectors like Desklib maintain high overall accuracy, statistical systems such as ZipPy exhibit length-dependent variability, underscoring how detector architecture and input characteristics jointly shape fairness outcomes. Our findings highlight that existing detectors, though effective in aggregate metrics, exhibit bias. This emphasizes the need for bias-aware auditing, training data diversity, and model calibration in AI detection research. We hope BAID will serve as a foundation for developing more equitable and transparent detection systems that perform reliably across different types of population, writing styles, and contexts.

References

- Anthropic. 2025. Claude 3.7 sonnet. <https://www.anthropic.com/news/claude-3-7-sonnet>.
- Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We can detect your bias:

- Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '20.
- Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. [Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature](#). *Preprint*, arXiv:2310.05130.
- Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic Dialectal Variation in Social Media: A Case Study of African-American English. In *Proceedings of EMNLP*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). *Preprint*, arXiv:2005.14165.
- Haoran Chu, Linjuan Rita Men, Sixiao Liu, Shupai Yuan, and Yuan Sun. 2024. [Nationality, race, and ethnicity biases in and consequences of detecting ai-generated self-presentations](#). *Preprint*, arXiv:2412.18647.
- Geoffrey M. Currie. 2023. [Academic integrity and artificial intelligence: is chatgpt hype, hero or heresy?](#) *Seminars in Nuclear Medicine*, 53(5):719–730. Pre-clinical.
- Desklib. 2025. [Desklib ai text detector v1.01](#). <https://huggingface.co/desklib/ai-text-detector-v1.01>.
- Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. [Raid: A shared benchmark for robust evaluation of machine-generated text detectors](#). *Preprint*, arXiv:2405.07940.
- Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. [Gltr: Statistical detection and visualization of generated text](#). *Preprint*, arXiv:1906.04043.
- Sophie Groenwold, Lily Ou, Aesha Parekh, Samhita Honnavalli, Sharon Levy, Diba Mirza, and William Yang Wang. 2020. [Investigating African-American Vernacular English in transformer-based text generation](#). In *Proceedings of EMNLP*.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *Preprint*, arXiv:2111.09543.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. [Radar: Robust ai-text detection via adversarial learning](#). *Preprint*, arXiv:2307.03838.
- Dahyun Jung, Seungyoon Lee, Hyeonseok Moon, Chanjun Park, and Heuseok Lim. 2025. [Flex: A benchmark for evaluating robustness of fairness in large language models](#). *Preprint*, arXiv:2503.19540.
- Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. [Gpt detectors are biased against non-native english writers](#). *Preprint*, arXiv:2304.02819.
- Alexander Loth, Martin Kappes, and Marc-Oliver Pahl. 2024. [Blessing or curse? a survey on the impact of generative ai on fake news](#). *Preprint*, arXiv:2404.03021.
- MayZhou. 2024. [Mayzhou/e5-small lora ai-generated detector](#). <https://huggingface.co/MayZhou/e5-small-lora-ai-generated-detector>. Model card on Hugging Face, accessed October 2025.
- Weiyao Meng, John Harvey, James Goulding, Chris James Carter, Evgeniya Lukinova, Andrew Smith, Paul Frobisher, Mina Forrest, and Georgiana Nica-Avram. 2025. [Large language models as 'hidden persuaders': Fake product reviews are indistinguishable to humans and machines](#). *Preprint*, arXiv:2506.13313.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. [Detectgpt: Zero-shot machine-generated text detection using probability curvature](#). *Preprint*, arXiv:2301.11305.
- OpenAI. 2024. [Gpt-4 technical report](#). <https://openai.com/research/gpt-4>.
- Shushanta Pudasaini, Luis Miralles, David Lillis, and Marisa Llorens Salvador. 2025. [Benchmarking AI text detection: Assessing detectors against new datasets, evasion tactics, and enhanced LLMs](#). In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, pages 68–77, Abu Dhabi, UAE. International Conference on Computational Linguistics.
- scrosseye. 2020. [Asap_2.0: Automated student assessment prize](#). https://github.com/scrosseye/ASAP_2.0.
- Seraaphonano. 2024. [Formal and informal english classification](#). <https://github.com/Seraaphonano>.
- Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps,

- Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. *Release strategies and the social impacts of language models*. *Preprint*, arXiv:1908.09203.
- Zhen Tao, Yanfang Chen, Dinghao Xi, Zhiyu Li, and Wei Xu. 2024. *Towards reliable detection of llm-generated texts: A comprehensive evaluation framework with cudrt*. *Preprint*, arXiv:2406.09056.
- R. Tatman. 2017. The national university of singapore sms corpus. <https://www.kaggle.com/datasets/rtatman/the-national-university-of-singapore-sms-corpus>.
- R. Tatman. 2020. Blog authorship corpus. <https://www.kaggle.com/datasets/rtatman/blog-authorship-corpus>.
- Lekkala Sai Teja, Annepaka Yadagiri, Partha Pakray, Chukhu Chunka, and Mangadoddi Srikar Vardhan. 2025. *Fine-grained detection of ai-generated text using sentence-level segmentation*. *Preprint*, arXiv:2509.17830.
- Jikesh Thapa, Gurrehmat Chahal, Serban Voinea Gabreanu, and Yazan Otoum. 2025. *Phishing detection in the gen-ai era: Quantized llms vs classical models*. *Preprint*, arXiv:2507.07406.
- Thinkst Applied Research. 2023. Zippy: Fast method to classify text as ai or human-generated. <https://github.com/thinkst/zippy>.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. *Llama: Open and efficient foundation language models*. *Preprint*, arXiv:2302.13971.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. *Multilingual e5 text embeddings: A technical report*. *Preprint*, arXiv:2402.05672.
- Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. *Seqxgpt: Sentence-level ai-generated text detection*. *Preprint*, arXiv:2310.08903.
- Sungduk Yu, Man Luo, Avinash Madusu, Vasudev Lal, and Phillip Howard. 2025. *Is your paper being reviewed by an llm? benchmarking ai text detection in peer review*. *Preprint*, arXiv:2502.19614.
- Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2020. *Defending against neural fake news*. *Preprint*, arXiv:1905.12616.
- Zijie Zeng, Shiqi Liu, Lele Sha, Zhuang Li, Kaixun Yang, Sannyuya Liu, Dragan Gašević, and Guanliang Chen. 2024. *Detecting ai-generated sentences in human-ai collaborative hybrid texts: Challenges, strategies, and insights*. *Preprint*, arXiv:2403.03506.

8 Appendix

8.1 Evaluation of AI-Generated Texts

We also ran the same subgroup-level evaluation on the AI-generated samples of the BAID dataset. However, it is important to note that these results do not reflect inherent bias in the same way as the human-written samples. Each AI-generated text was produced using prompts that explicitly instructed the model to rewrite an existing human-written document from the perspective of the original author’s subgroup. As such, any linguistic or stylistic variation in these samples is a result of the generation process rather than genuine demographic or experiential differences. Consequently, the subgroup-level patterns observed in this analysis should be interpreted cautiously, as they reflect prompt conditioning of the language models and quality analysis rather than authentic subgroup bias

Across most dimensions, detectors show noticeably higher recall on AI-generated text than on human-written samples, suggesting that synthetic outputs still carry the statistical fingerprints of machine generation. Among all systems, Desklib stands out for its consistency since it maintains high precision (0.8-0.9) and recall above 0.97 across subgroups, leading to F1 scores well over 0.9. In other words, Desklib reliably identifies generated text regardless of subgroup conditioning. E5 also achieves very high recall but at the expense of precision (0.55-0.60), indicating a tendency to predict AI falsely, an expected trade-off for detectors tuned toward recall. Radar, which incorporates adversarial training and paraphrase modeling, performs more unevenly, with F1 scores typically in the 0.6-0.7 range and weaker results on stylistic dimensions such as dialect and formality. This pattern suggests that Radar’s adversarial robustness does not fully capture stylistic or prompt-induced variation. Finally, ZipPy, the compression-based detector, behaves less predictably. It achieves very high recall (above 0.9) on longer, more regular texts but suffers from low precision (0.3-0.5) and inconsistent F1 scores across subgroups, reflecting its sensitivity to text length and lexical repetition.

Overall, this demonstrates that fairness metrics computed on AI-generated text primarily capture model calibration and sensitivity to surface-level linguistic properties, not representational bias. The uniformly high recall across subgroups confirms that detectors reliably recognize the statistical regularities of generated text, while small subgroup vari-

ations reveal the influence of prompt structure and lexical complexity. These results assess how detectors generalize across controlled synthetic variations, rather than how they behave toward real human diversity.

8.2 Prompt Template

Prompt Template for GenZ Rewriter

SYSTEM_PROMPT_GENZ

You are an AI rewriter that transforms and paraphrases existing text messages into Gen Z style.

Your job is to rewrite a piece of text to sound like it was written in the casual, slang-filled, playful, and internet-savvy style of Gen Z online communication. You will receive input text and your task is to rephrase it into a short, natural-sounding message written in Gen Z tone.

Rules to follow:

- Keep it short and tweet-like.
- Use Gen Z slang, abbreviations, exaggerations, or dramatic flair when natural.
- Don't include hashtags, links, or attribution.
- Do **not** include emojis or punctuation.
- The top priority is to make the rewritten message sound like something a real Gen Z person might post.
- If the text cannot be rewritten (e.g., inappropriate or nonsensical), return "ERROR_404".

Additional instruction: Just output the final text, and nothing else. Do not give pointers or explanations.

Dimension	Subgroup	Desklib			E5			Radar			ZipPy		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Gender	F	0.86	0.99	0.92	0.55	0.99	0.71	0.61	0.55	0.58	0.36	0.47	0.41
	M	0.87	0.99	0.93	0.57	0.99	0.73	0.62	0.59	0.60	0.36	0.47	0.41
Race	American Indian	0.74	0.98	0.84	0.54	0.98	0.70	0.55	0.48	0.51	0.32	0.41	0.36
	API	0.84	0.99	0.91	0.55	0.99	0.71	0.56	0.47	0.51	0.38	0.51	0.44
	African American	0.89	0.99	0.93	0.59	0.99	0.74	0.64	0.61	0.62	0.36	0.45	0.40
	Hispanic	0.87	0.99	0.92	0.56	0.99	0.72	0.62	0.58	0.60	0.37	0.47	0.41
	White	0.87	0.98	0.93	0.56	0.99	0.71	0.61	0.55	0.58	0.36	0.46	0.40
Two or more	0.86	0.99	0.92	0.55	0.99	0.71	0.64	0.53	0.58	0.33	0.40	0.36	
Economic Disadvantage	Y	0.86	0.98	0.92	0.58	0.99	0.73	0.62	0.57	0.59	0.35	0.43	0.39
	N	0.85	0.98	0.91	0.55	0.99	0.71	0.60	0.54	0.57	0.34	0.44	0.39
Disability Status	Y	0.84	0.99	0.91	0.61	0.99	0.76	0.63	0.59	0.61	0.34	0.39	0.36
	N	0.86	0.99	0.92	0.55	0.99	0.71	0.63	0.57	0.60	0.37	0.50	0.43
ELL Status	Y	0.82	0.99	0.90	0.59	0.99	0.74	0.62	0.61	0.61	0.36	0.43	0.39
	N	0.87	0.99	0.93	0.55	0.99	0.71	0.62	0.59	0.60	0.35	0.45	0.39
Grade Level	8	0.90	0.98	0.94	0.57	0.99	0.72	0.64	0.78	0.71	0.41	0.68	0.51
	9	0.99	0.99	0.99	0.58	0.99	0.73	0.74	0.77	0.75	0.42	0.73	0.54
	10	0.91	0.99	0.95	0.57	0.99	0.72	0.74	0.74	0.74	0.39	0.63	0.48
	11	0.93	0.99	0.96	0.55	0.99	0.71	0.69	0.74	0.72	0.42	0.69	0.52
	12	0.93	0.99	0.96	0.57	0.99	0.73	0.74	0.76	0.75	0.42	0.72	0.53
Age Level	Teens	0.86	0.49	0.62	0.57	0.60	0.59	0.19	0.17	0.18	0.21	0.01	0.02
	20s	0.82	0.53	0.64	0.56	0.77	0.65	0.24	0.25	0.24	0.36	0.01	0.03
	30s	0.81	0.60	0.69	0.55	0.83	0.66	0.30	0.32	0.31	0.42	0.03	0.05
	40s	0.76	0.62	0.68	0.54	0.86	0.67	0.35	0.40	0.38	0.57	0.05	0.09
Dialect Bias	Singlish	0.48	0.67	0.56	0.12	0.09	0.10	0.50	0.99	0.67	0.12	0.00	0.01
	AAVE	0.45	0.66	0.54	0.43	0.02	0.04	0.50	0.99	0.67	0.19	0.00	0.01
	SAE	0.57	0.88	0.69	0.01	0.00	0.00	0.50	0.99	0.67	0.05	0.00	0.00
Formality Bias	GenZ	0.31	0.40	0.35	0.00	0.00	0.00	0.50	0.99	0.67	0.00	0.00	0.00
	Standard	0.50	0.98	0.66	0.50	0.99	0.67	0.50	0.99	0.67	0.86	0.17	0.29
Topic Level	Arts	0.85	0.60	0.70	0.56	0.70	0.62	0.33	0.37	0.35	0.34	0.02	0.03
	Communication/Media	0.82	0.69	0.75	0.55	0.79	0.65	0.35	0.41	0.38	0.51	0.04	0.07
	Education	0.85	0.68	0.76	0.57	0.80	0.66	0.37	0.42	0.39	0.52	0.05	0.09
	Engineering	0.85	0.66	0.74	0.57	0.75	0.65	0.35	0.39	0.37	0.62	0.05	0.09
	Internet	0.81	0.61	0.69	0.54	0.79	0.64	0.35	0.41	0.38	0.57	0.04	0.08
	Law	0.87	0.76	0.81	0.55	0.84	0.67	0.35	0.40	0.37	0.74	0.07	0.13
	Non-profit	0.86	0.70	0.77	0.60	0.83	0.69	0.39	0.51	0.44	0.60	0.05	0.09
	Student	0.85	0.57	0.68	0.57	0.65	0.61	0.28	0.28	0.28	0.31	0.02	0.03
	Technology	0.82	0.65	0.73	0.56	0.78	0.65	0.33	0.38	0.36	0.51	0.03	0.06
Unknown	0.83	0.63	0.72	0.57	0.79	0.67	0.34	0.39	0.37	0.30	0.02	0.03	
Political Ideology	Left leaning	0.94	0.98	0.96	0.51	0.98	0.67	0.95	0.05	0.09	0.03	0.01	0.01
	Neutral	0.90	0.99	0.94	0.50	0.98	0.66	0.96	0.05	0.09	0.05	0.01	0.02
	Right leaning	0.95	0.99	0.97	0.51	0.98	0.67	0.93	0.04	0.08	0.04	0.01	0.01

Table 3: Performance by subgroup across all bias dimensions on AI-generated texts

Small Language Models for the Democratization of Financial Literacy: A Case Study in Fine-Tuning on Open Financial QA Data

Tagore Rao Kosireddy and Jeffrey Wall and Evan Lucas
Michigan Technological University / 1400 Townsend Drive
Houghton, Michigan, United States of America
trkosire, jdwall, eglucas, @mtu.edu

Abstract

This study seeks to test whether low-cost inference and efficient *Small Language Models* (SLMs) fine-tuned on existing open-source question answering datasets are capable of creating financial literacy chat bots that can answer financial questions for those with limited financial knowledge. The use of SLMs is growing in popularity across many domains, but SLMs are not thoroughly explored in the finance sector. This study offers an exploration of challenges and opportunities that exist in the finance sector to utilize SLMs for open-source financial question answering applications. In particular, this study examines the outputs of several open-source SLMs fine-tuned on the open-source FinGPT FiQA_QA financial question answering dataset. We fine-tuned two versions of each model, one with an instruction prompt and one without an instruction prompt and compared the model outputs with ground truth human responses from the dataset. Further qualitative rating and analysis are provided for model outputs and the dataset. The exploration highlighted challenges with available open data and the fine-tuned SLMs. Existing open data sets in the financial AI research community are not sufficient to produce high-quality outputs with SLMs. Successful fine-tuning of SLMs has occurred in other domains with high quality data sets. We thus issue a call for new and better open financial question answering datasets that could result in higher-quality small language models.

1 Introduction

Recent advances in *natural language processing* (NLP) show that *language models* (LMs) produce impressive results on various general-domain NLP tasks (Li et al., 2023a). A language model is a statistical model that predicts the likelihood of missing words given the context of sequences of words, based on the conditional probabilities observed in some large corpora of training text (Bengio et al.,

2000).

Specialized LMs have been tuned for a number of targeted domains, such as in health and finance (Ranjit et al., 2024; Araci, 2019; Fu et al., 2024). This study explores the application of LMs to the finance domain. Individuals must learn to successfully navigate financial systems to improve their socio-economic status. However, many in society lack access to quality financial tools and knowledge to improve their economic situation. Advancements in *financial technology* (FinTech) have facilitated further democratization of financial markets, such as providing free trades to *retail investors* (i.e., individual investors) through online and mobile brokerage applications. FinTech and social trends have prompted a surge in retail investor engagement in financial markets (Fisch, 2022).

Although access to markets is further democratized with FinTech, access to financial knowledge has not kept pace. Low levels of financial literacy are well documented across the general population, and financial illiteracy is associated with poorer financial decision making (Lusardi and Mitchell, 2014). This gap matters in the FinTech era: the recent surge in retail investor activity, combined with limited financial knowledge among some investors, has led to risky investing behavior. Some retail investors treat investing as gambling (Gao and Lin, 2015). Others rely on social influencers to make financial decisions or simply follow social trends in fear of missing out (Dalimunthe et al., 2023; Pedersen, 2022). Retail investors, particularly those who have limited access to financial knowledge or means, deserve better open, low-cost, and privacy preserving resources to develop financial literacy through FinTech applications. Open-source and low-cost LMs trained to answer financial questions could support financial literacy in a privacy preserving manner. We use *privacy preserving* in the practical sense of local, on-device inference: be-

cause a sufficiently small model can run on a user’s own consumer-grade hardware, financial questions need not be sent to a third-party API or provider. We do not claim a formal guarantee such as differential privacy.

In this study we ask: are existing open-source financial question answering datasets and low-cost, efficient, and privacy preserving LMs capable of generating financial tools to answer retail investors’ financial questions? To answer this question, we identify open-source financial question answering datasets and state-of-the-art LMs that can run at low-cost on consumer-grade technology to preserve privacy. Open-source datasets and smaller LMs could be the key to democratizing access to advanced financial tools for those in society who most lack financial knowledge and means.

We define *small language models* (SLMs) as LMs with three billion parameters or fewer, since such models are the most likely to run on consumer-grade hardware.

2 Related Work

This section highlights important existing work related to the use of LMs for financial applications, financial question answering datasets, and SLMs.

2.1 Financial Language Models

Many proprietary and open-source financial LMs exist. For example, Bloomberg developed an LM trained on their vast library of financial data, BloombergGPT (Wu et al., 2023). In response, researchers have developed open-source LMs with similar capabilities to democratize access to financial LMs.

The FinGPT project consists of several open-source financial datasets and fine-tuned LMs, such as a Llama-7B and Falcon-7B (ai4finance.org). The models provide classification outputs, such as sentiment analysis and financial named entity recognition (Yang et al., 2023a). Currently, FinGPT does not support question answering capability that could facilitate the development of financial literacy. Similarly, FinBERT is a fine-tuned LM that supports several classification tasks, such as sentiment analysis. FinBERT was tuned on a variety of open data (Araci, 2019). Another model, FinMA was tuned for similar classification tasks. Like FinGPT, FinMA is a Llama-based LM with 7B and 30B versions (Xie et al., 2023). Despite

the open nature of these models that could support democratized access to financial LMs, their question answering capabilities are limited to highly specific questions with often single value answers (e.g., yes/no). These models are not ideal for supporting the development of financial literacy, as they assume users already possess a certain level of financial literacy.

Although low-cost and privacy-preserving LMs, namely SLMs, have demonstrated state-of-the-art results on general tasks (Mehta et al., 2024; Abidin et al., 2024; Team et al., 2024) and several domain-specific tasks (Ranjit et al., 2024; Fu et al., 2024; Sen et al., 2024) when fine-tuned on quality data sets, our work shows that SLM quality is not enough to achieve quality outputs with available financial question answering datasets. We show that domains like finance need better open-source datasets to democratize access to financial LMs. The financial open-source community does not yet possess the datasets necessary to train high-quality LMs that can democratize access to financial literacy tools. High-quality, open datasets are crucial to create efficient and quality large and small LMs (Gunasekar et al., 2023; Li et al., 2023b).

2.2 Financial Question Answering Datasets

Several open datasets support finance-related tasks such as named entity recognition, sentiment analysis, and headline classification, but few target long-form financial question answering. Table 1 summarizes the question answering datasets most relevant to this study.

The FiQA datasets, scraped from social media sites including StackExchange, Reddit, and StockTwits, are among the most widely used and were published for an open research challenge in 2018 (Maia et al.); the original challenge link now returns a 404 error. Of the two FiQA subsets, the sentiment dataset (FiQA_SA) is a classification task and is not relevant here, while the textual opinion dataset (FiQA_QA) is. A version of FiQA_QA is available through the FinGPT HuggingFace page and consists of more than 17,000 rows, each with a question prompt, a human response, and an instruction prompt.

Other long-form datasets exist but are not openly usable. The FinMA team’s dataset (Huang et al.) provides primarily short-form answers (e.g., yes/no or numeric values) that are unsuitable for support-

ing financial literacy. FinTextQA (Chen et al., 2024) and the InvestLM dataset (Yang et al., 2023b) are both drawn from high-quality sources such as textbooks and government or academic material and would suit our objectives, but neither is linked in its respective paper or repository (AbciNLP), plausibly because textbook-derived data raises copyright concerns. High-quality financial corpora therefore exist, including proprietary resources such as Bloomberg’s, but are not shared openly.

Given the limited number of openly available long-form financial question answering datasets, we selected the FinGPT FiQA_QA dataset for fine-tuning. Truly democratizing access to financial knowledge will require open LMs together with better open financial question answering datasets.

Table 1: Financial question answering datasets most relevant to this study.

Dataset	Primary source	Long-form	Open
FiQA_QA	Social media	Yes	Yes
FinMA QA	Social media / mixed	No	Yes
FinTextQA	Textbooks, gov.	Yes	No
InvestLM	Textbooks, academic	Yes	No

2.3 Small Language Models

SLMs offer a meaningful opportunity to democratize access to low-cost, efficient LMs that can run on consumer-grade hardware. As defined above, we treat SLMs as transformer-based LMs (Vaswani et al., 2017) with three billion parameters or fewer. Several such models are openly available and have shown strong results despite their size, including Google’s Gemma-2B (Team et al., 2024), Microsoft’s Phi series (Abdin et al., 2024), Apple’s OpenELM family (270M to 3B) (Mehta et al., 2024), and TinyLlama (Zhang et al., 2024). We fine-tune and evaluate models from these families in this study.

Fine-tuning adapts these models to a domain at far lower cost than training from scratch (Wei et al., 2021), and fine-tuned SLMs have performed well across domains such as meeting summarization (Fu et al., 2024) and hate speech detection (Sen et al., 2024). For example, Ranjit et al. (2024) fine-tuned an SLM on radiology-specific data and obtained quality outputs with fewer hallucinations than the base model, and Li et al. (2023b) showed that the Phi-1.5 SLM, when trained on high-quality data, matched or exceeded much larger models on rea-

soning tasks, underscoring that data quality matters as much as scale. To our knowledge, no prior study has explored fine-tuning SLMs for financial question answering.

3 Method

This research study consisted of selecting a dataset and SLM for fine-tuning, and assessing the outputs of the model.

3.1 Data Selection and Preparation

As outlined in Section 2.2, the FinGPT FiQA_QA dataset on HuggingFace was the only open financial question answering dataset we identified that met the criteria for this study, and we therefore selected it for fine-tuning. We randomly partitioned the 17,110 data points into a fine-tuning set (13,688 samples) and a test set (3,422 samples). The human responses in the dataset have a mean length of 258.2 tokens and a median of 195.0 tokens, indicating a positively skewed length distribution. This distribution informed our generation settings: we capped generation at 250 new tokens (Section 3.4), a value close to the mean human response length, so that model outputs were comparable in length to the ground-truth answers while remaining bounded for the majority of examples.

3.2 Small Language Model Selection

Prior work has begun to examine whether SLMs are suitable for financial question answering. Kosireddy et al. (2024) studied several base SLMs without fine-tuning to assess whether they possessed qualities appropriate for the task. They examined Gemma-2B, Phi-2, OpenELM-270M, OpenELM-540M, OpenELM-1B, OpenELM-3B, and TinyLlama for characteristics that would make the models suitable for democratizing LMs, and found that several were promising and warranted further fine-tuning. We build on their findings by fine-tuning these models to determine whether fine-tuning further improves their ability to answer financial questions. We also include ChatGPT-4o as a benchmark, since it is a widely used LLM that some individuals have turned to for financial guidance.

3.3 Model Fine-tuning

As outlined in Section 2.2, each FinGPT FiQA_QA example contains an input question, an output containing the human response, and an instruction prompt. The instruction prompt is not authored

by us: the dataset supplies one of five fixed instruction paraphrases per example (for example, “Utilize your financial knowledge, give your answer or opinion to the input question or subject. Answer format is not limited.”). All five instructions are listed in Appendix A. To determine the benefit of including an instruction prompt when fine-tuning SLMs, we fine-tuned each model twice, once with the dataset-provided instruction prompt and once without it. Instruction fine-tuning has been used in multiple studies to improve model outputs (Mishra et al., 2022; Sanh et al., 2022; Ouyang et al., 2022; Marvin et al., 2023; Wei et al., 2022).

The fine-tuning of each model was completed with the SFTTrainer class from the HuggingFace transformers library on a NVIDIA RTX 3090 or A6000 GPU for 2 epochs. The AdamW optimizer (Loshchilov and Hutter, 2017) was utilized with bf16 mixed precision (Burgess et al., 2019). The models were evaluated every 250 steps during fine-tuning and the best model was saved based on evaluation loss.

The initial learning rate was set to $5e-5$ with a weight decay of 0.01 to 0 using a linear learning rate schedule with warmup steps set to 500, train batch and eval batch sizes set to 2, and gradient accumulation steps set to 1 with gradient clipping at 1.0. These hyperparameters, including the train and evaluation batch sizes and the gradient accumulation steps, were held identical across both the RTX 3090 and A6000 GPUs, so the effective batch size was the same for every run regardless of which GPU was used.

3.4 Model Output Evaluation

After fine-tuning the models, we generated outputs from the various fine-tuned and their base models to evaluate the responses. Hereafter, we refer to the fine-tuned models with FT and the base zero-shot models as ZS. The models with an instruction prompt are identified with an I, such as FT+I or ZS+I. Outputs for each model were generated from the test dataset using the same generation parameters for all models (`max_new_tokens = 250`, `top_k = 30`, `top_p = 0.8`, `no_repeat_ngram_size = 5`, `num_return_sequences = 1`).

We evaluated the model outputs in several ways. First, we compared each model’s outputs to the human responses from the dataset using quantitative metrics. We treated the human answers as

ground truth for similarity comparison metrics. We calculated ROUGE metrics, which are commonly used for summarizing and evaluating the degree of overlap between words, bi-grams, or common substrings between a candidate and reference sentence (or sentences) (Lin, 2004). The ROUGE metrics range between 0 and 1, with higher scores indicating higher similarity between the produced and the reference text. ROUGE scores do not take semantic meaning into account, and have been criticized for this shortcoming (Akter et al., 2022; Schluter, 2017). We also calculated BERTScore, which measures the similarity between a candidate and a reference using contextual embeddings (Zhang et al., 2019), with scores that in principle range from -1 to 1. We used the default English configuration of the bert_score library, which uses roberta-large with the library’s default layer selection and without baseline rescaling. Because raw, non-rescaled BERTScore for fluent English text occupies a narrow band near the top of its range, the absolute scores cluster between roughly 0.80 and 0.84 across all models; the small variation we observe is therefore expected behavior of the metric under this configuration rather than evidence that the outputs are near-identical. BERTScore is also less sensitive to small errors when the candidate text is lexically or stylistically similar to the reference text (Hanna and Bojar, 2021). Further, we calculated the Semantic Textual Similarity (STS), which produces similarity scores from 0 to 1 between a candidate and reference sentence (or sentences) by using a Cross-Encoder which achieves better performance than a Bi-Encoder (Reimers and Gurevych, 2019; Risch et al., 2021).

We also calculated a Flesch readability score (Flesch, 2007) for the outputs of each model. The Flesch readability score assesses the complexity of text on a scale from 0 to 100, with higher values representing text that is more complex and difficult to read. Further, we calculated a diversity score (four distinct n-grams) (Li et al., 2015), which assess the lexical diversity in model outputs on a scale from 0 to 100 with higher values representing greater lexical diversity in model responses.

Second, three domain experts, consisting of the business researcher co-authors, rated the outputs of each model. These individuals include an Accounting researcher, a Finance researcher, and an Information Systems researcher. Since the FiQA_QA dataset is founded on social media data that con-

tains personal opinion, the outputs were not rated for factual accuracy. Instead, the raters evaluated the outputs on whether the output answered the question or not without regard for accuracy.

Following best practice in multiple-rater situations, the three business researchers met to identify and reach consensus on a set of rating behaviors to enhance rating reliability (Sattler et al., 2015). The evaluation proceeded in three stages: a small calibration round to establish shared rating behaviors, a larger reliability round on the outputs of a single model to test whether those behaviors produced consistent ratings at scale, and a final cross-model round on a carefully screened set of questions for the substantive comparison across models. In the calibration round, the authors randomly selected 10 questions, which the three business researchers rated individually. Then, in a video conference, the three researchers shared their ratings and explained why they gave each rating. Where rating discrepancies existed, the researchers discussed the differences and arrived at a set of common rating behaviors for assessing the relevance of the model outputs.

The business researchers then conducted a larger reliability round of 100 outputs from a single model, OpenELM-270M, rated on a 5-point Likert scale (1 = irrelevant to 5 = relevant). Holding the model fixed for this round was deliberate, so that any disagreement reflected the rating protocol rather than differences in model quality. We report inter-rater reliability as pairwise scores, using the Pearson Correlation Coefficient and Cohen’s Kappa, rather than a single omnibus statistic such as Fleiss’ or Krippendorff’s alpha. With only three raters, the pairwise breakdown is diagnostic: it shows that raters 1 and 2 agreed reasonably well while agreement involving rater 3 was substantially lower, a pattern that a single aggregate coefficient would obscure. While one correlation was above 0.8 for raters 1 and 2, the other correlation scores were less than 0.55. Similarly, the highest Kappa was close to 0.65 for raters 1 and 2, but lower than 0.32 for the other rater comparisons. The authors met and discussed the results. The raters noted that many of the responses had expansive hallucinated narratives around the core answer or lack thereof, which made it difficult to rate the relevance. Rater 1 had a mean rating of 3.22 ± 1.0971 . Rater 2 had a mean rating of 3.22 ± 1.2519 . Rater 3 had a mean rating of 3.89 ± 1.4695 . Based on the concerns over hallu-

ination and the means of two of the raters scoring close to the center of the 5-point scale, the authors determined to simplify the scale to a 3-point Likert scale from irrelevant to relevant for the final round. This was done to simplify the issue of interpreting the relevance of hallucinated narratives and to push ratings away from the center of the scale.

For the final domain expert evaluation, the raters reviewed a sample of the same 10 questions across all of the different models. The questions were more carefully screened for the final round than for the random selection used in the 100-output reliability round. This was done because many of the entries in the dataset were not actually questions. In Appendix B.0.1 examples of this issue are provided with a calculation of the number of potentially problematic questions. The 10 questions for the final evaluation were hand selected from a random sample of 100 questions by one of the domain experts, who selected the first 10 valid questions from that sample. Here, a *valid* question is an entry that is an actual, well-formed question rather than a statement or malformed prompt; as quantified in Appendix B.0.1, a substantial fraction of dataset entries do not meet this bar.

Third, after reviewing the quantitative and qualitative results, one of the business researchers performed a post-hoc qualitative review of the question prompts, the human outputs, and the model outputs on a small random subset of the data to identify patterns that could explain the seemingly odd quantitative results.

4 Results

The quantitative and qualitative results are provided in the following sub-sections.

4.1 Quantitative Output Comparisons

When comparing the model outputs to the human responses designated as the ground truth for models with an instruction prompt, OpenELM-3B FT performed the best on the ROUGE-1 (0.2458), ROUGE-2 (0.0373) and ROUGE-L (0.1340). In comparison, ChatGPT-4o scored slightly higher for ROUGE-1 and ROUGE-2, but lower for ROUGE-L and BERTScore. The TinyLlama-1.1B ZS model performed the best on the BERTScore (0.8334), which for comparison was higher than the ChatGPT-4o score. Phi-2 ZS performed the best on the STS metric (0.4454). For comparison,

ChatGPT-4o scored well above all models on STS. Table 2 shows further comparison details. In most cases, the metrics are not substantially different across the models. As depicted in the table, many of the base models (i.e., ZS models) performed better than the fine-tuned models. The implications of these findings are discussed later.

To understand the impact of the fine tuning and instruction prompt, we perform a short ablation study and present the results in Table 3. We compare the automatic metrics for Phi 1.5 with (notated by +) and without (notated by -) fine-tuning and the instruction prompt. We find that the instruction prompt has only a small impact and that the fine-tuning has a larger positive impact.

In addition to similarity comparisons, we also compared the readability of each model using the Flesch readability measure (Flesch, 2007) and the lexical diversity of the text with a diversity score (Li et al., 2015) and report the results in Table 5 and 4. The results suggest that the OpenELM-450M FT with instruction prompt (80.98), OpenELM-270M FT without instruction prompt (79.10), and OpenELM-270M FT with instruction prompt (78.50) performed the best for readability, even better than the readability of the human output (63.68). The top three models with the greatest lexical diverse were Phi-2B FT with instruction prompt (72.58), Phi-2B FT without instruction prompt (72.02) and Gemma-2B FT with instruction prompt (71.79), which are slightly lower than the human output (78.38). Many models showed an increase in diversity after fine-tuning. Fine-tuning also seemed to increase readability for models, likely because the social media fine-tuning data was simpler than the original training data for models, which included scholarly articles in many cases.

4.2 Qualitative Ratings

Despite following best practices to establish consistency in rating (as outlined in Section 3.4), interrater agreement was low. Averaged across models, the pairwise Cohen’s Kappa values were 0.1213, 0.2236, and 0.1506 (Table 6), which corresponds to only slight-to-fair agreement on the Landis and Koch scale. Per-model pairwise Kappa ranged widely, from -0.273 (below chance) to 0.667 (substantial), and the Pearson correlation scores ranged from -0.333 to 0.843. In other words, the raters agreed well on a minority of models but barely

above chance on most, and the aggregate signal is weak agreement rather than moderate agreement.

We do not treat this low agreement as solely a shortcoming of the rating protocol. Three factors specific to the data make consistent relevance judgments difficult, even for trained domain experts. First, many question prompts are truncated, missing context, grammatically incorrect, or otherwise ill-formed (Sections 4.3.1 and 4.3.2), so there is often no single defensible notion of what a relevant answer would be. Second, many outputs embed an answer or non-answer inside an expansive hallucinated narrative that mimics a social media responder, which leaves genuine room for disagreement over whether the question was actually answered. Third, because the outputs were rated for whether they answered the question rather than for factual accuracy, the judgment is inherently subjective when the output rambles. The implication is that when domain experts cannot reliably agree on the relevance of outputs derived from this data, the difficulty is located largely in the question and answer pairs themselves, not only in the raters. This observation is what motivated the post-hoc qualitative analysis in Section 4.3.

Although lower than desired, the scores are not surprising to the raters. Several of the question prompts in the FinGPT FiQA_QA dataset seem truncated, are missing context, are grammatically incorrect, and/or make little sense. Further, many of the fine-tuned models suffered from extreme hallucinations in the form of narratives that mimicked a social media responder. These narratives, with answers or non-answers scattered throughout the text, made it difficult to assess whether the output answered the question or not. The model with the highest rating was the Phi-1.5 ZS model with an average rating of 2.60/3.00 as depicted in Table 2. Table 3 shows the quantitative assessments for all models run using Phi-1.5. Although far from ideal, the results show promise for future work and the development of higher quality datasets.

4.3 Post-hoc Qualitative Analysis

This study sought to answer the question: are existing open-source financial question answering datasets and low-cost, efficient, and privacy preserving LMs capable of generating reasonable responses to answer individuals’ financial questions? Based on the results of this study, the answer appears to be, not yet with existing datasets. In some

Table 2: Evaluation Metrics for all models, including an instruction prompt. ZS = Zero Shot, FT = Fine Tuned ChatGPT-4o is included for comparison purposes.

Model Name	Hum. Eval	ChatGPT-4o	R-1	R-2	R-L	BertScore	STS
TinyLlama-1.1B FT	2.37	1.20	0.2229	0.0330	0.1246	0.8331	0.4267
TinyLlama-1.1B ZS	2.07	1.20	0.2243	0.0334	0.1255	0.8334	0.4280
Phi-1 FT	1.40	1.00	0.2356	0.0348	0.1316	0.8262	0.3710
Phi-1 ZS	1.90	1.00	0.2384	0.0358	0.1320	0.8265	0.3752
Phi-1.5 FT	2.27	1.60	0.2395	0.0368	0.1316	0.8324	0.4384
Phi-1.5 ZS	2.60	1.60	0.2209	0.0209	0.1132	0.8057	0.2460
Phi-2 FT	2.13	1.40	0.2236	0.0339	0.1249	0.8317	0.4307
Phi-2 ZS	2.23	1.40	0.2366	0.0371	0.1296	0.8324	0.4454
Gemma-2B FT	2.43	1.60	0.1941	0.0286	0.1147	0.8303	0.4073
Gemma-2B ZS	-	-	0.1977	0.0294	0.1157	0.8304	0.4132
OpenELM-1.1B FT	2.27	1.20	0.2348	0.0351	0.1281	0.8319	0.4285
OpenELM-1.1B ZS	2.03	1.00	0.2345	0.0356	0.1283	0.8323	0.4298
OpenELM-3B FT	2.30	1.10	0.2458	0.0373	0.1340	0.8285	0.4061
OpenELM-3B ZS	1.67	1.10	0.2433	0.0351	0.1327	0.8227	0.3472
ChatGPT-4o	-	-	0.2723	0.0420	0.1310	0.8205	0.5591

Table 3: Ablation study for Best Performing Model (Phi-1.5) ZS = Zero Shot, FT = Fine Tuned, I = Instruction Prompt Included

	R-1	R-2	R-L	BS	STS
FT+I	0.240	0.037	0.132	0.832	0.438
ZS+I	0.221	0.021	0.113	0.806	0.246
FT-I	0.242	0.037	0.132	0.832	0.440
ZS-I	0.239	0.028	0.120	0.811	0.342

Table 4: Readability scores for the top three models and human written ground truth answers

Model	Readability
OpenELM-450M FT+I	80.98
OpenELM-270M FT	79.10
OpenELM-270M FT+I	78.50
Ground Truth Answers	63.68

Table 5: Diversity scores for the top three models and human written ground truth answers

Model	Diversity
Phi-2B FT+I	72.58
Phi-2B FT	72.02
Gemma-2B FT+I	71.79
Ground Truth Answers	78.38

cases, fine-tuning harmed model performance. In other cases it helped performance. The poor quality of the only openly available dataset made both the qualitative assessment of the results difficult

Table 6: Average inter-rater reliability metrics between three expert human raters - Rater 1, Rater 2, Rater 3 across all models

Metric	1 vs 2	1 vs 3	2 vs 3
Avg. Pearson Corr.	0.3175	0.3927	0.4321
Avg. Cohen's Kappa	0.1213	0.2236	0.1506

and the quantitative assessment inconclusive. We treat this difficulty as an informative result in its own right: the inability to draw firm conclusions from this dataset localizes the current bottleneck for SLM-based financial question answering in the available open data rather than in model capacity, and it directly motivates the call for better datasets that follows.

After reviewing the quantitative scores and the qualitative rating results, the authors wanted to better understand why the fine-tuned models didn't perform. One of the business researchers on the team reviewed a small, random subset of the training data and the model results to identify re-occurring patterns. The following are key themes that emerged after analyzing the data further.

4.3.1 Unclear and Cryptic "Questions" in FiQA_QA Dataset

Because the data was scraped from social media sites, it contains a variety of odd, unclear, and grammatically incorrect questions. This created difficulties for the models and human raters. Appendix B.0.1 provides some examples from the dataset.

These types of question prompts provided room for too many possible answers, or the inability to meaningfully provide an answer. The human responses in the data set to question prompts with these issues represent either one of many possible interpretations of the question or unrelated answers to an uninterpretable question. Training data points with this issue provide too much ambiguity to produce a quality model.

4.3.2 Missing Context in FiQA_QA Questions

Second, the FiQA_QA dataset has many questions that seem to be truncated or missing important context. This became evident after reviewing several question/human answer pairs. The human answers seemed to contain important context related to the original question that was missing in the question prompt. This could have occurred if the actual question was truncated during scraping or data cleaning, or if the question was clarified in the discussion thread and therefore not scraped fully. Either way, this issue raises a data quality concern. Appendix B.0.2 provides examples of this data problem.

The missing context in some questions poses a problem when pairing the vague, de-contextualized question prompts to the contextually rich answers. Models, particularly SLMs, trained on too much of this low quality data cannot be expected to perform well.

4.3.3 Unanswered Questions with Hallucinated Context

In some cases, a model did not answer a question, but mimicked the narrative nature of the social media fine-tuning data. It is possible that the hallucinated context in these outputs caused overestimated similarity scores by mimicking the context, but not the answer. These narrative hallucinations also caused issues with qualitative ratings. Appendix B.0.3 provides examples of results with these types of hallucinations.

4.3.4 Hallucinated Context in Relevant Answers

Some models produced relevant answers to questions, but did so through narratives with hallucinated contextual details. Appendix B.0.4 provides examples of some of these hallucinations.

5 Conclusions

Based on the qualitative findings of this study, quality financial question answering datasets will need:

- 1) clearer questions with less ambiguity that contain the full context of the question,
- 2) more professional, yet readable answers that remove unnecessary personal information to avoid contextual hallucinations,
- 3) good instruction prompts to guide the behavior of the model toward answering in a professional manner and to further avoid hallucinations, and
- 4) models and dataset responses that re-direct uniformed questions toward better questions, as many of the questions asked in the social media dataset exhibited the difficulty many retail investors have in forming appropriate questions.

These findings demonstrate the need for better, open financial question answering datasets. We encourage more faculty, students, and researchers in financial and computing disciplines to engage in a collaborative effort to create these high quality datasets. The democratic benefits of better open datasets could help LMs support retail investors as they develop financial literacy. Other domains are having success with SLMs. It is time to make that a possibility for the finance domain as well. These findings also provide room for other opportunities to improve the quality of SLM output in financial literacy applications, such as through retrieval augmented generation. However, methods like these will also require high-quality open-source data, not opinion-based social media data.

Limitations

This study examined a number of SLMs. Due to the rapid pace at which new models emerge, we did not include more recent SLMs, such as those from Meta (though we did include Tiny-Llama from the Llama community), and Deepseek. We also did not consider other models with more than three billion parameters, such as the common 7B models. However, these models require more computing power. Future studies could compare LORA and QLORA 7B models compared to FP32 or BF16 models with three billion parameters or less.

Although the inter-rater reliability was lower than we expected, we believe it to be less of a limitation of the study and more of a limitation of the existing data. The existing data has many issues that make consistent subjective rating, even among experts, difficult. The difficulty in achieving higher inter-rater reliability helped point the research team toward the deeper qualitative analysis that ultimately answered the research question and identified the

larger issue, namely the lack of quality datasets.

6 Ethics Statement

To avoid copyright violations, we chose to fine-tune the model on a widely used open-source dataset. Our focus on democratizing LMs and LM training data also helped to keep our focus grounded in ethical pursuits. Our call for more accurate open-source financial question answering datasets is also in line with our ethical pursuit to produce low-cost, privacy preserving models for financial literacy applications for those who lack an appropriate financial education to benefit from societal financial structures.

Acknowledgments

This study was made possible with funding from Michigan Technological University’s College of Computing and College of Business.

References

- AbciNLP. Investmentlm on github (abacinlp/investlm). <https://github.com/AbaciNLP/InvestLM>. [Accessed 7-11-2024].
- Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.
- ai4finance.org. GitHub - AI4Finance-Foundation/FinGPT: FinGPT: Open-Source Financial Large Language Models! Revolutionize We release the trained model on HuggingFace. — github.com. <https://github.com/AI4Finance-Foundation/FinGPT>. [Accessed 11-06-2024].
- Mousumi Akter, Naman Bansal, and Shubhra Kanti Karmaker. 2022. Revisiting automatic evaluation of extractive summarization task: Can we do better than rouge? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1547–1560.
- Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
- Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.
- Neil Burgess, Jelena Milanovic, Nigel Stephens, Konstantinos Monachopoulos, and David Mansell. 2019. Bfloat16 processing for neural networks. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pages 88–91. IEEE.
- Jian Chen, Peilin Zhou, Yining Hua, Yingxin Loh, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024. Fintextqa: A dataset for long-form financial question answering. *arXiv preprint arXiv:2405.09980*.
- Zuliani Dalimunthe, Ananda Chairunnisa, and Rachmadi Agus Triono. 2023. Are social media users blindly following influencers’ recommendations on investing? *The Indonesian Capital Market Review*, 15(1):1.
- Jill E Fisch. 2022. Gamestop and the reemergence of the retail investor. *BUL Rev.*, 102:1799.
- Rudolf Flesch. 2007. Flesch-kincaid readability test. *Retrieved October*, 26(3):2007.
- Xue-Yong Fu, Md Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and Shashi Bhushan TN. 2024. Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization? *arXiv preprint arXiv:2402.00841*.
- Xiaohui Gao and Tse-Chun Lin. 2015. Do individual investors treat trading as a fun and exciting gambling activity? evidence from repeated natural experiments. *The Review of Financial Studies*, 28(7):2128–2166.
- Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.
- Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.
- Jimin Huang, Ruodyu Xiang, Chen Zhengyu, and Clémentine Fourrier. Thefnai/flare-finqa. <https://huggingface.co/datasets/TheFinAI/flare-finqa>. [Accessed 7-11-2024].
- Tagore Rao Kosireddy, Jeffrey D. Wall, and Evan Lucas. 2024. Exploring the readiness of prominent small language models for the democratization of financial literacy. *Preprint*, arXiv:2410.07118.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.
- Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023a. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 374–382.
- Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

- Annamaria Lusardi and Olivia S. Mitchell. 2014. The economic importance of financial literacy: Theory and evidence. *Journal of Economic Literature*, 52(1):5–44.
- Macedo Maia, André Freitas, Alexandra Balahur, Siegfried Handschuh, Manel Zarrouk, Ross McDermott, and Brian Davis. Financial opinion mining and question answering. <https://sites.google.com/view/fiqa/home>. [Accessed 7-11-2024].
- Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2023. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, pages 387–402. Springer.
- Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, et al. 2024. Openelm: An efficient language model family with open-source training and inference framework. *arXiv preprint arXiv:2404.14619*.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. [Cross-task generalization via natural language crowdsourcing instructions](#). *Preprint*, arXiv:2104.08773.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Lasse Heje Pedersen. 2022. Game on: Social networks and markets. *Journal of Financial Economics*, 146(3):1097–1119.
- Mercy Ranjit, Gopinath Ganapathy, Shaury Srivastav, Tanuja Ganu, and Srujana Oruganti. 2024. Rad-phi2: Instruction tuning phi-2 for radiology. *arXiv preprint arXiv:2403.09725*.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic answer similarity for evaluating question answering models. *arXiv preprint arXiv:2108.06130*.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chafin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. [Multitask prompted training enables zero-shot task generalization](#). *Preprint*, arXiv:2110.08207.
- David N Sattler, Patrick E McKnight, Linda Naney, and Randy Mathis. 2015. Grant peer review: improving inter-rater reliability with training. *PloS one*, 10(6):e0130450.
- Natalie Schluter. 2017. [The limits of automatic summarisation according to ROUGE](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.
- Tanmay Sen, Ansuman Das, and Mrinmay Sen. 2024. Hatetinyllm: Hate speech detection using tiny large language models. *arXiv preprint arXiv:2405.01577*.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. [Finetuned language models are zero-shot learners](#). *Preprint*, arXiv:2109.01652.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.
- Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. [Pixiu: A large language model, instruction data and evaluation benchmark for finance](#). *Preprint*, arXiv:2306.05443.
- Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023a. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*.
- Yi Yang, Yixuan Tang, and Kar Yan Tam. 2023b. Investlm: A large language model for investment using financial domain instruction tuning. *arXiv preprint arXiv:2309.13064*.
- Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix: Dataset Instruction Prompts

The FinGPT FiQA_QA dataset assigns one of five fixed instruction prompts to each example. We did not author or modify these instructions; they are used verbatim as provided by the dataset. The five instructions are:

1. *Utilize your financial knowledge, give your answer or opinion to the input question or subject. Answer format is not limited.*
2. *Offer your insights or judgment on the input financial query or topic using your financial expertise. Reply as normal question answering.*
3. *Based on your financial expertise, provide your response or viewpoint on the given financial question or topic. The response format is open.*
4. *Share your insights or perspective on the financial matter presented in the input.*
5. *Offer your thoughts or opinion on the input financial query or topic using your financial background.*

These five instructions are near-synonymous paraphrases of the same request. Because they were applied per example rather than chosen or tuned by us, the small effect of the instruction prompt observed in our ablation (Section 4.1) most plausibly reflects that generic, semantically overlapping instructions add little signal for these models on this data, rather than indicating that a poorly chosen prompt or rigid instruction inherently harms performance.

B Appendix: Examples of Poor Quality Data

The following sections provide examples of issues that exist within the dataset. Where quantitative analysis was feasible, quantitative data is also provided.

B.0.1 Examples of Unclear and Cryptic "Questions" in FiQA_QA Dataset

The following question is a statement and not a well formed question:

Question: Living in my own rental property

The next question prompt is also a statement, but there is at least an implicit question:

Question: Short Term Capital Gains tax vs. IRA Withdrawal Tax w/o Quarterly Est. Taxes

Further, cryptic question prompts exist in the dataset. The intention of the questioner is open to vast interpretation in the following question prompt:

Question: Buying an option in the money, at the money, or out of the money

As a simple quantitative test of the number of potential questions that were unclear and poorly formed, we examined whether all of the questions ended with a question mark. Of the 17,110 questions in the dataset, 4,614 questions (26.967%) lacked an ending question mark.

B.0.2 Examples of Questions with Missing Context in FiQA_QA Dataset

The following question was missing context brought up in the human response. Even the question itself makes reference to "advice" that is not mentioned in the question prompt. Clearly, something is missing from the question prompt.

Question: Is this mortgage advice good, or is it hooey?

Answer: That makes no sense at all. They try to compare and that's exactly the same as comparing apples versus oranges. Mortgage is long-term loan, so for the first many years the huge part of the payment will go to repaying interest, so that ratio 1 will indeed be something like 20% or more despite the fact that the interest rate on the mortgage is much lower - something around 6%. HELOC will have the interest rate of 6%, but it will have the same structure so that you have equal payments, so if you compute that ratio 1 it will be very close to that of the mortgage. The bottom line is - if HELOCs were that great noone would apply for mortgages. You should stick to making

extra payments towards the principal on the mortgage.

Similarly, the following question seems to be missing important context. The human response makes mention of conversation provided by the questioner that doesn't appear in the question itself.

*Question: Is Cost of Living overstated?
Answer: You mention: High rent places are usually also high property value places. Given the tax incentives, it seems like a good long-term idea to grab a house, so if we assume you have the option of working and buying a house in a high CoL or a low CoL city, I think you'd prefer the high cost. Because essentially, after 30 years, you'd have a million dollar house vs a quarter million dollar house. You've captured three quarters of a million dollars in rent, given my napkin math hypothetical. I think you're forgetting about some of the associated costs with "owning" a home, including:*

B.0.3 Examples of Unanswered Questions with Hallucinated Context

The following response from the OpenELM-270M model does not directly answer the question, but simply tells a story about starting a business in the UK while asking questions related to the original prompt.

*Question: How to safely earn interest on business profits (UK)
Answer: I recently bought my first business. I am a bit nervous as I do not know much about the UK market, so I have some questions. The business is a food shop with a very small store. I'm currently working part time and earning £15/hour. I would like to be able to earn £20/hour. The business operates Monday to Friday. I have to work on my lunch break, and I do not have any time to work after work. My only plan is to buy a business loan. My lender says that I could be eligible to get up to £4000 if I buy the business outright. Is this correct? I have no idea what a business loan is and I am worried about the interest rates. I know that it's not ideal to buy a business if I do*

not have the cash to pay it off in full. Is there any way I could avoid this? Thank you so much in advance for any advice you could give me! I would be really happy if you could tell me how I can achieve the £2

Even the 3B parameter version of OpenELM showed similar issues. For example, the following response shows the model responding as though it had an investment account with Schwab. Further, it hallucinated multiple index funds and their prices. The answer only tangentially answered the question.

*Question: Should I continue to invest in an S&P 500 index fund?
Answer: You're right. I've been using Schwab for a while and haven't noticed the 500-dollar error in a 5000 index fund. But let's say you don't want to use Schwab. You can use any "Index funds" from Vanguard or TI (Trudeau). 10000 Vanguard's "Tax-Free" funds will cost you \$10,000, and \$5000 will go to buy the same number of shares of 500 Indexes. (If you're using Schwab's website, \$10,500 will go for the same thing). 3000 Indexes (500-funds) will cost you \$3000 (you can use Schwab, or a bank or a mutual fund company like Fidelity). 500 (5000-fund) will cost you about \$500. The Vanguard and TI are all mutual fund companies, so you pay a fund company to manage your money (in a bank, you would go*

The OpenELM models were not the only models to suffer from unanswered questions with hallucinations. Examples of this issue can be found across models. For example, the Gemma-2B model hallucinated that it was an insurance agent with a client whose house burned down. Although the response was tangentially related to the question, it made no mention of high-net worth individuals and the necessity of insurance for individuals with substantial wealth.

*Question: For very high-net worth individuals, does it make sense to not have insurance?
Answer: You should have insurance to cover your potential liability. For example, I have*

a client with a house that burned down. His insurance policy covered the cost to replace the house. If he had not had that coverage, he would have been forced to come up with a lot of cash to pay for a new home.

tenants pay \$30k for their unit. They have had no problems with it, and it is just an extra \$3000/yr to have a place that is only \$250/yr.

B.0.4 Examples of Hallucinated Context Surrounding Relevant Answers

The following response from OpenELM-270M hallucinates a wife and child while answering the question (though with some contradictions in the answer).

Question: Is this mortgage advice good, or is it hooey?

Answer: I think the advice is a bit overstated. I am not a realtor, but my gut says that this advice is overstated. This is the advice that I would give my wife and daughter if they are considering buying a house. If I had to buy a house and it was my wife and daughter, I would buy it for a number of reasons. For example: I know that my wife and daughter are young and don't want to spend a lot of money on a house. They are not ready for a house yet. This advice would help them realize that they are ready to buy a house. It could save them from a lot of regret later on.

Although the response answers the question to some extent, it does so through a made up narrative.

Similarly, in the following response, the same model invented a life in Toronto while at least partially answering the question.

Question: Is Cost of Living overstated?

Answer: It's not overstated. I live in Toronto, and we do pay about \$2500/yr for utilities. If you were living in a place with a \$1500/year utilities bill, I'd still suggest you move. The \$300 you mentioned as your rent for the unit is actually a discount on your monthly cost. The cost to you is \$4800/yr. If you were \$150k/yr out of work, I'd suggest you move. You can probably save money in your own area and you'll be a better tenant. Edit - The \$30k is actually the rent. I've had

From Understanding to Engagement: Personalized pharmacy Video Clips via Vision Language Models (VLMs)

Suyash Mishra^a, Qiang Li^b, Anubhav Girdhar^c, Srikanth Patil^c

^aRoche, ^bAccenture, ^cInvolead,

suyash.mishra@roche.com, qiang.i.li@accenture.com, anubhav.girdhar@involead.com, srikanth.patil@involead.com

Abstract

Vision Language Models (VLMs) are poised to revolutionize the digital transformation of pharmaceutical industry by enabling intelligent, scalable, and automated multi-modality content processing. Traditional manual annotation of heterogeneous data modalities (text, images, video, audio, and web links), is prone to inconsistencies, quality degradation, and inefficiencies in content utilization. The sheer volume of long video and audio data further exacerbates these challenges, (e.g. long clinical trial interviews and educational seminars).

Here, we introduce a domain-adapted Video-to-Video Clip Generation framework that integrates Audio-Language Models (ALMs) and Vision Language Models (VLMs) to produce highlight clips. Our contributions are three-fold: (i) a reproducible Cut & Merge algorithm with fade-in/out and timestamp normalization, ensuring smooth transitions and audio/visual alignment; (ii) a personalization mechanism based on role definition and prompt injection for tailored outputs (marketing, training, regulatory); (iii) a cost-efficient e2e pipeline strategy balancing ALM/VLM-enhanced processing. Evaluations on Video-MME benchmark (900) and our proprietary dataset of 16,159 pharmacy videos across 14 disease areas demonstrate 3–4× speedup, 4× cost reduction, and competitive clip quality. Beyond efficiency gains, we also report our methods improved clip coherence scores (0.348) and informativeness scores (0.721) over state-of-the-art VLM baselines (e.g., Gemini 2.5 Pro), highlighting the potential of transparent, custom extractive, and compliance-supporting video summarization for life sciences. [Demo access](#).

* Patent application to EPO: 25175653.2

1 Introduction

In contemporary digital content landscape, efficient management and understanding of video assets are

paramount, particularly within specialized domains such as medical and sectors (Zhang et al., 2024). Large volumes of long-form pharmaceutical and medical videos, including clinical trial interviews (Srinivasan et al., 2025), drug manufacturing workflows (Otesteanu et al., 2021), educational seminars, and long conference recordings, are routinely produced and remain difficult to consume, review, and reuse on scale.

Traditionally, the review and repurposing of such long-form videos is a labor-intensive process, often requiring days or weeks of manual effort by multiple stakeholders (Wu et al., 2018; Corin and Li, 2021; Yang et al., 2025a). Users typically rely on titles and brief summaries before manually navigating long recordings to assess relevance, a workflow that frequently leads to reduced interest. Generating concise highlight clips tailored to medical and pharmaceutical content can substantially reduce review time (Liu et al., 2020), foster viewer engagement, and improve reuse of existing video assets (Liu et al., 2020; Guo et al., 2024a). Recent research and commercial systems for video summarization and clip generation can be broadly categorized into three paradigms: **Frames-to-Video**, **Direct Video-to-Video**, and **Prompt- or Image-conditioned Video Generation**.

Frames-to-Video approaches extract and assemble key frames into short clips, offering high customization allowing frame adjustments, removal, additions, along with audio manipulation, but incurring very long preprocessing and generation times, and often suffering from temporal discontinuities (e.g. jumping frames) (OpusClip, 2023; Pika Labs, 2024; Synthesia, 2024; HeyGen, 2024). **Direct Video-to-Video** methods process entire videos end-to-end, enabling faster generation and smoother outputs (less than 1-2 minutes for one short clip), but typically operate as black-box systems with limited transparency or control over clip selection, and post-selection requires more time to

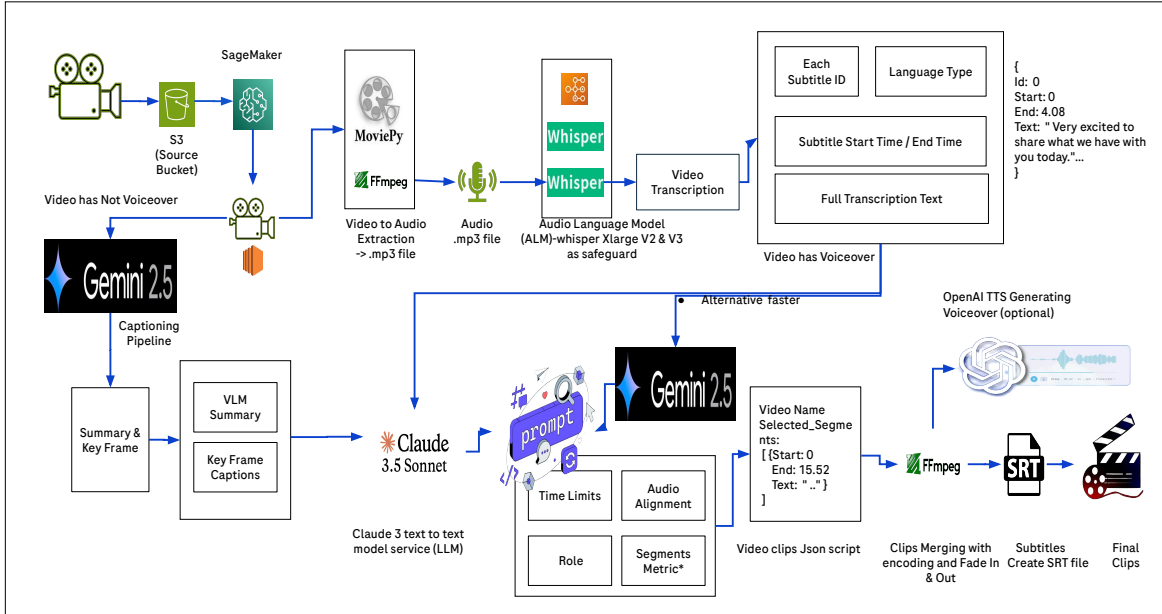


Figure 1: Solution architecture blueprint of the underlying LLM/VLM tech stack for video clip generation.

choose the best short clips. (Wang et al., 2018; Liu et al., 2021; Bansal et al., 2018; Li et al., 2025).

Recent advances in VLMs and ALMs have enabled prompt-based video generation and multi-modal understanding (Brooks et al., 2024; Yang et al., 2025b). Companies like Stability AI (Stability AI, 2024), Sora (Brooks et al., 2024), Elevenlabs (ElevenLabs, 2023), DeepBrain (DeepBrain AI, 2024), Kaiber (Kaiber AI, 2024), and Animoto (Animoto, 2024) adopt **Single Image+Prompts-to-Video** methods, which can produce visually coherent clips efficiently, but limited in Image/Video duration length (e.g single-image input, enterprise APIs may access frames, <20MB, max approx. 1 hour (video only) input) and computationally expensive. Their direct application remains ill-suited for clinical or pharmaceutical settings where preserving original frames (e.g., expert explanations, interviews, or procedural steps), exact source timestamps & original audio, and traceability are critical (Hu et al., 2024; Jiang et al., 2025). This setting introduces additional constraints, including long video durations (<2 minutes to 3 hours), strict latency and cost requirements, and the need for auditable, role-specific clip selection.

In this paper we investigate the following center research questions: **RQ1:** Can an hybrid ALM/VLM-based pipeline generate high-quality highlight clips from long medical videos under strict non-synthetic, efficiency and cost constraints? **RQ2:** How to improve temporal coherence and

transition smoothness compared to direct video flame concatenation? **RQ3:** How do role-based prompt personas influence clip selection behavior?

To address these questions, we propose an *Infinite Video-to-Video Clips Generation* framework designed for long-form pharmaceutical and medical videos. Our contributions are as follows:

- **Cut & Merge Algorithm:** A reproducible, patent-pending algorithm that normalizes timestamps and applies fade-in/out boundaries to eliminate jump cuts, audio glitches, and frame freezing (see Fig:2, Alg:1)
- **Personalization Mechanisms via Role Definition & Prompt Injection:** Systematically tailoring clip generation (e.g., promotional vs. educational styles) while maintaining transparency (see Fig:8, 6, 12).
- **Infinite Video-to-Video Clips Generation framework,** as illustrated in Fig:1, suitable for industrial production adoption, characterized by lower cost 4x, 3-4x high speed (see Table 2, Figure 3) and better clip-quality scores (Clip Coherence, Informativeness, Redundancy scores in Table 4).
- We also present five key technical findings substantiated by comprehensive evaluation, namely, speech detection and voiceover extraction, a security safeguard for the Whisper

model, multimodality versus single modality, alignment and clip personalization.

2 Related Work

Research on video summarization and generation has evolved rapidly in recent years. Diffusion-based approaches (Ho et al., 2022; Wang et al., 2025; Xing et al., 2023) have become the dominant paradigm video generation, replacing earlier GAN- and autoregressive-based methods (Vondrick et al., 2016; Yan et al., 2021). While diffusion models achieve impressive visual fidelity, scaling them to long videos remains challenging due to high computational costs, poor frame coherence and consistency. For long video processing — essential in domains such as medical procedure, patient interview analysis, and pharmaceutical education — recent work has focused on compressing or summarizing video content into token representations suitable for VLM input. Specific works like Video-XL (Shu et al., 2024), FiLA-Video (Guo et al., 2025), and LongVLM (Weng et al., 2024) have shown promise by using token-compression architectures to balance global context and local detail.

The rise of vision-language models (VLMs) has further enabled multimodal understanding across video, image, and audio modalities (Brooks et al., 2024; Yang et al., 2025b). Benchmarks such as Video-MME (Fu et al., 2025) provide standardized evaluation protocols for long-video comprehension. Large-scale models including Qwen-VL (Bai et al., 2023), Gemini 2.5 (Google DeepMind, 2025), and GPT-4o (OpenAI, 2023) demonstrate strong performance on alignment and summarization tasks. However, these systems are often costly to deploy at scale, operate as black-box models with limited transparency, and typically focus on video-to-text outputs rather than extractive video-to-video generation, particularly for long-duration inputs. Several recent works explore LLM-driven video summarization and clip generation. Lee et al. (Lee et al., 2025a) demonstrate the potential of LLMs for clip generation but highlight challenges in maintaining logical flow and informativeness, e.g. generate synthetic or poorly aligned content that risks clinical misinterpretation (Guo et al., 2024b), (Tariq et al., 2025) or fail to provide customize clips. Representative academic pipelines such as *LLMVS* use frame captioning followed by LLM-based scoring to generate summaries optimized for standard video summarization bench-

marks (Lee et al., 2025b). *V2Xum-LLM* explores cross-modal video-to-text and video-to-video summarization using instruction-tuned datasets such as Instruct-V2Xum (Hua et al., 2024). Prompt-to-Summaries methods enable zero-shot or query-controlled video skimming, but are primarily designed for short, publicly available video (Alaa et al., 2024).

While these approaches demonstrate strong performance on academic benchmarks, they operate under assumptions that differ fundamentally from our setting. In pharmaceutical video-to-video clip generation, the primary objective is not to synthesize visually appealing transitions or animations, but to identify and extract clinically meaningful segments (e.g., interviewer speech, key frames, or specific clinical interventions). Preserving original frames and audio is often more critical than generating new, visually enhanced footage, particularly under compliance and traceability requirements (Hu et al., 2024; Jiang et al., 2025). Moreover, real-world industrial deployments must handle videos ranging from under two minutes to over three hours, mostly long videos, significantly exceeding the duration of standard benchmark datasets. Such systems often face strict latency, cost, and privacy constraints across multiple commercial ALM/VLM backends, while ensuring that extracted clips remain auditable and attributable to exact source timestamps. These constraints limit the direct applicability of video generation or short video-to-text summarization pipelines, and motivate an extractive, hybrid ALM/VLM design that balances scalability with clinical reliability, a key requirement for deploying VLM-powered video-to-video systems in the pharmaceutical industry.

3 Dataset And Experimental Settings

Here, we primarily adopt well-established Video-MME (Fu et al., 2025), along with evaluations of major 11 SOTA VLMs, as well as our pharmacy proprietary dataset shown in Table 1, Figure 10.

Video-MME (Fu et al., 2025) is the first full-spectrum multi-modal evaluation benchmark designed specifically for video-based MLLMs. It stands out from existing benchmarks with several key features: (1) Diversity in video types, covering six primary visual domains with 30 subfields to ensure broad scenario generalizability; (2) Temporal coverage, including short-, medium-, and long-term videos ranging from 11 seconds to 1 hour. It

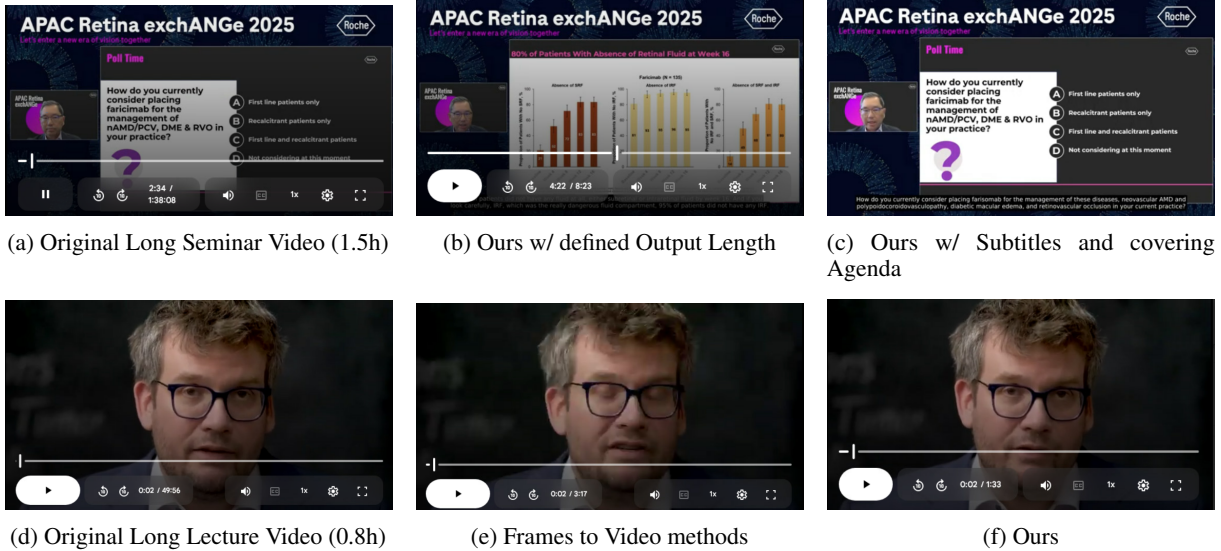


Figure 2: **Qualitative comparison of our Infinite Video-to-Video Clips pipeline against frame-based approaches** (e.g. Runway Gen-2 (RunwayML, 2023)). Our method supports arbitrary input durations, allows user-defined output lengths, automatically extracts agenda-relevant segments, adds subtitles and vertical playback, while overcoming choppy transitions and frame skipping/freezing, e.g., (e).

Table 1: Statistics of Our Proprietary Dataset.

Category	Details
VLM Models Covered	Gemini 1.5 Pro, 2.0-Flash, Gemini 2.5 Pro, Gemini 2.5 Flash, Qwen-7B-VL, Qwen-72B, Claude 3.5 Sonnet, GPT-4o.
ALM Models Covered	Whisper-turbo V3 and Whisper-large V2.
Number of Videos	Over 16,159 Long Videos. Sampled 300
Number of Audios	Over 888 .
Covered Variants	Over 14 Diseases areas. From Nephrology, Ophthalmology to Hematology, Immunology, Dermatology.
Video format Types	8 types : MP4, M4V, QuickTime, WMV, WebM, MSVideo, MPG, and 3GPP.
Audio format Types	4 types : '.mp3', '.wav', '.m4a', '.flac'.
Video Lengths	Major Longer length video from >30 min to over 3 hours.
Language Types	Over 20 languages, including German, Italian, English, Mandarin, Hokkien, Hindi, Korean, French, Dutch, Spanish, and more.

comprises 900 manually selected videos, totaling 254 hours, with 300 videos in each categories.

Furthermore, our findings are validated using proprietary data from 14 disease areas, including sampled over 300 long-form videos (<2mins minutes to over 3 hours, 8+ formats) and 888 audio.

4 Methodology, Business Impact And Technical Features

As illustrated in Figure 1, our pipeline integrates ALMs, VLMs, various practical libraries, prompt-based segment selection, and a Cut & Merge post-processing algorithm to generate extractive highlight clips from long-form videos.

Given an input video, we first extract voiceover transcriptions using Whisper V2 and V3 ALMs (Radford et al., 2022; OpenAI, 2023) by a predefined schema. We employ both versions for quality control: while Whisper V3 provides four times faster processing and more accurate language type detection, Whisper V2 often yields more complete sentence boundaries. This complementary behavior is analyzed in Table 2 and Figure 5. The resulting transcripts are then aligned with precise timestamps and serve as one of primary textual input for downstream segment selection.

Using the aligned transcription, we extract candidate video segments through structured prompt injection under four constraints: (1) user-defined target clip length, (2) video role or style (e.g., marketing, training, educational etc.), (3) audio-visual alignment with smooth fade-in and fade-out transitions, and (4) Segment selection metrics. These metrics prioritize full-video coverage (including beginning and end segments), workflow transitions, agenda-highlighted content or video introduction, and audio cues such as pauses or changes in speaker intonation. For videos without voiceover, VLM-based visual understanding is used to identify salient segments based on visual content alone.

Rather than directly concatenating selected segments using e.g. FFmpeg, we designed Cut & Merge algorithm (Algorithm 1) to ensure visual & audio coherence and smooth transitions. Given

Algorithm 1 Cut & Merge Clip (Fade In/Out + Re-encoding)

Require: \mathcal{S} (selected segments with start, end), video V , output dir D

Ensure: merged highlight clip V_{out}

```
1: mkdir( $D$ )
2:  $\mathcal{L} \leftarrow []$   $\triangleright$  concat segment list file entries
3: for  $i \leftarrow 1$  to  $|\mathcal{S}|$  do
4:    $(s, e) \leftarrow (\mathcal{S}[i].\text{start}, \mathcal{S}[i].\text{end})$   $\triangleright$  Timestamp
5:    $p \leftarrow D/\text{clip}_i.\text{mp4}$ 
6:   PROCESSCLIP( $V, s, e, p$ )  $\triangleright$  w/ fade + re-encode
7:    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\text{"Processed segment ' } p \text{' }\}$ 
8: end for
9: CONCATCLIPS( $\mathcal{L}, V_{\text{out}}$ )  $\triangleright$  ffmpeg concat demuxer
```

ProcessClip(V, s, e, p)

```
ffmpeg -y -ss  $s$  -to  $e$  -i  $V$  -vf format=yuv420p -af
"afade=in:0:0.5,afade=out:( $e-s-0.5$ ):0.5"  $\triangleright$ 
audio fade-in/out -c:v libx264 -preset fast -crf
23  $\triangleright$  Avoid direct concatenation '-c copy' -c:a
aac -b:a 128k p  $\triangleright$  re-encoding to avoid jump cuts and
frame freezing
```

ConcatClips($\mathcal{L}, V_{\text{out}}$)

```
ffmpeg -y -f concat -safe 0 -i list.txt -c copy
V_out  $\triangleright$  concatenate processed clips using FFMpeg concat
demuxer
```

the precise segment timestamps produced by our ALM/VLM pipeline, each segment is then processed individually as follows: (i) audio fade-in and fade-out are applied within a fixed temporal window to suppress background noise and abrupt audio transitions; (ii) video frames are re-encoded with synchronized visual fade-in/out (typically within a ± 0.5 s window) to mitigate frame freezing and incompatibilities issues; (iii) processed segments are then concatenated in their desired order to produce the final clip, where user can also enlarge or remove segments slices (optional). Figure 2 (e-f) compares our method with naive concatenation using standard tools (e.g., FFMpeg), illustrating improved transition smoothness and reduced skipping/freezing frames.

Our framework supports both vertical and horizontal playback formats and optional subtitle integration. It is designed to handle input videos of arbitrary length, overcoming the duration and storage constraints of many VLM-based systems (e.g. 20MB up to 1GiB). The pipeline offers transparent customization by allowing users to tailor video clips and remove or enhance specific sections. It is optimized for scalable deployment, achieving a 94.44% time effort reduction for longer video summarization and 88% time reduction across all pharmacy video categories compared to manual inspection. Finally, medical experts qualitatively

assessed whether extracted clips preserved factual correctness and speaker intent. This evaluation is intended to assess alignment and usability in real life, and representative examples are reported in Appendix Table 6.

5 Main Results

How to secure the complex & fast voice extraction? In our pipeline, FFMpeg (FFmpeg developers, 2024) is used for key operations such as format conversion and audio extraction to ensure broad compatibility. It general performs well on 1–2 hour videos with clean audio, averaging costs only 46.38 seconds per video. However, FFMpeg also encounter issues with long filenames, unsupported characters, incorrect encoding or compatibility errors. To handle more complex cases in reality, we also integrate MoviePy (Zulko, 2015) as a fallback solution. MoviePy offers a Pythonic interface for programmatic and efficient audio extraction across video batches. While slightly slower (averaging 64.71 seconds), it delivers comparable extraction quality. This combined approach ensures robust audio processing across a wider range of video files/kinds, with extraction times typically ranging from 10 to 79 seconds, as shown in Table 2).

Fragmented Sentences? Safe and accurate combination of Whisper V3/V2. As illustrated in Figure 1, 5), Whisper V3 (OpenAI, 2023) Whisper V3 offers significant speed improvements (up to 4x faster on industry datasets and VideoMME) and reliable language type detection. However, it often produces fragmented sentences, leading to unstable segmentation and less precise timestamp alignment (see Fig 5). In contrast, Whisper V2 (Radford et al., 2022) Whisper V2 provides more accurate timestamps, crucial for tasks like clip-cutting, and is therefore our primary model. While V2 occasionally excels at capturing full sentences, real-world tests with English videos revealed sporadic language misidentification (e.g., mistaking English for Welsh), resulting in corrupted transcriptions. To address this, we use V3 as a secondary validation layer for security guidance (see Figure 1).

Modality: A Key to Cost Reduction, Speed, Clips coherence As shown in Figures 3 and 4, utilizing ALM for accurate transcription, combined with Gemini 2.5 Pro (text-to-text), significantly reduces costs (up to 4-4.5 \times less, Table 13) and accelerates processing time (up to 4 \times faster, 30-55 seconds per video) compared to using Gemini 2.5

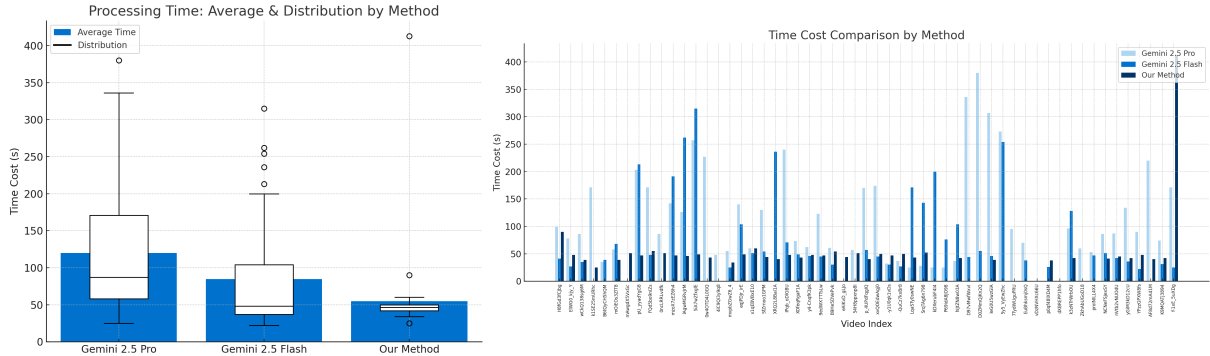


Figure 3: **Processing Time Comparison:** Gemini 2.5 Pro / Flash vs Our Methods for generating video clips script on VideoMME Long Video Dataset. Gemini Pro is the slowest (avg. ~ 120 s/video), with peaks on longer videos (e.g., 380s). Flash is faster (~ 80 – 85 s) but still slower than our method (~ 30 – 55 s), except for one outlier (413s). Our method is in general 3-4x faster.

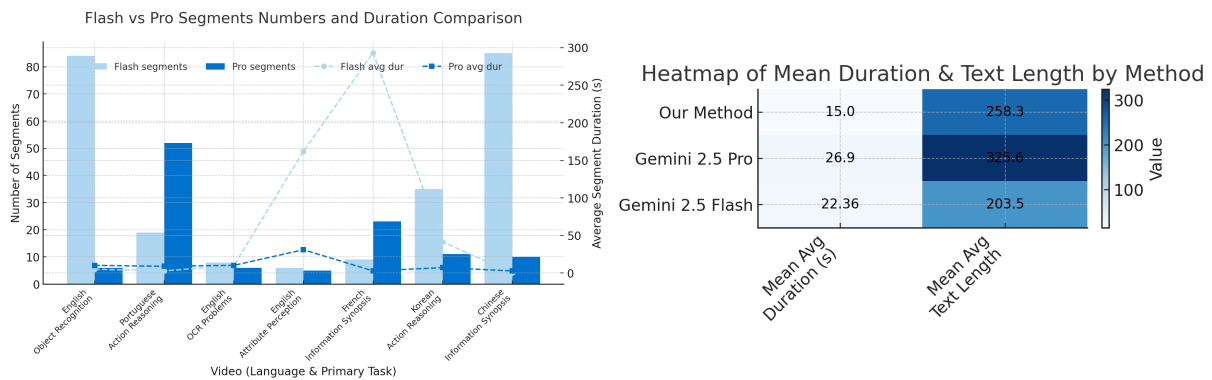


Figure 4: **Comparison between the Gemini 2.5 Flash, Pro vs our, based on the number of select segments and the quality.** Here, we assess quality based on factors like segment length / numbers or the presence of coherent text. Flash often returns many but fragment segments (e.g. “Video ID: tsIKtm6Le1s”: 85 piece of segments). Pro tends to pick fewer, longer segments (reflected in its lower segment counts but higher average durations). Our method achieves balanced selection 4.37 segments vs. 7.38 (Gemini 2.5 Pro) and 13.30 (Gemini 2.5 Flash) for final clips.

Pro on full video input (average 120s, max 350s per video). Our method maintains comparable accuracy in generating less redundant, high coherent, highly informative video clip scripts (258 meaningful words) compared to Gemini 2.5 Pro (325) and Gemini 2.5 Flash (203).

Furthermore, as shown in Table 4, Clips coherence scores (fraction of original segments covered by any summary clips segment), Informativeness (cosine similarity to its best-matching original segment, averaged across segments), and Redundancy scores (mean pairwise cosine similarity among segments (off-diagonal)), our approach efficiently provides comparable coherence (0.348), higher informativeness (0.721), reduced redundancy (0.339), and increased stability.

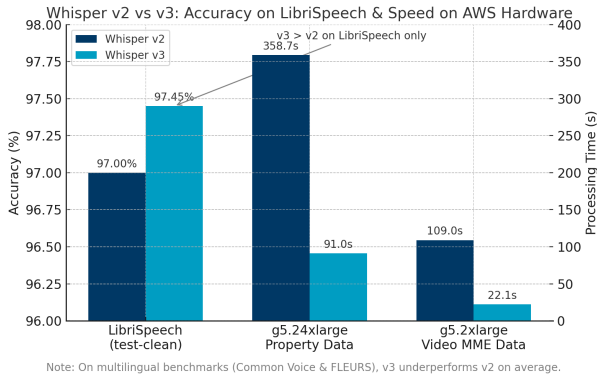
Gemini 2.5 Flash while generally faster (averaging 80–85 seconds per video) than 2.5 Pro, struggles more with non-English, abstract, or non-verbal video content. Gemini Pro tends to select fewer but longer segments, often influenced by visual infor-

mation, leading to lower segment counts, higher average durations and standard deviations, and even reduced informativeness scores. Flash, conversely, selects more fragment but shorter clips (see Fig 4).

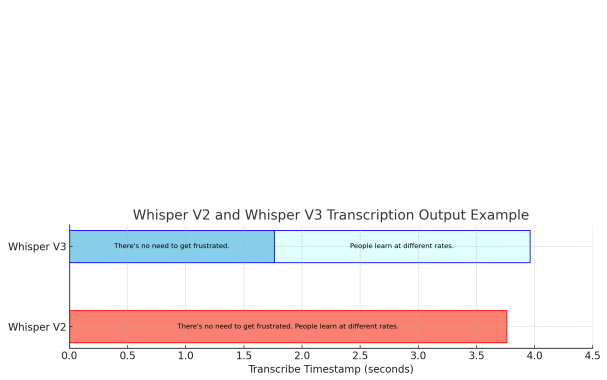
Audio and Speaker Alignment: Direct Concatenation Will Not Work! As shown in Figures 2(e), directly concatenating video segments using frame-level and ALM-generated timestamps often results in frame jumps and background noise. To address this, we developed an algorithm that combines segment encoding (for noise reduction) with audio/video fade-in and fade-out transitions (± 0.5 seconds for smooth transitions). This approach avoids the visual and auditory glitches typically seen with direct FFmpeg concatenation. Furthermore, we standardize ALM-generated timestamps to two decimal places to improve segment precision and overall clip smoothness. Figures 2(e)–(f) clearly demonstrate the advantages of our method: where other approaches show at the same timestamp a frozen eye or losing frames, but

Table 2: **Processing time comparison in each step.** Experiment setting: Proprietary Dataset and VideoMME, SDPA, FPS=0.01, AWS ml.g5.24xlarge, g5.2xlarge instance. * means Gemini 2.5 Pro has Timeout for longer waiting or error cases. - means not applicable.

Processing time cost (avg / per video, in seconds)							
Video Type	VLM (Directly Video to Text)	ALM (Transcription by Whisper Turbo V3 V3 vs Large-V2)	FFmpeg/Moviepy (Voice Over abstraction, video to audio file)	LLM (Bedrock Claude Sonnet / Gemini 2.5 Pro Extract Timestamp and improve summary quality)	Cut & Merge Video Clips	Adding Subtitles	Video amount VideoMME / Proprietary Data
Longer video (above 30 minutes long)	Generate Summary and Key Frame Caption 1-4 Min for Gemini 2.5 Pro *	90.96s (15.16 Mins in total, Turbo v3 model) 358.74s (59.79 Mins in total, Large-V2) (Proprietary Data & on g5.24xlarge) avg, 22.08s (3h for 489 videos on video mme by Turbo V3 model) vs avg, 109s (15h for 493 videos on by Large-V2 model) (on g5.2xlarge)	79.2s	24.37s	30s-1Min	4Mins	300 / 100
Medium (2 minutes < length < 30 minutes)	1-4 Mins	17.04s (2.84 minutes in total, Turbo V3) 59.52s (9.92 Mins in total, Large-V2)	29.76s	24.37s	30s-1mins	<4mins	300 / 100
Short (less than 2 minutes)	-	4.26s (0.71 minutes In total for turbo)	10.56s	-	-	-	300 / 100
Total (avg)							1200



(a) Accuracy On LibriSpeech & Speed On AWS Hardware.



(b) V3 has more fragment transcriptions (upper V3).

Figure 5: **Qualitative comparison of Whisper V2 vs. V3: transcription accuracy on LibriSpeech (Panayotov et al., 2015) test-clean and inference speed on AWS hardware.** Whisper V3’s performance is more sensitive to GPU type—achieving roughly 4–8× speedups and higher Accuracy compared to V2, but cuts speech more aggressively, resulting in increased sentence fragmentation that complicates downstream timestamp alignment and segment merging.

our transitions consistent with the original video.

Personalization: Prompt Injection, Role Definition, and Selection Metrics. In prompting, users can select a role and specify a maximum duration for video clips (e.g., up to 3-4 minutes). Additionally, the current process guides Gemini 2.5 Pro to explain its reasoning for selecting specific segments (See Appendix Figure 11, Table 12). This enables Gemini 2.5 to focus on key factors, such as smoothly transitions (finishing speaker’s sentence), important key segments presented agenda topics/slides, keywords, full-length coverage, speaker’s voice and pauses, and noun emphasis, rather than being distracted by numerous unrelated elements.

6 Ablation study on Role Definition and Prompt injections

How does each segment’s metrics play a different role in the final outcome? We investigate how our segment metrics shape the final outcome by conducting an ablation that isolates those factors: (i) Keywords, ii) Agenda, (iii) Speaker-voice tone. As shown in Appendix Figure 8, 1) Removing tone consistency can lead to visual-voice mismatches, 2) Agenda helps ensure clips retain key information, paragraphs, and segments, 3) Incorporating keywords (e.g., nouns, medical terms) increases the likelihood of important information being captured in segments, 4) Length coverage can prevent abrupt, incomplete clips, jump cut and sudden End-

Table 3: **Overall Accuracy of VLMs in summarization on Video-MME.** (A) Overall accuracy on Video-MME (900 videos) with/without audio transcription. (B) Keyframe/time alignment and summary accuracy on long-video subset ($n = 300$). Gemini models achieve high accuracy on meaningful summary, but still struggle with timestamps.

(a) Overall Acc. with/without audio transcription				(b) Long-video Keyframe & Summary Accuracy	
Model	w/o	w/	Δ	Method	Summary Acc. (%)
Gemini 2.5 Pro	84.7	85.2	+0.5	Gemini 2.5 Flash	94.6
Gemini 1.5 Pro	75.0	81.3	+6.3	Qwen -7b	74.3
Qwen2-VL	71.2	77.8	+6.6	Method	KeyFrame Time Acc. (%)
GPT -4o	69.0	77.2	+8.2	Gemini 2.5 Flash	35.1
LLaVA -Video	76.0	76.9	+0.9	Qwen -7b	5.4
Gemini 1.5 Flash	72.6	75.0	+2.4		
Oryx -1.5	67.3	74.9	+7.6		
InternVL2.5	67.6	74.0	+6.4		
Aria	70.3	72.1	+1.8		
LinVT	65.6	71.7	+6.1		
TPO	66.2	71.5	+5.3		

Table 4: **Clip Coherence Scores / Informativeness based on Video-MME Benchmark (All Sampled Long Videos, $n = 300$) @ $\tau = 0.6$.** **Our Method:** Concise, $4\times$ faster, $3.5\text{--}4\times$ cheaper, consistently informative (0.721 informativeness, low redundancy 0.339). **Gemini 2.5 Pro:** Best logical flow but slower, more expensive, less consistent. **Gemini 2.5 Flash:** Overly long, fragmented, highly unpredictable summaries with high deviation.

Metric	Our Method	Gemini 2.5 Pro	Gemini 2.5 Flash
Clip Coherence scores (0-1) (How logically connected adjacent clips are)	0.348 ± 0.118	0.446 ± 0.111	0.410 ± 0.127
Informativeness scores (0-1) (How well summary clips represent the source)	0.721 ± 0.078	0.674 ± 0.158	0.701 ± 0.124
# Segments number of clips	6.46 ± 1.67	9.21 ± 10.40	18.32 ± 31.03
Redundancy Scores (Overlap among clips) (0-1)	0.339 ± 0.106	0.415 ± 0.117	0.379 ± 0.138
Clips meaningful text length	258.3	325.6	203.5
Speed(Generation time per clips)	30–55s / video	120s avg / video	80–85s avg / video
Cost in Dollar	0.3 input / 2.5 output per M Token	1.25 input / 10 output per M Token	0.3 input / 2.5 output per M Token

ing. To support reproducibility, we open-source all role-specific prompt instructions in the appendix Figure 15 and to complement the demo link, we also upload additional sample clips based on non-sensitive healthcare videos: [Sample Video Clips](#).

7 Conclusion

In this work, we introduce a novel industrial, practical video-to-video clip highlights framework. This addresses the critical need for efficient video summarization and highly customizable video clip generation from long-form content. Our framework significantly facilitates content reuse in industrial settings by enabling users to generate diverse clips cost-effectively, reducing processing time from hours to mere minutes. Our key contributions include: (i) Cut & Merge algorithm that ensures smooth transitions through fade-in/out boundaries and timestamp normalization; (ii) Personalization mechanism using role definition and prompt injection to generate clips tailored for regulatory, educa-

tional, or promotional contexts; and (iii) end-to-end framework that leverages ALM pipelines for efficiency and VLM-enhanced processing for visual-heavy content. Evaluations on Video-MME and a proprietary dataset of long-form videos across 14 disease areas demonstrate $3\text{--}4\times$ speedups, $4\times$ cost reduction, and competitive clip quality compared to state-of-the-art VLM baselines.

We further highlight following key findings: 1) Modality matters, using voice transcriptions as a single modality can achieve very fast, high-quality clips at a very low cost. 2) Models like Gemini 2.5 Pro/Flash often rely on external tools (e.g., Google Search or Data APIs) rather than processing frame by frame, which can cause losing focus and produce fragmented segments. 3) Combining sophisticated prompt Injections—including transitions, agenda topics, keywords, full-length coverage, speaker voice/pauses—substantially enhances the final quality of the generated clips, surpassing the results of direct concatenation.

8 Limitations

Our methodology focuses on applying Vision–Language Models (VLMs) to life sciences, rather than proposing entirely new model architectures. A main limitation of this study is to focus on justification of the Pharm-specific benefits. We provide a baseline comparison of more than eleven VLMs using both the Video-MME benchmark and our proprietary dataset. Future work should extend this line of research to other regulated domains, such as financial services and manufacturing, to further validate the generalizability of framework.

9 Acknowledgments

We sincerely thank Samik Adhikary and Puneet Srivastava for their sponsorship support from Roche. We also appreciate the insightful discussions and technical assistance provided by Janina Kummerfeldt, and Kathrin Schwan from Accenture, Jennifer McGuire’s business support from Roche. This platform, R-ICH/RICI, would not have been possible without their contributions. We further extend our gratitude to the backend engineering teams supported R-ICH/RICI development, as well as to the healthcare professionals (HCPs), testers, and Roche Lab users whose consistent feedback brought our Video Clips use cases and enabled continuous improvement. Through this work, we aim to highlight current limitations of leading VLMs and ALMs and to contribute practical solutions to the research community, and share industry lessons learned and valuable large-scale GenAI experiments in the pharmaceutical domain.

References

- Toqa Alaa, Ahmad Mongy, Assem Bakr, Mariam Diab, and Walid Gomaa. 2024. [Video summarization techniques: A comprehensive review](#). *Preprint*, arXiv:2410.04449.
- Animoto. 2024. [Animoto: Image + text → video clip generation](#). Web-based video creator; builds short visuals using static images and prompts/text.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.
- Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. 2018. Recycle-gan: Unsupervised video retargeting. In *European Conference on Computer Vision (ECCV)*.
- Tim Brooks, Bill Peebles, Connor Homes, Will DePue, Yufei Guo, Leo Jing, Daniel Schnurr, Jason Taylor, Troy Luhman, Eric Luhman, C. W. Y. Ng, R. Wang, and Aditya Ramesh. 2024. [Video generation models as world simulators](#). OpenAI Technical Report.
- Oteteanu Corin and Qiang Li. 2021. [All you need is cell attention: A cell annotation tool for single-cell morphology data](#). In *International Conference on Learning Representations (ICLR)*.
- DeepBrain AI. 2024. [Deepbrain ai: Image-based video synthesis with prompts](#). Commercial AI platform; synthesizes video clips from a still image guided by textual input.
- ElevenLabs. 2023. [How to use ai for creating dynamic video narratives](#). Company blog; describes using ElevenLabs’ text-to-speech API to generate natural narration for AI-generated videos.
- FFmpeg developers. 2024. [Ffmpeg: A complete, cross-platform solution to record, convert and stream audio and video](#). FFMpeg Documentation.
- Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. 2025. Video-mm e: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24108–24118.
- Google DeepMind. 2025. Gemini 2.5: Multi-modal foundation models for text, image, audio, and video understanding. <https://deepmind.google/discover/blog/gemini-2-5-updates/>. Accessed: 2025-09-29.
- Yanan Guo, Wenhui Dong, Jun Song, Shiding Zhu, Xuan Zhang, Hanqing Yang, Yingbo Wang, Yang Du, Xianing Chen, and Bo Zheng. 2025. Fila-video: Spatio-temporal compression for fine-grained long video understanding. *arXiv preprint arXiv:2504.20384*.
- Yawen Guo, Xiao Liu, Anjana Susarla, and Rema Padman. 2024a. Go to youtube and call me in the morning: Use of social media for chronic conditions. *Management Information Systems Quarterly*.
- Yawen Guo, Xiao Liu, Anjana Susarla, and Rema Padman. 2024b. Go to youtube and call me in the morning: Use of social media for chronic conditions. *Management Information Systems Quarterly*.
- HeyGen. 2024. [Heygen: Ai avatar-based video clip generation](#).
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2022. Video diffusion models. In *NeurIPS*.

- Ming Hu, Kun Yuan, Yaling Shen, Feilong Tang, Xiaohao Xu, Lin Zhou, Wei Li, Ying Chen, Zhongxing Xu, Zelin Peng, Siyuan Yan, Vinkle Srivastav, Diping Song, Tianbin Li, Danli Shi, Jin Ye, Nicolas Padoy, Nassir Navab, Junjun He, and Zongyuan Ge. 2024. Ophclip: Hierarchical retrieval-augmented learning for ophthalmic surgical video-language pretraining. *arXiv preprint arXiv:2411.15421*.
- Hang Hua, Yunlong Tang, Chenliang Xu, and Jiebo Luo. 2024. V2xum-llm: Cross-modal video summarization with temporal prompt instruction tuning. *arXiv preprint arXiv:2404.12353*.
- Songtao Jiang, Yuan Wang, Sibao Song, Yan Zhang, Zijie Meng, Bohan Lei, Jian Wu, Jimeng Sun, and Zuoqiu Liu. 2025. Omniv-med: Scaling medical vision-language model for universal visual understanding. *arXiv preprint arXiv:2504.14692*.
- Kaiber AI. 2024. [Kaiber ai: Single image-to-video with prompt-driven animation](#).
- Min Jung Lee, Dayoung Gong, and Minsu Cho. 2025a. Video summarization with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Min Jung Lee, Dayoung Gong, and Minsu Cho. 2025b. Video summarization with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhuoling Li, Hossein Rahmani, Qihong Ke, and Jun Liu. 2025. Longdiff: Training-free long video generation in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kangning Liu, Shuhang Gu, Andres Romero, and Radu Timofte. 2021. Unsupervised multimodal video-to-video translation via self-supervised learning. In *Proceedings of the IEEE Conference on Applications of Computer Vision (WACV)*.
- Tianrui Liu, Qingjie Meng, Athanasios Vlontzos, Jeremy Tan, Daniel Rueckert, and Bernhard Kainz. 2020. Ultrasound video summarization using deep reinforcement learning. *arXiv preprint arXiv:2005.09531*.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*.
- OpenAI. 2023. [Gpt-4 technical report](#). *CoRR*, abs/2303.08774.
- OpenAI. 2023. [Introducing whisper large-v3](#). OpenAI Blog / Model Card.
- OpusClip. 2023. [Opusclip: Ai-powered clip generation from long-form video content](#).
- Corin Oteşteanu, Martina Ugrinic, Gregor Holzner, Yun-Tsan Chang, Christina Fassnacht, Emmanuella Guenova, Stavros Stavrakis, Andrew deMello, and Manfred Claassen. 2021. [A weakly supervised deep learning approach for label-free imaging flow-cytometry-based blood diagnostics](#). *Cell Reports Methods*.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5206–5210. IEEE.
- Pika Labs. 2024. [Pika labs: Animate still images or frames into video](#).
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. [Robust speech recognition via large-scale weak supervision \(whisper\)](#).
- RunwayML. 2023. [Runway gen-2](#). <https://runwayml.com/>. Accessed: 2025-06-01.
- Yan Shu, Peitian Zhang, Zheng Liu, Minghao Qin, Junjie Zhou, Tiejun Huang, and Bo Zhao. 2024. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*.
- Murali Srinivasan, Porawit Kamnoedboon, Dusit Nantapiboon, Piero Papi, and Umberto Romeo. 2025. [Non-surgical management of peri-implantitis with photodynamic therapy: A systematic review and meta-analysis of clinical parameters and biomarkers](#). *Journal of Dentistry*.
- Stability AI. 2024. [Stable video diffusion: Image-to-video generation from a single frame](#).
- Synthesia. 2024. [Synthesia: Ai avatars and voice-based video clips](#).
- Amara Tariq, Rimita Lahiri, Charles Kahn, and Imon Banerjee. 2025. Position: Restructuring of categories and implementation of guidelines essential for vlm adoption in healthcare. In *Proceedings of the 2025 Conference on Vision-Language Models in Healthcare*.
- Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. [Generating videos with scene dynamics](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29.
- TingChun Wang, MingYu Liu, JunYan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.
- Yimu Wang, Xuye Liu, Wei Pang, Li Ma, Shuai Yuan, Paul Debevec, and Ning Yu. 2025. Survey of video diffusion models: Foundations, implementations, and applications. *arXiv preprint arXiv:2504.16081*.

- Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2024. Longvlm: Efficient long video understanding via large language models. *arXiv preprint arXiv:2404.03384*.
- Siqi Wu, Marian-Andrei Rizoiu, and Lexing Xie. 2018. Beyond views: Measuring and predicting engagement in online videos. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*.
- Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. 2023. A survey on video diffusion models. *ACM Computing Surveys*.
- Xu Yan, Xinyang Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*.
- Shentao Yang, Haichuan Yang, Linna Du, Adithya Ganesh, Bo Peng, Boying Liu, Serena Li, and Ji Liu. 2025a. Swat: Statistical modeling of video watch time through user behavior analysis. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1, KDD '25*. Association for Computing Machinery.
- Xindi Yang, Baolu Li, Yiming Zhang, Zhenfei Yin, Lei Bai, Liqian Ma, Zhiyong Wang, Jianfei Cai, Tien-Tsin Wong, Huchuan Lu, and Xu Jia. 2025b. Towards physically plausible video generation via vlm planning. *arXiv preprint arXiv:2503.23368*.
- Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*
- Zachary Zulko. 2015. Moviepy: Video editing with python. <https://github.com/Zulko/moviepy>.

A Appendix

In this section we provide the supplementary compiled together with the main paper includes:

Our Proprietary video dataset (<2 min, up to 3 h) distribution by scientific area, as shown in Table 5; The VideoMME Dataset, detailed in Table 10; Personalization and Key Segments Logic, along with sample outputs in Figure 6, 11 and Table 14; Our video Clips landing scenarios on medical domain in Figure 9, Table 6; Ablation Study on exciting video clip generation SaaS solutions, presented in Figure 7, and Tables 7, 8 9, 10; Gemini Video-to-Text Limitations, including issue analysis and failure case examples in Figure 11, the cost breakdown in Table 13; Prompt Instruction 15.

Table 5: **Proprietary video dataset (<2 min, up to 3 h) distribution by scientific area.** Video includes clinical trials, interviews, medical lectures, promotional/non-promotional drug materials (tutorials, advertising), oral presentations, disease case studies, medical imaging/genomics demonstrations (CT scans, microscopy, DNA/sequencing), cancer morphology animations, and public health/disease education.

Scientific Area (A)		Scientific Area (B)	
Area	Count	Area	Count
Oncology	5824	Movement Disorder	263
Neuroscience	3306	Nephrology	232
Hematology	2369	Infectious Disease	155
Not Applicable	1556	Inflammatory Disease	146
Ophthalmology	1664	Dermatology	30
Respiratory Disease	312	Cardiovascular	14
Immunology	282	Metabolism	6

(a) Prompt Logic

(b) Generated Highlights Clips Scripts

Figure 6: **Key Segments Selection Logic.** This logic has four main components: Role: Defining the role for GenAI in tasks. Output Requirements: Specifically on timestamp format for uniformity and error avoidance, and rephrasing needs. Example Output: Providing examples for clip script files. Core Task Definition: This includes segment selection criteria that cover key ideas, transitions, agenda points, distributed throughout the video (including beginning and end sections if relevant). For videos, consider the speaker’s tone and pauses for smooth clip flow. Users can also input the role and additional requirements.



Figure 7: **Qualitative comparison of our Infinite Video-to-Infinite Video pipeline against single-image-to-video baselines and commercial tools** (e.g., Runway Gen-2 (RunwayML, 2023), OpusClip (OpusClip, 2023)). While existing methods generate from one or a few static frames, limited to <30 s outputs, prone to choppy transitions and frame skipping, and relying on shot-selection heuristics. Our methods support arbitrary input durations, user specified output lengths, optional subtitles, and vertical playback.

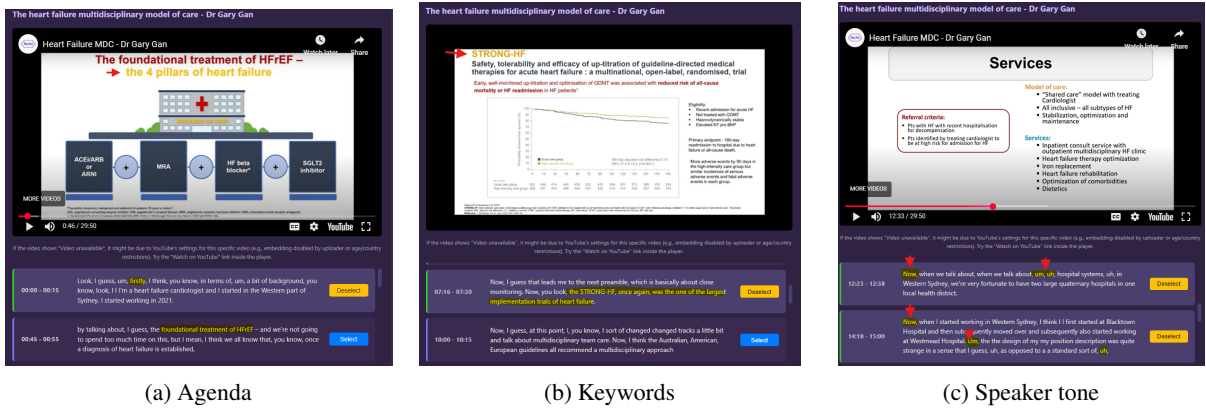


Figure 8: **Impact of role definition and prompt-selection metrics.** Agenda and keyword alignment ensure coverage of key sections; speaker tone, transitions, and fade-in/out reduce visual-voice mismatch and jump cuts; full-length coverage prevents unfinished and fragmented clips.

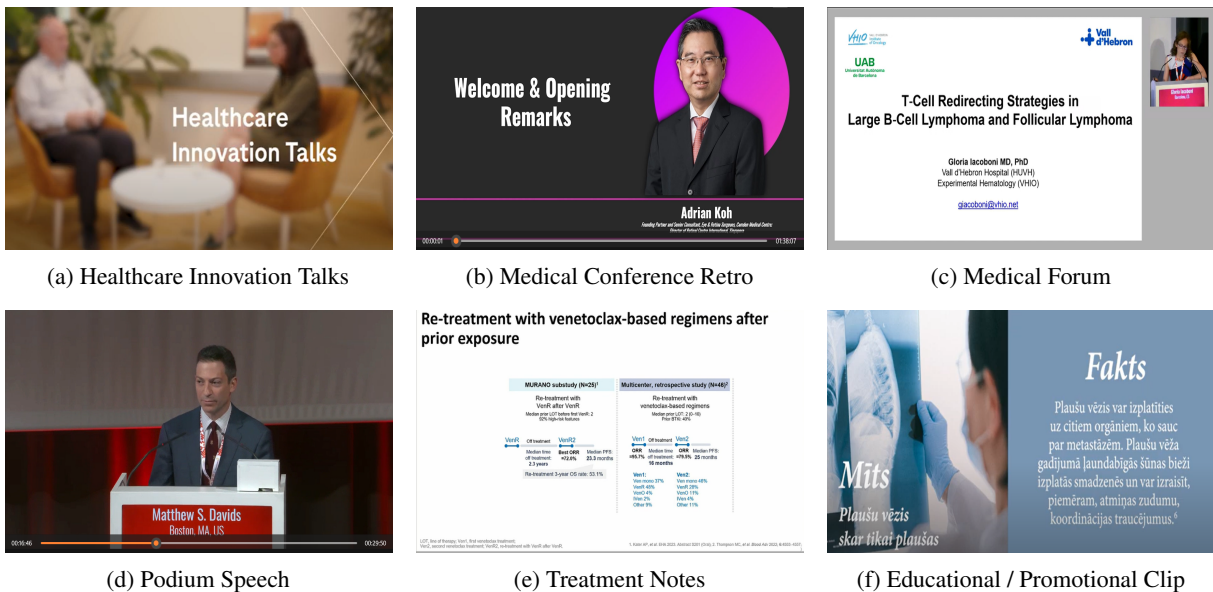


Figure 9: **Sampled video inputs:** medical interview and lecture styles processed by our pipeline (talk-show interview, keynote remark, forum lecture, podium speech, treatment-slide session, and educational/promo clips).

Landing scenarios	Patients/Doctors Interviews Listening
<p>Consolidate summary across interviews - Q1: Consolidate summaries for each interview quickly and cost-effectively via our pipeline?</p>	<p>Dr. T: Presents a study showing low rates of hypersensitivity with Phesgo, supporting its home administration. She outlines the benefits of home administration for patient quality of life and identifies suitable patient populations and implementation challenges.</p> <p>Dr. C: Provides instructions on preparing and administering Phesgo as a subcutaneous injection, highlighting steps like gathering supplies, cleaning, drawing the correct dose, and patient monitoring.</p> <p>Dr. V: Discusses the development of subcutaneous drugs like Herceptin and Phesgo, emphasizing improved patient experience by moving administration out of hospitals to more convenient locations. He also covers dosing, safety, and training.</p> <p>Dr. L: Discusses the AFFINITY trial, which demonstrated an overall survival benefit for adding pertuzumab to trastuzumab and chemotherapy in HER2+ early breast cancer patients with higher risk factors, particularly those with node involvement.</p>
<p>Track who said what at when - Q2: Would you prefer to use Phesgo in the neoadjuvant setting and in which patient population, or in adjuvant Residual Disease?</p>	<p>Dr. T: mentions its use in the adjuvant setting after completion of chemotherapy for a year of therapy (segments 40-43, approx. 3:12.12-3:28.12). She also describes its use after pre-operative (neoadjuvant) therapy if a patient achieves a PCR and needs to complete a year of HER2-targeted therapy (segments 48-49, approx. 3:38.64-3:48.52). She states, 'In the preoperative setting, we're usually giving chemo with PheSGO so that wouldn't make sense [for home administration as they are coming in for chemo]' (segments 45-47, approx. 3:31.64-3:38.64)</p> <p>Dr. C: mentions his unit has been using 'FESCO' (Phesgo) 'for the last four years in the metastatic neoadjuvant and adjuvant setting' (segment 2, 0:20.52-0:25.76). This confirms its use in these settings but his interview focuses on the practical administration rather than clinical decision-making or preference between neoadjuvant and adjuvant residual disease, or specific patient populations beyond general HER2-positive eligibility.</p> <p>Dr. V: discusses his center's involvement with 'FESGO' (Phesgo) starting with the Federica trial and rapid transition to its use once commercially available (segments 5-8, approx. 0:41.58-1:06.76). He mentions giving Phesgo with chemotherapy (segment 53, approx. 5:43.44-5:51.02: 'because they're getting chemotherapy at the same time patients are usually in the hospital when they get the first doses of subcut treatment'), which is common in the neoadjuvant setting and potentially for concurrent adjuvant chemotherapy. He also discusses patients continuing treatment after chemotherapy, fitting the adjuvant setting. He states, 'most people having chemotherapy sitting in a chair' (segment 44, approx. 4:57.9-5:05.18) in the context of identifying patients for Phesgo.</p> <p>Dr. L: focuses entirely on the AFFINITY trial results, which investigated the addition of (IV) pertuzumab to trastuzumab and chemotherapy. Phesgo is not mentioned in her transcript.</p>
<p>Key theme per molecules, trial/drugs- Q3: In light of upcoming Destiny Breast-09 readout, do you believe that all patients would need Phesgo till progression or would you consider induction Phesgo followed by Perjeta - Herceptin maintenance strategy, for which patients and what data you would need to see to implement that</p>	<p>Dr. T: does not explicitly state a *preference* for neoadjuvant vs. adjuvant residual disease, but describes its current use in both contexts where appropriate (adjuvant after chemo, or continuing HER2 therapy post-neoadjuvant chemo/surgery if PCR achieved).</p> <p>Dr. C: mentions his unit has been using 'FESCO' (Phesgo) 'for the last four years in the metastatic neoadjuvant and adjuvant setting' (segment 2, 0:20.52-0:25.76). This confirms its use in these settings but his interview focuses on the practical administration rather than clinical decision-making or preference between neoadjuvant and adjuvant residual disease, or specific patient populations beyond general HER2-positive eligibility.</p> <p>Dr. V: doesn't express a specific preference for neoadjuvant versus adjuvant residual disease for Phesgo itself, nor does he detail specific patient population criteria beyond HER2-positivity. His focus is on decentralization and patient experience.</p> <p>Dr. L: interview focuses entirely on the AFFINITY trial results, which investigated the addition of (IV) pertuzumab to trastuzumab and chemotherapy. Phesgo is not mentioned in her transcript.</p>

Table 6: **Landing scenario: Structured medical interview Q&A output from our pipeline.** Given multiple expert interviews, the pipeline extracts, identifies, and consolidates opinions around specific clinical queries (e.g., Phesgo usage). As queries become more domain-specific and knowledge-intensive, our system effectively handles such cases with precise localization of quoted statements. This demonstrates how structured summarization can support rapid evidence synthesis from long-form medical interviews.

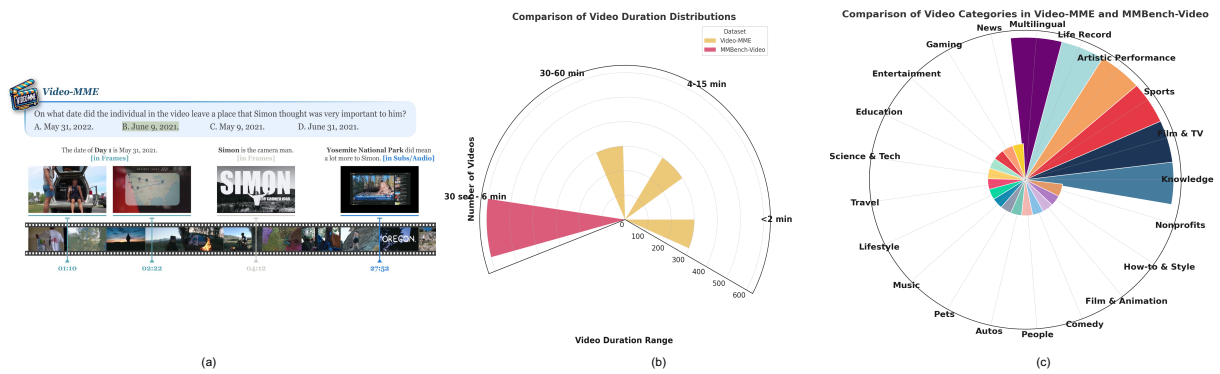


Figure 10: Comparison of the Video-MME (Fu et al., 2025) and MMBench-Video datasets (Liu et al., 2023) in terms of video categories and duration distributions. The Video-MME dataset consists of 900 videos spanning six primary visual domains with 30 subfields, categorized into 300 short-term (<2 min), 300 medium-term (4-15 min), and 300 long-term (30-60 min) videos.

Table 7: AI Video Generators from Video API (Commercial)

Company / Startup	Use case & Key features
DeepBrain API	Talking avatars; stylized visuals for short reels.
Runway API	Raw video generation from text prompts.
ElevenLabs + D-ID	Generate audio and sync to an avatar face; entertainment-industry grade.
Stability AI Sora	Image-to-video with full-scene 3D generation.

Table 8: AI Video Generators from Keyframe Images (Commercial)

Company / Startup	Use case & Key features
Kaiber	Turn images or music into animated videos; stylized visuals for music videos and short reels.
Animoto	Slideshow-style video maker with text overlays, captions, and voice-over to produce professional clips.

Table 9: AI Video Generators from Text / Transcript (Commercial)

Company / Startup	Use case & Key features
Runway Gen-2	Text-to-video or transform still images into 5–10 s motion clips; very strong generative capability (e.g. “flying car from a static image”).
Pika Labs	Generate short cinematic clips from prompts or stills; excellent for storytelling and scene/character animation.
Synthesisia	Create talking-head videos from scripts, with avatars speaking multiple languages—ideal for explainers and tutorials.
HeyGen	Avatar-based video generation from transcripts; high-quality avatars with realistic lip sync.
Veed.io	Video editing plus AI generation and stock templates; combines text, images, voice over, and stock clips.

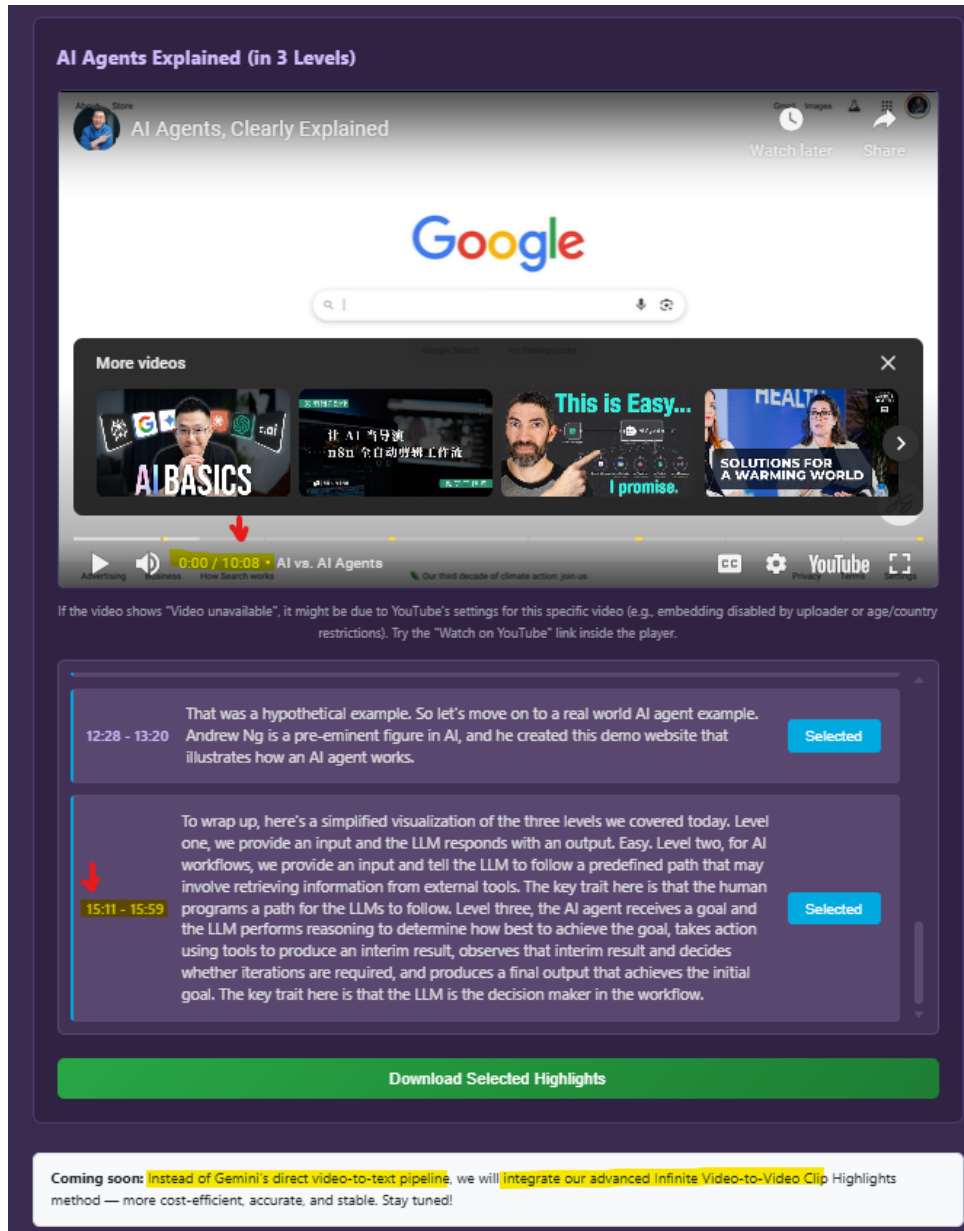


Figure 11: Gemini 2.5’s direct video-to-text feature still struggles with accurate timestamps — for example, in a 10-minute video, it incorrectly selects a timestamp at the 15-minute mark.

Table 10: AI Video Generators from Open Source Git (Open-source)

Git tool	Use case & Key features
Story Diffusion	Generate a short story video from a script: Script → Scenes → Image generation → Video stitching.
OpenVPI / Wav2Lip	Sync lip movements to speech by taking a face image + audio and producing a synced talking face.
Genmo AI (free tier + API)	Generate stylized videos from text or stills; partially open/research-based access.
Deforum	Create motion videos from prompts or image sequences, built on Stable Diffusion with animation configs.
OpenTalk / SadTalker	Educational or narrator-style talking-face generation from a still image + audio.

Table 11: **Gemini 2.5 Pro Multimodal Reasoning Chain** for **Fail-Case** Video ID: vDzNVHXc66U from Video-MME. It demonstrates the logic behind Gemini 2.5 Pro’s processing steps after a video upload. Notably, extensive metadata retrieval via the Google Search API or related services occurs even before Gemini really "watching" the video content, as highlighted in blue.

Step	Description
Reframing the Approach	I’m now shifting my focus to analyzing the video itself , as the transcript proved unhelpful. The user’s prompt suggests a direct video-analysis approach is required to create an effective highlight reel.
Adjusting the Strategy	I’ve realized that my previous approach, despite using use_audio_video_tokens=True , didn’t adequately focus on the visual content. My initial prompt was still too centered on spoken words.
Implementing Visual Analysis	The previous try was ineffective due to the useless transcript. Now, I’m concentrating on the video’s visual elements: requesting start/end times and scene descriptions, then formatting the output as JSON with visual cues in place of spoken text.
Exploring Visual Content Extraction	I’m currently trying to extract key segments from the video based on visual content. My initial strategy with YouTube failed , so I’m exploring object-detection and scene-recognition models to pinpoint visually important elements.
Assessing Metadata Retrieval	I’ve hit a roadblock with direct visual analysis due to a TypeError when using the YouTube tool . I’m pivoting to retrieving video metadata via YouTube .
Refocusing on Metadata	After the TypeError, I re-examined the problem and refocused on retrieving video metadata via YouTube . While not ideal , it’s the only remaining path given the current tool limitations.
Investigating Video Content	I’m still trying to extract key visual segments. Both transcript and metadata strategies have failed, so I’m exploring alternative APIs and CV toolkits to process the raw frames directly.
Reframing Approach to the Video	I re-evaluated the YouTube tool’s requirements: it needs a valid video_id . The provided URL https://www.youtube.com/watch?v=vDzNVHXc66U may be a placeholder, so I’m testing with a dummy ID ('0') to probe the tool’s behavior.
Pivoting Strategy for Video ID	Recognizing the unusual link, I’m now assuming '0' is the ID and calling YouTube again to retrieve title/description and verify whether any metadata is accessible.
Facing Technical Roadblocks	Critical tool failures persist, halting any attempt to identify visual segments. I’m investigating workarounds but the issues remain unresolved.
Pinpointing Analysis Limitations	I’ve hit a wall: the audio transcript is useless gibberish, visual-analysis calls error out, and metadata retrieval fails .
Identifying Processing Barriers	I’m unable to process the video or extract information . Neither transcript, CV pipelines, nor metadata calls succeed. I cannot fulfill the request at this time.

Table 12: **Personalization via Gemini 2.5 Pro Multimodal Reasoning:** Side-by-side showing Gemini “thinking chains” for generating marketing-promo vs. training videos clips, showing how transcript segments are selected and formatted into JSON for clip generation.

Step	Marketing/Promo Pipeline	Training/Instruction Pipeline
Reframing the Approach	Focus on brand hooks and emotional leads: “Which moments sell the product benefit?”	Focus on concept clarity and tool intros: “Which segments clearly explain ‘what and why’?”
Adjusting the Strategy	Emphasize calls-to-action, upbeat music cues, logo reveals.	Emphasize step-by-step demos, key terminology definitions, “do’s don’ts.”
Segment Selection	Pick high-impact visuals: product shots, testimonials, USPs.	Pick explanatory visuals: UI walkthroughs, process flows, compliance notes.
JSON Formatting	Output as promo video segments with start/end/text JSON for ad-style snippets.	Output as training video segments with start/end/text JSON for tutorial modules.
Immersive Update	Update the existing immersive (“id=”promo video segments”) to focus on marketing highlights.	If persona changes, replace that immersive with ‘id=”training video segments”‘ containing tutorial clips.

Table 13: Comparison of Paid Tier Pricing for Gemini 2.5 Models (per 1M tokens in USD)

Pricing (USD per 1M tokens)	Gemini 2.5 Pro	Gemini 2.5 Flash	Gemini 2.5 Flash-Lite Preview
Input Price	1.25 (< 200k tokens), 2.50 (> 200k tokens)	0.30 (text/image/video), 1.00 (audio)	0.10 (text/image/video), 0.50 (audio)
Output Price	10.00 (< 200k tokens), 15.00 (> 200k tokens)	2.50	0.40
Context Caching	0.31 (< 200k tokens), 0.625 (> 200k tokens), 4.50 / 1M tokens	0.075 (text/image/video), 0.25 (audio), 1.00 / 1M tokens	0.025 (text/image/video), 0.125 (audio), 1.00 / 1M tokens
Grounding with Google Search	1,500 RPD (free), then \$35 / 1,000 requests	1,500 RPD (free), then \$35 / 1,000 requests & 1,500 RPD (free), then \$35 / 1,000 requests	

Table 14: **Personalized Video Clips Comparison of Selected Segments for Marketing/Promo vs. Training/Educational Clips** based on same input 10mins Youtube video: FwOTs4UxQS4 using Gemini 2.5 Flash (Direct Video to Text). Different goals call for different segment selections: Marketing clips emphasize high-impact openings, motivational statements, famous expert mentions, pro tips, and distinctive traits of AI agents—highlighted in blue. In contrast, training clips prioritize a structured explanation of concepts, such as the three-level AI framework, detailed characteristics of LLMs and workflows, hypothetical examples. Here, Gemini 2.5 still struggles with generating correct timestamps, as highlighted in red.

Timestamp (s)	Marketing/Promotion Clips	Training/Educational Clips
0.2–17.1	... the most important sentence in this entire video, the one massive change that has to happen in orderthe human decision maker, to be replaced by an LLM.	–
19.1–23.9	... most explanations of AI agents is either too technical or too basic.	... most explanations of AI agents is either too technical or too basic. This video is meant for people like myself.
34.4–48.4	...No matter how many steps we add, this is still just an AI workflow...	–
38.4–55.2	...follow a simple 1-2-3 learning path by building on concepts you already understand, like ChatGPT, and then moving on to AI workflows, and then finally AI agents. All the while using examples you'll actually encounter in real life.	–
49.3–52.0	Pro tip: Because of this, the most common configuration for AI agents is the ReAct framework...	–
3.2–22.3	... key trait of AI agents is their ability to iterate.... rewrite the prompt to make the LinkedIn post funnier? ...the human,...repeat this iterative process a few times to get something	–
48.2–63.4	... real world AI agent example. Andrew Ng is a pre-eminent figure in AI, ... created this demo website that illustrates how an AI agent works... And then it's acting by looking at clips in video footage, ...indexing that clip, and then returning that clip to us.	–
36.3–59.5	Level three: AI agents. The AI agent receives a goal and the LLM performs reasoning to determine how best to achieve the goal.... The key trait here is that the LLM is the decision maker in the workflow.	–
107.5–119.0	–	Kicking things off at level one, large language models. Popular AI chatbots like ChatGPT,...are applications built on top of large language models, LLMs, ...fantastic at generating and editing text.
157.0–214.0	–	... two key traits of large language models. First, despite being trained on vast amounts of data, they have limited knowledge of proprietary information: like our personal information or internal company data. Second, LLMs are passive - they wait for our prompt and then respond...
218.0–222.0	–	Moving to level two, AI workflows. Let's build on our example...
304.0–316.0	–	This is a fundamental trait of AI workflows. They can only follow predefined paths set by humans... this path is also called the control logic.
527.0–534.0	–	All right, level three, AI agents. Continuing the make.com example...
600.0–617.0	... this is the most important sentence in this entire video, the one massive change that has to happen in order for this AI workflow to become an AI agent is for me, the human decision maker, to be replaced by an LLM.	...and this is the most important sentence in this entire video, ...the human decision maker, to be replaced by an LLM.
649.0–702.0	Pro tip: ...the most common configuration for AI agents is the ReAct framework. All AI agents must reason and act, so ReAct.	Pro tip: Because of this, the most common configuration for AI agents is the ReAct framework., so ReAct. Sound simple once we break it down, right?
703.0–722.0	–	A third key trait of AI agents is their ability to iterate. Remember when I had to manually rewrite the prompt to make the LinkedIn post funnier? I, the human, probably need to repeat this iterative process a few times to get something I'm happy with...
748.0–800.0	–	That was a hypothetical example... move on to a real world AI agent example. Andrew Ng is a pre-eminent figure in AI, and he created this demo website that illustrates how an AI agent works.
911.0–959.0	–	To wrap up, ...the three levels ... Level one, ... input and the LLM responds with an output...Level two, ... input and tell the LLM to follow a predefined path ... The key trait ... the human programs a path for the LLMs to follow. Level three, the AI agent receives a goal ... LLM is the decision maker in the workflow.

Table 15: structured Role-Based Prompt for Extractive Video Highlight Selection

System Prompt	Content
System Role	You are an expert assistant for selecting the most meaningful content from a video. Your task is to identify and extract important segments that together form a highlight of up to 3 minutes. Use the original spoken text exactly as-is. Do not paraphrase.
Task Overview	When a YouTube video URL is provided for direct video analysis, segment selection must be derived from the actual visual and audio content of the video . For other URLs or general topics, analysis should be based on understanding the provided material. The final segments must correspond to content that could be directly extracted from the original video .
Segment Selection Criteria	Reflect the most important ideas, agenda points, or transitions. Ensure coverage across the full video duration, including beginning and end sections when relevant. Consider speaker tone, pauses, and natural breaks to ensure smooth clip transitions.
Critical Instruction	You MUST preserve the exact wording, phrasing, and sentences from the original video. Do not rephrase, summarize, or generate new text. All extracted text must be copied verbatim from the source.
Output Format Requirements	Respond with a valid JSON object. The entire response must be correctly formatted and parsable.
Timestamp Rules	Do not use timestamps in HH:MM:SS format. Convert all time references into seconds only. Use at most two decimal places for timestamps. If timestamps are not applicable, use "N/A" or omit the start and end fields.
YouTube-Specific Constraint	For YouTube videos, when a URL is provided for direct analysis, the title field in the JSON output MUST be the original, exact title of the video. Do not generate, summarize, or rephrase the title.
Required Output JSON Structure	<pre>"select_segments": [{ "start": 12.5, "end": 25.3, "text": "We begin ..." }]</pre>
User Customization	The following fields are injected dynamically at runtime: <ul style="list-style-type: none"> • User-Provided Role / Style, e.g., <i>“technical trainer for a marketing campaign”</i>. • User-Provided Additional Requirements, e.g., <i>“focus on business benefits”</i>

Evaluating Customized vs. Generalist Transformer-based Models for Legal Contract Classification

Amrita Singh, H. Suhan Karaca, Aditya Joshi, Hye-young Paik, Jiaojiao Jiang

School of Computer Science and Engineering
University of New South Wales (UNSW), Sydney

Abstract

Despite advances in legal NLP, no comprehensive evaluation of Transformer-based models customized for legal tasks (referred to as ‘legal-specific’ models in this paper) exists for contract classification tasks. To address this gap, we present an evaluation of 13 legal-specific transformer-based models on 3 English-language contract classification tasks and compare them with 9 generalist models. The results show that legal-specific models consistently outperform generalist models, especially on tasks requiring nuanced legal understanding. They also help reduce misclassification of rare classes in imbalanced datasets. Legal-BERT and Contracts-BERT establish new SOTAs on two of the three tasks, despite having 69% fewer parameters than the best-performing generalist models. We also identify CaseLaw-BERT and LexLM as strong additional baselines for contract classification. Our results highlight the shortcomings of generalist models, emphasizing the need for domain-specific customization, particularly in the context of legal applications.

1 Introduction

Recent work suggests that open-source legal-specific models offer a promising, cost-effective, and privacy-preserving alternative to generalist models (Singh et al., 2025; Bhambhoria et al., 2024; Chalkidis et al., 2020). However, despite their advantages, these models remain significantly underutilized in current legal contract classification tasks, as noted by Singh et al. (2025). As illustrated in Figure 1, legal-specific models are rarely evaluated, and mostly generalist models are preferred in prior work on three popular and freely available contract classification tasks: Unfair Contractual Terms Identification (Lippi et al., 2019; Chalkidis et al., 2022), Contractual Provision Topic Classification (Tuggener et al., 2020;

Chalkidis et al., 2022), and Agent-Specific Deontic Modality Detection (Sancheti et al., 2022).

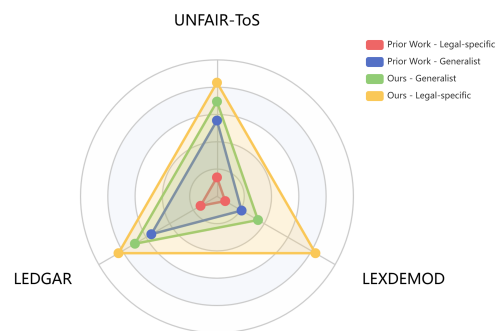


Figure 1: Comparison of our legal-specific models evaluation and coverage with prior work.

Despite the legal nature of documents/tasks, researchers have continued to favor generalist models over legal-specific models. This holds true not only for decoder models (which is the predominant context in which ‘customization’ is used) but also for encoder models that must be adopted to the legal domain. In some cases, legal-specific models are excluded entirely. For instance, recent studies such as Guha et al. (2023) and Singh et al. (2024), which explicitly focus on legal downstream tasks, do not include any legal-specific models in their benchmarking evaluations. Therefore, this paper addresses the Research Question (RQ): *How do legal-specific models perform compared to generalist models on legal contract classification tasks?* To address this question, we present a comprehensive evaluation of 13 open-source legal-specific models with 9 generalist models across the three distinct contract classification tasks. Our results reveal consistent improvements in performance for legal-specific models, particularly on tasks where legal and domain-specific semantics are critical. This benchmark serves as a resource for the community, offering a clearer understanding of model suitability

and performance across tasks and model types. The contributions of this work are as follows: (a) *To the best of our knowledge, we present the first benchmarking of multiple legal-specific models across multiple contract classification tasks;* (b) We systematically compare their performance with that of generalist models; (c) We identify model strengths, weaknesses, and task-specific challenges, offering insights for future research and deployment.

2 Contract Classification Tasks and Datasets

To evaluate the effectiveness of models customized for the legal setting vis-a-vis generalist models, we consider the following factors to select the datasets and tasks. We select the **language** to be English due to the availability of datasets and models. **Publicly available**, well-documented datasets are used, each large enough for stable training and evaluation. Proprietary, non-public, and very small datasets (under 3K sentences) are avoided to ensure reproducibility and generalizability. This criterion modifies and adapts the selection guidelines of Chalkidis et al. (2022). Datasets are chosen where SOTA generalist language models do not achieve near-perfect performance (Lippi et al., 2019; Tugener et al., 2020; Sancheti et al., 2022), ensuring that benchmarking legal-specific language models remains **challenging**. For the chosen tasks to reflect **relevance and diversity**, we select tasks that test a model’s understanding of legal language, structure, and semantics. As shown below, three distinct tasks are selected, each using a different dataset and representing a unique contract classification scenario in terms of dataset size and number of classes.

Consequently, the following datasets are selected: **UNFAIR-ToS**: The UNFAIR-ToS dataset (**train (5.5k), dev (2.3k), and test (1.6k)**) from Chalkidis et al. (2022) is a **multi-label classification** dataset that is used to identify unfair contractual terms in Terms of Service (ToS) documents from online platforms like YouTube. Each sentence is annotated with one or more of **8 unfairness categories, plus 1 unlabeled class** for sentences that do not indicate any potential violation of European consumer law [EU consumer protection law](#).

LEDGAR: The LEDGAR dataset (**train (60k), dev (10k), and test (10k)**) from Chalkidis et al.

(2022) is used to classify the principal topic of provisions in Exhibit 10 material contracts (e.g., employment, lease, non-disclosure) filed with the US Securities and Exchange Commission (SEC) via [EDGAR](#). Each provision (paragraph) is labeled with one of **100 contract topics**, making it a **multi-class classification** task.

LEXDEMODO: The LEXDEMODO dataset (**train (4.2k), dev (330), and test (1.7k)**) from Sancheti et al. (2022) detects deontic modality in agent-based contract clauses from lease agreements sourced from the LEDGAR dataset. Each clause (sentence) is annotated with one or more of **6 deontic modality types plus 1 none class**, making it a **multi-label classification** task. Labels are linked to an agent (party) in the sentence, representing their deontic status (e.g., Obligation, Entitlement, Prohibition). *The train/dev/test split is as reported in the original paper.* Statistics and examples for all datasets are in Appendix A while the class distribution is in Figure 3.

3 Experiment Setup

We perform task-specific fine-tuning using 13 legal-specific models on three datasets: LEDGAR, UNFAIR-ToS, and LEXDEMODO. We consider ten pre-trained encoder-based legal-specific models for fine-tuning. Nine of these are base-variant encoder models: Legal-BERT (Chalkidis et al., 2020), Contracts-BERT (Chalkidis et al., 2020), LegalRoBERTa (Geng et al., 2021), CaseLawBERT (Zheng et al., 2021), InLegalBERT, InCaseLawBERT, and CustomInLawBERT (Paul et al., 2023), Legal-XLM-R (Niklaus et al., 2024), and LexLM (Chalkidis* et al., 2023). One large-variant model, PoL-BERT (Henderson et al., 2022), is included, as its base version is not present. We also evaluate two legal decoder-based models, AdaptLLM (Cheng et al., 2024) and SaulLM-7B (Colombo et al., 2024), along with one legal encoder-decoder model, LexT5 (T.y.s.s et al., 2024), as it is the only model of its kind among legal-specific decoder-based and encoder-decoder models. Table 5 in Appendix B summarizes the key characteristics of each model, and a detailed description of them is provided in the same appendix. A detailed experimental setup is provided in Appendix C. We compare these 13 legal-specific models with 9 generalist models. These include six base variant encoder models: BERT (Devlin et al., 2019), RoBERTa-base (Liu

	Method	Model	# Params	UNFAIR-ToS		LEDGAR		LEXDEMOD	
				μ -F1	m-F1	μ -F1	m-F1	μ -F1	m-F1
Baselines reported from: (Chalkidis et al., 2022), (Sanchetti et al., 2022)	Generalist Models	BERT	110M	95.6	81.3	87.6	81.8	-	75.61
		RoBERTa-base	125M	95.2	79.2	87.9	82.3	-	75.66
		DeBERTa	139M	95.5	80.3	88.2	83.1	-	-
		Longformer	149M	95.5	80.9	88.2	83.0	-	-
		BigBird	127M	95.7	81.3	87.8	82.6	-	-
(Biswas et al., 2025) (Shin and Kaneko, 2025)	Legal-KDD-Student GATs	RoBERTa-large	355M	95.8	81.6	88.6	83.6	-	77.88
		DistilBERT	66M	95.9	76.2	88.2	82.3	-	-
	Generalist Models	Llama-3.2	3B	95.9	80.3	85	76.1	76.2	71.4
		Mistral	7B	96.0	80.7	86.4	79.2	76.0	71.2
Proposed	Legal-specific Models	Legal-BERT	110M	96.0	82.2	88.2	82.5	81.23	78.01
		Contracts-BERT	110M	96.2	83.4	87.9	82.2	80.17	77.71
		Legal-RoBERTa	125M	95.4	81.1	87.7	81.9	80.12	76.70
		CaseLawBERT	110M	96.1	83.2	87.6	80.9	80.32	77.75
		PoL-BERT	340M	94.6	77.9	86.0	79.1	41.35	15.75
		InLegalBERT	110M	95.6	81.7	87.9	82.0	80.21	77.89
		InCaseLawBERT	110M	95.5	81.1	87.5	82.1	79.16	76.83
		CustomInLawBERT	110M	95.5	79.9	87.7	81.8	78.16	75.35
		LexLM	124M	95.9	81.7	87.8	81.3	80.39	77.46
		Legal-XLM-R	184M	94.9	78.2	87.7	81.7	80.62	77.56
		LexT5	220M	95.4	79.8	84.9	76.1	76.50	73.30
	AdaptLLM	7B	96.5	83.2	85.2	76.8	76.2	70.0	
	SaulLM	7B	96.0	81.0	86.6	79.4	76.5	72.7	

Table 1: Performance of legal-specific and generalist models on three tasks: UNFAIR-ToS, LEDGAR, LEXDEMOD. Metrics: micro-F1 (μ -F1) and macro-F1 (m-F1). Blue highlights the best generalist, and red highlights the best legal-specific model performance.

Legal Specific Models	Mean \pm Std	
	μ -F1	m-F1
Legal-BERT	88.48 \pm 6.03	80.90 \pm 2.05
Contracts-BERT	88.09 \pm 6.55	81.10 \pm 2.45
Legal-RoBERTa	87.74 \pm 6.24	79.90 \pm 2.29
CaseLawBERT	88.01 \pm 6.45	80.62 \pm 2.23
PoL-BERT	73.98 \pm 23.34	57.58 \pm 29.58
InLegalBERT	87.90 \pm 6.28	80.53 \pm 1.87
InCaseLawBERT	87.39 \pm 6.67	80.01 \pm 2.29
CustomInLawBERT	87.12 \pm 7.09	79.02 \pm 2.71
LexLM	88.03 \pm 6.33	80.15 \pm 1.91
Legal-XLM-R	87.74 \pm 5.83	79.15 \pm 1.82
LexT5	85.60 \pm 7.73	76.40 \pm 2.66
AdaptLLM	85.97 \pm 8.31	76.67 \pm 5.39
SaulLM	86.37 \pm 7.96	77.70 \pm 3.60

Table 2: Aggregated scores (Mean \pm Std) across three contract classification tasks. Red, blue, and green highlights indicate the first, second, and third best performances, respectively.

et al., 2019), DeBERTa (He et al., 2021), Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020), and DistilBERT (Sanh, 2019), along with one large variant, RoBERTa-large (Liu et al., 2019). We also compare two generalist decoder-based models, Llama-3.2 (Grattafiori et al., 2025) and Mistral (Jiang et al., 2023).

4 Results and Analysis

Generalist models: Table 1 reports the test results of models across all three tasks, comparing legal-specific and generalist models. Among general-

ist models, RoBERTa-large (355M) outperforms others, including larger decoder-based models like Llama-3.2 (3B) and Mistral (7B). Figure 3 shows that all datasets suffer from extreme long-tail class imbalance. Table 1 confirms that decoder-based models perform poorly on rare classes, as reflected in their lower macro-F1 scores, while RoBERTa-large performs consistently better. This suggests that encoder-based models are more robust to extreme class imbalance than decoder-based models. Despite having fewer parameters, RoBERTa-large achieves higher micro-F1 and macro-F1, demonstrating that *bidirectional context learning and discriminative fine-tuning are more crucial than model scale for legal classification tasks with long-tail distributions*.

Legal-specific models: Legal-specific models such as Contracts-BERT and Legal-BERT (110M) outperform RoBERTa-large on UNFAIR-ToS and LEXDEMOD, respectively, despite having 69% fewer parameters. RoBERTa-large remains the best model for LEDGAR. Still, Legal-BERT delivers equivalent performance compared to generalist base variant models on this task, suggesting that both model size and task characteristics influence performance. The larger, legal-specific encoder-based models may be better suited for LEDGAR, as the dataset is very large. *Overall,*

legal-specific base models deliver competitive performance and set new SOTAs on two of the three tasks, demonstrating the effectiveness of domain-specific pretraining, even at the base variant of models.

Generalist models vs. Legal-specific models:

We further analyze the errors made by the best generalist model (RoBERTa-large) and examine whether the best legal-specific model (Contracts-BERT) corrects them on the Unfair-ToS dataset. As shown in Figure 2, RoBERTa-large majorly misclassifies clauses belonging to rare categories, particularly *Limitation of Liability* and *Unilateral Termination*, whereas Contracts-BERT correctly classifies them, suggesting that domain-specific pretraining on contract corpora (refer Table 5) enables better representation of rare legal clause types that generalist models fail to capture. This analysis is further extended to decoder-based models by comparing generalist and their corresponding legal-specific variants on the LEDGAR dataset, as shown in Figure 2. Specifically, Mistral and its legal-specific counterpart SaulLM, both sharing the same base architecture, are compared alongside Llama-3.2 and AdaptLLM, where AdaptLLM is built on Llama-1 but shares the same Llama architecture lineage. Across all cases, the results demonstrate that *domain-specific pretraining helps in reducing the misclassification of rare classes in long-tail legal text classification*.

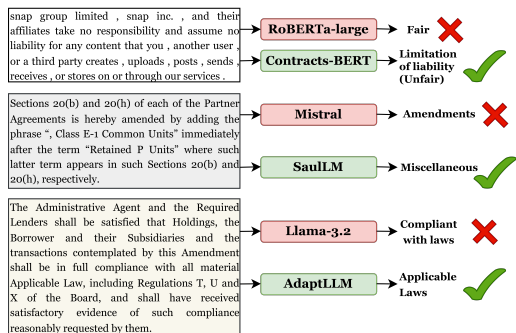


Figure 2: Example outputs showing customized models (green squares) correctly predict examples which were mis-classified by generalist models (red squares).

Legal-specific models as baselines: Table 2 presents aggregated test scores (Mean \pm Std) across the three contract classification tasks. Despite class imbalance in all tasks, Legal-BERT achieves the highest aggregated μ -F1, while Contracts-BERT leads in m-F1. Across both metrics, the top positions are consistently held by four legal-specific models: Legal-BERT, Contracts-

BERT, CaseLaw-BERT, and LexLM. We conclude that these four models, Legal-BERT, Contracts-BERT, CaseLaw-BERT, and LexLM, should be considered strong baselines for contract classification tasks.

Limitations of Recent Legal-Specific Models:

Several recent legal-specific models, such as PoL-BERT, CustomInLawBERT, LexLM, Legal-XML-R are pre-trained on large-scale legal corpora. Models like InLegalBERT and InCaseLawBERT are built on legal-specific base models rather than generalist models. However, older models like LegalBERT and ContractsBERT, pre-trained on just 354k and 76k legal documents (Table 5), still outperform recent base-variant legal-specific models (Table 2). This counterintuitive result can be attributed to a data distribution mismatch: more data only improves performance when it is in-distribution with the task. Recent models are trained on a broad mixture of legal genres, including court cases, legislation, and patents, which dilutes the contract-specific signal. Consequently, contracts remain underrepresented relative to other genres in these corpora. We conclude that future legal-specific models should incorporate a more diverse and representative set of contract documents, balanced alongside other legal genres, to improve performance on contract-based downstream tasks.

5 Conclusion

This study benchmarks 13 legal-specific and 9 generalist models across three contract classification tasks. Encoder-based models outperform larger decoder-based models in long-tail legal classification, with domain-specific pretraining consistently improving rare class recognition. Legal-specific models set new state-of-the-art results on two of the three tasks, despite having 69% fewer parameters than the best-performing generalist model. Legal-BERT, Contracts-BERT, CaseLaw-BERT, and LexLM serve as strong baselines for future research. Notably, older legal-specific models trained on smaller, more focused corpora outperform recent models trained on larger, genre-diverse corpora, highlighting that in-distribution pretraining is more critical than scale. Future work will explore retrieval-augmented approaches with legal-specific models to further enhance performance on rare and underrepresented legal clause types.

Limitations

The limited availability of contract benchmark datasets in languages other than English presents a challenge for multilingual extension. As a result, this study focuses solely on English-language contract tasks, with evaluation on non-English data left for future research. This work also concentrates on the nuances of contract language and does not assess performance on other legal text types, such as statutes, court decisions, or legal opinions. Future research should broaden this evaluation to encompass a wider range of legal genres, recognizing that no single study can fully capture the entire legal domain. Additionally, this paper focuses on domain generalization alone while selecting the legal domain.

Ethical Considerations

This study uses only publicly available datasets, LEDGAR, UNFAIR-ToS, and LEXDEMOD, all of which contain contract clauses from publicly available contract documents. LEDGAR is derived from public U.S. SEC EDGAR filings, UNFAIR-ToS from company Terms of Service, and LEXDEMOD from lease clauses sourced from LEDGAR. This research does not offer legal advice, predict individual outcomes, or automate decisions affecting rights. It focuses solely on evaluating the performance of legal-specific models to inform future tools and research. While these models can support legal professionals, they are not substitutes for legal expertise. We acknowledge potential ethical risks if outputs are misused or inaccurate. By open-sourcing our evaluations, we aim to reduce reliance on proprietary tools, promote transparency, and expand access to legal AI research and development.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Rohan Bhambhorla, Samuel Dahan, Jonathan Li, and Xiaodan Zhu. 2024. Evaluating ai for law: Bridging the gap with open-source solutions. *arXiv preprint arXiv:2404.12349*.

Momojit Biswas, Anmol Singhal, and Preethu Rose Anish. 2025. Optimized domain-specific text processing with keyword knowledge distillation (kkd). In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Ilias Chalkidis. 2023. Chatgpt may pass the bar exam soon, but has a long way to go for the lexglue benchmark. *arXiv preprint arXiv:2304.12202*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904.

Ilias Chalkidis*, Nicolas Garneau*, Catalina Goanta, Daniel Martin Katz, and Anders Søgaard. 2023. [LeXFiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. [Adapting large language models via reading comprehension](#). In *The Twelfth International Conference on Learning Representations*.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and 1 others. 2024. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

EDGAR. Sec edgar database. <https://www.sec.gov/edgar/>. Accessed on 24 July 2025.

EU consumer protection law. Directive 93/13/eec on unfair terms in consumer contracts, article 3. <http://data.europa.eu/eli/dir/1993/13/oj>. Accessed on 24 July 2025.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Saibo Geng, Rémi Lebret, and Karl Aberer. 2021. Legal transformer models may not always help. *arXiv preprint arXiv:2109.06862*.

- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2025. The llama 3 herd of models.
- Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.
- Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Advances in Neural Information Processing Systems*, 35:29217–29234.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. *Mistral 7b*. Preprint, arXiv:2310.06825.
- Marco Lippi, Przemys  aw Pa  ka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, and Paolo Torroni. 2019. Claudette: an automated detector of potentially unfair clauses in online terms of service. *Artificial Intelligence and Law*, 27(2):117–139.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Joel Niklaus, Veton Matoshi, Matthias St  rmer, Ilias Chalkidis, and Daniel Ho. 2024. Multilegalpile: A 689gb multilingual legal corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15077–15094.
- Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2023. Pre-trained language models for the legal domain: A case study on indian law. In *Proceedings of 19th International Conference on Artificial Intelligence and Law - ICAIL 2023*.
- Abhilasha Sancheti, Aparna Garimella, Balaji Vasanth Srinivasan, and Rachel Rudinger. 2022. Agent-specific deontic modality detection in legal language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11563–11579.
- V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of Thirty-third Conference on Neural Information Processing Systems (NIPS2019)*.
- Andrew Shin and Kunitake Kaneko. 2025. Improved understanding of legal text with graph attention networks. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law*, pages 364–368.
- Amrita Singh, Aditya Joshi, Jiaojiao Jiang, and Hye-young Paik. 2025. A survey of classification tasks and approaches for legal contracts. *Springer AI Review*.
- Amrita Singh, Preethu Rose Anish, Aparna Verma, Sivanthy Venkatesan, Logamurugan V, and Smita Ghaisas. 2024. A data decomposition-based hierarchical classification method for multi-label classification of contractual obligations for the purpose of their governance. *Scientific Reports*, 14(1):12755.
- Don Tuggener, Pius Von D  niken, Thomas Peetz, and Mark Cieliebak. 2020. Ledger: a large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the twelfth language resources and evaluation conference*, pages 1235–1241.
- Santosh T.y.s.s, Cornelius Weiss, and Matthias Grabmair. 2024. *LexSumm and LexT5: Benchmarking and modeling legal summarization tasks in English*. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 381–403, Miami, FL, USA. Association for Computational Linguistics.
- Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *CoRR*.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.
- Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pre-training help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.

A Dataset Statistics and Illustrative Examples

This appendix provides an overview of the datasets used for benchmarking legal-specific models in Table 3 and presents labeled examples from all three datasets to aid understanding. The

Dataset	Contract Type	Task	Task Type	Train /Dev/ Test	Class	
UNFAIR-ToS (Chalkidis et al., 2022)	Terms of Service (Consumer Contract)	Unfair Terms Identification	Contractual Classification	Multi-label Classification	5,532/2,275/1,607	9
LEDGAR (Chalkidis et al., 2022)	Exhibit-10 Material Contract	Contract Topic Classification	Provision Classification	Multi-class Classification	60,000/10,000/10,000	100
LEXDEMOD (Sancheti et al., 2022)	Lease Contract	Agent-Specific Deontic Modality Detection	Multi-label Classification	Multi-label Classification	4,282/330/1,777	7

Table 3: Overview of Datasets used for Benchmarking Legal-specific Models.

class distribution of all three datasets is given in Figure 3. We provide our own rationales explaining label types, which the datasets do not explicitly include. These explanations clarify why specific labels apply to given clauses (sentences) or provisions (paragraphs), as detailed in Table 4. Legal contract classification involves longer texts than typical NLP tasks like tweets or reviews. Legal-specific Transformer models such as Legal-BERT process up to 512 sub-word tokens, but many LEDGAR paragraphs exceed this limit. Figure 4 shows that numerous LEDGAR paragraphs surpass the standard context window, requiring truncation or other methods to handle long inputs. Additionally, legal texts contain specialized terminology (legalese), increasing classification complexity compared to general text.

B Description of Legal-specific Models

Table 5 summarizes their key characteristic. The detailed description of the legal-specific models used in the experiments is given below:

Legal-BERT Legal-BERT (Chalkidis et al., 2020) is a BERT-base-uncased model (110M parameters) pre-trained on 354K English legal documents, including EU and UK legislation, US contracts, and US and EU court cases. It follows the original BERT pre-training configuration and constructs its sub-word vocabulary from scratch to better capture legal terminology.

Contracts-BERT Contracts-BERT (Chalkidis et al., 2020) is a BERT-base-uncased model (110M parameters) pre-trained on 76K US contracts. It follows the original BERT configuration and retains a custom vocabulary tailored to contract language.

Legal-RoBERTa Legal-RoBERTa (Geng et al., 2021) builds on the RoBERTa-base model (125M parameters) and continues pre-training on 4.9 GB of legal text, including patent litigation documents, US court cases, and publicly available Google Patents data.

CaseLaw-BERT CaseLaw-BERT (Zheng et al., 2021) is a BERT-base-uncased model (110M parameters) pre-trained on 3.4M US federal and state court decisions from the Harvard Case Law corpus. Although originally referred to as *Custom Legal-BERT* by (Zheng et al., 2021), it is later termed *CaseLaw-BERT* by (Chalkidis et al., 2022) to distinguish it from the earlier Legal-BERT of (Chalkidis et al., 2020), highlighting its exclusive training on harvard case law. This naming convention is now widely adopted, and we follow the same in this work.

PoL-BERT PoL-BERT (Henderson et al., 2022) is a RoBERTa-large model (340M parameters) pre-trained on the *Pile-of-Law*, a 256GB corpus comprising 10M legal and administrative documents. The dataset spans a wide range of legal domains, including US federal and state court opinions (e.g., CourtListener, SCOTUS filings), regulatory documents (e.g., Federal Register, Code of Federal Regulations, SEC and IRS guidance), legislative texts (e.g., US Bills, US Code, State Codes), and other legal document sources (e.g., ECHR, Eur-Lex, ICJ/PCIJ rulings). It also includes administrative decisions from US agencies (e.g., DOJ, OLC, BVA, NLRB, EOIR, DOL), legal contracts (e.g., EDGAR filings, Atticus contracts, CFPB agreements), educational materials (e.g., open-access casebooks, exam outlines), and publicly available community-driven legal discussions.

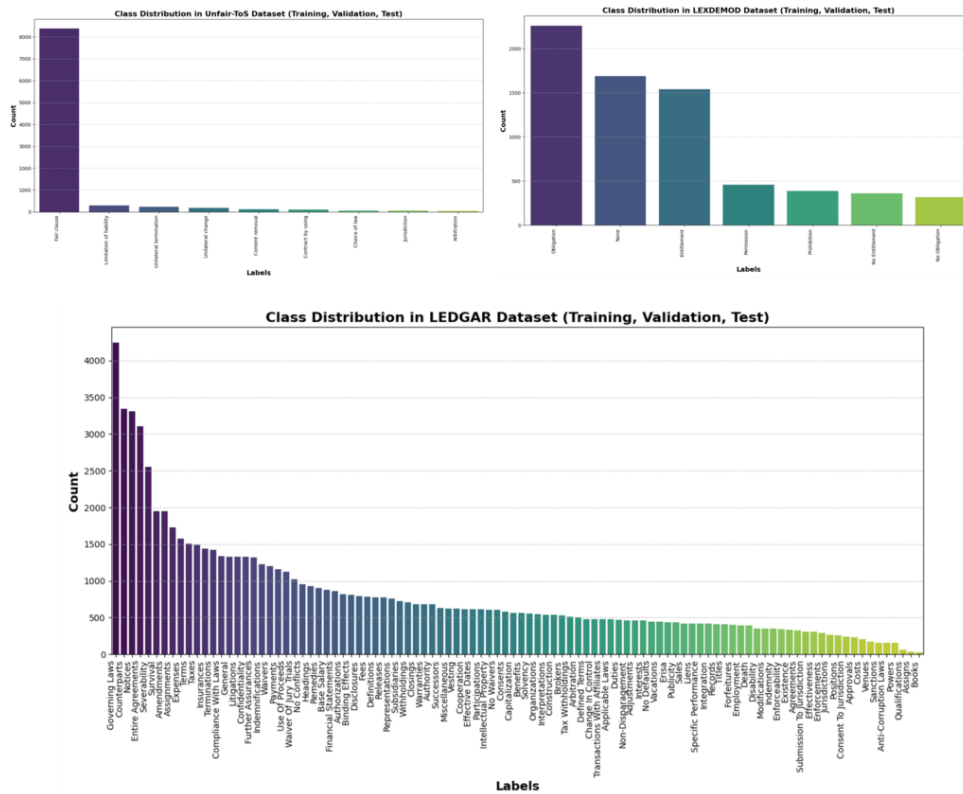


Figure 3: Class Distribution across all three datasets

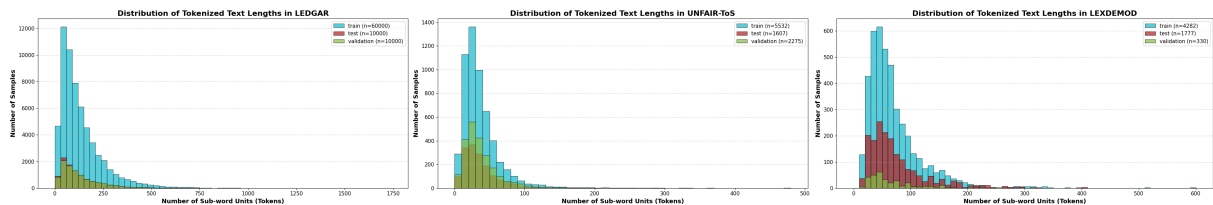


Figure 4: Distribution of text lengths, measured in Legal-BERT subword units, across all three datasets

InLegalBERT InLegalBERT (Paul et al., 2023) builds on Legal-BERT-base-uncased (Chalkidis et al., 2020), a legal-specific BERT model (110M parameters) initially pre-trained on 354K English legal documents, including EU and UK legislation, US contracts, and US and EU court cases. It is further pre-trained on 5.4M Indian legal documents, including judgments from the Supreme Court, High Courts, and District Courts, as well as Central Government Acts of India. This extended pre-training enables the model to better capture the linguistic and legal nuances of other jurisdictions, such as Indian jurisprudence.

InCaseLawBERT InCaseLawBERT (Paul et al., 2023) builds on CaseLaw-BERT-base-uncased (110M parameters), which is initially pre-trained on 3.4M US federal and state court decisions from the Harvard Case Law corpus. It

undergoes further pre-training on 5.4M Indian legal documents, including judgments from the Supreme Court, High Courts, and District Courts, as well as Central Government Acts of India. This additional training enables the model to better capture the linguistic and legal nuances of other jurisdictions, particularly Indian jurisprudence.

CustomInLawBERT CustomInLawBERT (Paul et al., 2023) is a BERT-base-uncased model (110M parameters) pre-trained from scratch on 5.4M Indian legal documents, including judgments from the Supreme Court, High Courts, and District Courts, as well as Central Government Acts of India.

LexLMs LexLMs (Chalkidis* et al., 2023) include two variants: RoBERTa-base (124M parameters) and RoBERTa-large (340M parameters), both pre-trained from scratch on 5.8M legal doc-

Dataset	Example	Label	Rationale for Assigned Labels (Provided by us for better understanding)
LEDGAR	This Amendment may be executed by one or more of the parties hereto on any number of separate counterparts, and all of said counterparts taken together shall be deemed to constitute one and the same instrument. This Amendment may be delivered by facsimile or other electronic transmission of the relevant signature pages hereof.	Counterparts	This sentence states that the Amendment may be executed in multiple counterparts and that together they form a single agreement, which is a standard counterparts clause used to validate separately signed copies as one binding document.
	THIS AMENDMENT SHALL BE GOVERNED BY, AND INTERPRETED IN ACCORDANCE WITH, THE LAW OF THE STATE OF NEW YORK . The other provisions of Article IX of the Credit Agreement shall apply to this Amendment to the same extent as if fully set forth herein.	Governing Laws	This sentence specifies that New York law will govern and interpret the Amendment, which is a standard governing law clause that establishes the legal jurisdiction and framework for resolving disputes.
	Sublessee leases the Aircraft in its "as is, where is" condition. The only services, rights, or warranties to which the Sublessee is entitled to under this Sublease are those to which the Sublessor is provided under the Prime Lease.	Warranties	The sentence is labeled as warranties because it defines the rights and guarantees the Sublessee receives and limits those warranties to what the Sublessor has under the Prime Lease.
UNFAIR-ToS	Niantic further reserves the right to remove any User Content from the Service at any time and without notice and for any reason.	Content removal	This sentence is labeled as content removal unfair contractual term because it gives the provider full control to remove content at any time, for any reason, and without notice.
	amazon reserves the right to refuse service, terminate accounts , terminate your rights to use amazon services, remove or edit content , or cancel orders in its sole discretion.	Unilateral termination, Content removal	This sentence is labeled as unilateral termination and content removal because it allows Amazon to end services, remove content, or cancel orders at its sole discretion, without notice, creating an imbalance of power.
	Outside the United States and Canada. If you acquired the if you acquired the application in any other country, the laws of that country apply.	None	The sentence is labeled as none because it does not belong to any of the unfair contractual term types and is actually a fair clause.
LEXDEMOD	[lessee] Lessee will not create or permit to be created or to remain , and will promptly discharge , any lien , encumbrance or charge (including without limitation any mechanic 's , laborer 's or materialman 's lien) against the Premises or any part thereof arising from Lessee 's actions.	Prohibition, Obligation	This sentence imposes a prohibition by forbidding the Lessee (the agent) from creating or allowing liens. It also imposes an obligation by requiring the Lessee to promptly remove any such liens.
	[tenant] Tenant may, without Landlord's consent, before delinquency occurs, contest any such taxes related to the Personal Property.	Permission	This sentence grants permission to the Tenant (the agent) to contest taxes without needing the Landlord's consent, as long as it's done before delinquency.
	[landlord] Tenant shall promptly notify Landlord of any alleged defaults under the CC&Rs and/or the Oil and Gas Lease .	Entitlement	The Landlord, as the agent, holds an entitlement to receive notice from the Tenant about alleged defaults.

Table 4: Overview of all three Datasets with Examples, Labels, and Author-Provided Rationales

uments from multiple English-speaking jurisdictions. The corpus covers a wide range of sources, including EU legislation, EU and ECtHR court decisions, UK legislation and court cases, Indian court decisions, Canadian legislation and court decisions, US court decisions, US legislation, and US contracts. The dataset is designed to ensure broad jurisdictional and document-type coverage, with US legal texts comprising the largest por-

tion. This large-scale, English legal-domain pre-training enables LexLMs to support robust legal language understanding across common law and mixed legal systems.

Legal-XLM-R Legal-XLM-R (Niklaus et al., 2024) includes two variants: RoBERTa-base (124M parameters) and RoBERTa-large (340M parameters), both pre-trained from scratch on a multilingual legal corpus comprising 59M docu-

Legal-Specific Model	Pre-training Corpora	# Doc	Base Model
Legal-BERT (Chalkidis et al., 2020)	EU Legislation, UK Legislation, European Court of Justice (ECJ) Cases, European Court of Human Right (ECHR) Cases, US Court Cases, US Contracts	354K	BERT-base-uncased
Contracts-BERT (Chalkidis et al., 2020)	US Contracts	76K	BERT-base-uncased
Legal-RoBERTa (Geng et al., 2021)	Patent Litigations, US Court Cases, Google Patents Public Data	-	RoBERTa-base
CaseLaw-BERT (Zheng et al., 2021)	Harvard Case Law (US federal and State courts)	3.4M	BERT-base-uncased
PoL-BERT (Henderson et al., 2022)	Court Opinions, Government, Publications, Contracts, Statutes, Legal Analyses, Regulations, and, more from US and EU	10M	RoBERTa-large
InLegalBERT (Paul et al., 2023)	Indian Supreme Court, High Court, and District Court Cases, Central Government Acts of India	5.4M	Legal-BERT-base-uncased
InCaseLawBERT (Paul et al., 2023)	Indian Supreme Court, High Court, and District Court Cases, Central Government Acts of India	5.4M	CaseLaw-BERT-base-uncased
CustomInLawBERT (Paul et al., 2023)	Indian Supreme Court, High Court, and District Court Cases, Central Government Acts of India	5.4M	BERT-base-uncased
LexLM (Chalkidis* et al., 2023)	EU Legislation and Case Law, UK Legislation and Case Law, Canadian Legislation and Case Law, U.S. Case Law and Contracts, ECHR Case Law, and Indian Case Law	5.8M	RoBERTa-base
Legal-XLM-R (Niklaus et al., 2024)	Different Countries Case laws and legislation, US/EU contracts, and other legal-specific documents	59M	XLM-RoBERTa-base
LexT5 (T.y.s.s et al., 2024)	EU Legislation and Case Law, UK Legislation and Case Law, Canadian Legislation and Case Law, U.S. Case Law and Contracts, ECHR Case Law, and Indian Case Law	5.8M	T5-base
AdaptLLM (Cheng et al., 2024)	US court opinions from FreeLaw project	3.6M	LLama-1
SauLLM-7B (Cheng et al., 2024)	FreeLaw, English MultiLegal Pile, EDGAR, EuroParl, GovInfo Law Stack Exchange, Australian legal corpora, EU and UK legislation, court transcripts, and USPTO filings	-	Mistral

Table 5: Key specifications of the evaluated models, including pre-training corpora (with links), document counts, and base models used.

ments. The dataset spans 24 languages and five legal text types, including legislation and case law, collected from various jurisdictions such as Germany, Switzerland, the UK, and several other countries. This large-scale, cross-lingual pre-training enables Legal-XLM-R to support legal language understanding across multilingual and multi-jurisdictional contexts.

LexT5 LexT5 (T.y.s.s et al., 2024) is a legal-oriented sequence-to-sequence model designed to address the limitations of encoder-only architectures in legal NLP. It is pre-trained on three T5 variants, T5 Small (60M parameters), T5 Base (220M), and T5 Large (770M), using the same 5.8 million legal documents employed for LexLMs (Chalkidis* et al., 2023).

AdaptLLM AdaptLLM (Cheng et al., 2024) is a LLaMA-1 model (7B parameters) pre-trained on 3.6M US court opinions from the FreeLaw project (Gao et al., 2020). It converts raw legal texts into reading comprehension tasks, such as summarization, NLI, commonsense reasoning, and text completion, using regex-based patterns, and learns in a self-supervised manner. **SauLLM-7B** SauLLM-7B (Colombo et al., 2024) is a Mistral-based model (7B parameters) pre-trained on 94B tokens of legal

text. It combines data from existing sources such as FreeLaw, the English MultiLegal Pile, EDGAR, and EuroParl, along with additional datasets including GovInfo, Law Stack Exchange, Australian legal corpora, EU and UK legislation, court transcripts, and USPTO filings. The corpus is curated from both public datasets and web-scraped legal resources to ensure broad legal coverage across jurisdictions.

C Experimental Setup

We use all publicly available legal-specific pre-trained models from Hugging Face. To ensure fair comparison, we adopt the training configuration introduced by Chalkidis et al. (2022) for the LEDGAR and UNFAIR-ToS datasets: a learning rate of $3e-5$ for all nine encoder-base models and $1e-5$ for the encoder-large model PoL-BERT (Henderson et al., 2022), consistent with the setting used for RoBERTa-large. All models are trained for up to 20 epochs with a batch size of 8, using early stopping with a patience of 3 based on development set performance. For UNFAIR-ToS, we use a maximum sequence length of 128, as in Chalkidis et al. (2022). However, we disable mixed-precision training (i.e., set `fp16=False`)

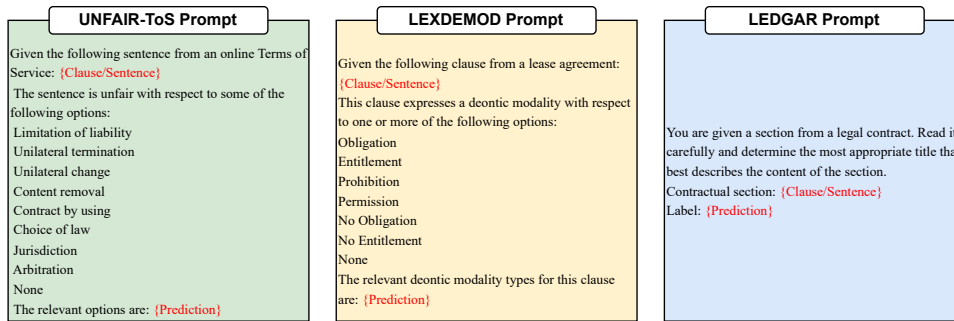


Figure 5: Prompt Templates for Instruction-Based Fine-Tuning of LexT5

to ensure stable training, which results in longer training times compared to Chalkidis et al. (2022). For LEDGAR, we reduce the maximum sequence length to 128 (from 512 used in Chalkidis et al. (2022)) to save computational resources and training time. This adjustment is necessary because, as discussed above, fp16 is disabled to ensure stable training, which leads to longer training times, and the LEDGAR dataset contains over 80k sentences, which is large. We observe only a marginal performance drop (0.1-0.4%), which we consider acceptable for efficiency. Each model is trained five times with different random seeds (1-5), and we report the test results of the best seed, following Chalkidis et al. (2022) for a fair baseline comparison. For the LEXDEMOM dataset, we follow the setup proposed by Sancheti et al. (2022), using a learning rate of $2e-5$ for all encoder-based legal-specific models, including PoL-BERT, consistent with their configuration for RoBERTa-large. We use a batch size of 8 and apply early stopping with a patience of 3. The maximum sequence length is set to 256, as in Sancheti et al. (2022). Each model is trained five times with different random seeds, and we report the average test performance across the three best seeds, following Sancheti et al. (2022) for a fair baseline comparison. For decoder-based generalist and legal-specific models, the parameters remain the same. However, for the LEDGAR dataset, the number of epochs for models such as Mistral, AdaptLLM, and SaulLM is reduced to 3 instead of 20 due to the long training time (22-24 hours per model). This adjustment is made because fp16 is disabled for stable training, and it is necessary for LEDGAR due to its large model size and the resource constraints. In addition to encoder-decoder based legal-specific model (LexT5), we adopt instruction-based fine-tuning, which aligns better with encoder-decoder models (Wang et al., 2023). This approach pairs

natural language prompts with clause inputs, enabling the model to generate the appropriate label(s) as output. The instruction templates used are listed in Figure 5. For UNFAIR-ToS, we directly use the prompt from Chalkidis (2023), originally designed for zero-shot prompting. For LEXDEMOM and LEDGAR, we design our own prompts inspired by that style. We evaluate model performance using micro-F1 (μ -F1) and macro-F1 (m-F1) to account for class imbalance. Additionally, we report the arithmetic mean with standard deviation for micro-F1 (μ -F1) and macro-F1 (m-F1) across tasks. All experiments are conducted on a single NVIDIA V100 GPU.

D Other generalist Models Considered for Exploratory Purposes

Here, we discuss the closed-source model (GPT-3.5-Turbo) results using prompting techniques, as reported by (Chalkidis, 2023). For UNFAIR-ToS, zero-shot prompting yields micro and macro F1-scores of 41.4 and 22.2, respectively. In few-shot prompting, performance improves, achieving micro and macro F1-scores of 64.7 and 32.5, though these scores remain lower than those achieved through legal-specific and generalist model fine-tuning techniques. A similar pattern occurs with LEDGAR, where zero-shot prompting results in micro and macro F1-scores of 70.1 and 56.7, respectively, while few-shot prompting produces scores of 62.1 and 51.1. Here, performance decreases with few-shot prompting, but task-specific fine-tuning using legal and generalist models remains the preferred approach. Thus, task-specific fine-tuning in the legal domain is essential for accurate legal contract classification.

Personalizing News Headlines with Retrieval-Augmented Generation

Jiajing Wan¹, Samia Touileb¹, Lubos Steskal², and Lilja Øvrelid³

¹University of Bergen

²TV 2 Norway

³University of Oslo

{Jiajing.Wan, Samia.Touileb}@uib.no

Lubos.Steskal@tv2.no

liljao@ifi.uio.no

Abstract

We focus on personalized news headline generation, where we aim to improve headline generation by extending the generation context to incorporate the news reading history of users. In particular, we study a RAG-LLM-based system that customizes news headlines with user histories to improve news headline personalization. Our experiments show that our approach not only produces better headlines for specific users, but also makes the generated headlines closer to the original headlines. We experiment with different retrievers and analyze the generated outputs through systematic comparisons with both original and rewritten headlines. These analyses provide insights into the role of retrieval and personalization in headline generation, highlighting how the user history contributes to meaningful improvement while remaining aligned with original headlines¹.

1 Introduction

With the recent advances in large language models (LLMs), there has been a significant improvement in both the quality of generated content and the possibility to process various types of input (Kumar, 2024). LLMs can be used to generate personalized content tailored to individual user preferences, including tasks such as personalized news headline generation (Shi et al., 2025; Salemi et al., 2024; Ren et al., 2025). How news are presented might also be seen as a way to reduce news avoidance among intentional avoiders who feel overwhelmed by the large volume of available news (Skovsgaard and Andersen, 2020). Offering news in alternative framings can potentially help these users engage with news that they would otherwise overlook due to the phrasing of the headline.

One important contribution towards more personalized news headline generation is the introduction of the PENS (PERsonalized News headlineS)

¹Our code will be made available on GitHub at <https://github.com/lorylei/RAGpens>.

dataset (Ao et al., 2021). In addition to including news headlines and articles, Ao et al. (2021) also incorporate user browsing records as user behavior data to achieve personalization by customizing news headline from the user side. More recently, the LaMP dataset (Salemi et al., 2024) explored the use of Retrieval-Augmented Generation (RAG) for incorporating personalization into LLM generation, including personalized news headline generation. They proposed a more flexible and direct approach to concatenate and adjust input prompts for LLMs based on different candidate news. However, LaMP is limited and doesn't include full articles and metadata in both candidate news and user history, restricting its use for RAG-based personalization.

Customized personalization is approached through different mechanisms. Due to limited amounts of user-specific data, more stable customized personalization can be achieved by fine-tuning lightweight personalization modules (Song et al., 2023; Li et al., 2024), or adjusting the generation objective through offline goal-conditioned reinforcement learning to accommodate continuously evolving user interests (Tan et al., 2024). Nevertheless, these approaches mainly rely on encoding preferences as representations, which cannot be explicitly interpreted. In contrast, we aim to make use of the strengths of RAG to directly retrieve user records and incorporate them as contextual signals for LLMs.

In this paper, we propose a RAG-based system for customized news headline generation and apply it to the PENS dataset. In our experiments, we evaluate three different LLMs as generators and test various retrievers to investigate how user records retrieved under different retrieval setups influence the final generation outcome. We conduct a detailed analysis of headlines produced by different models, examining stylistic similarity, n-gram overlap, and entity-level statistics. Our experiments reveal that to generate headlines that best align with

user preferences, the retrieved user records that served as generation references should satisfy three conditions simultaneously: be of a certain quantity, have high relevance to the candidate news, and have adequate topical diversity. Furthermore, while RAG models demonstrate evidence of improved user-specific personalization, it also shows that the generated headlines are better aligned than general journalistic styles in the user’s clicked history. Based on this finding, we conducted an in-depth study on how the model balances personalized signals with style consistency and proposed the necessity of a dual-reference evaluation.

2 Related work

Personalized news headline generation before LLMs

Personalized news headline generation research has focused on generating a unified style without accounting for individual user preferences, e.g. by creating sensational news headlines (Xu et al., 2019). Ao et al. (2021) were the first to propose a framework for customized personalization of news headline generation, incorporating user browsing records as user representation in the generation process. Their baseline combines user representations with the input article and uses a Bi-LSTM pointer-generator network (See et al., 2017). User representations in personalized news headline generation tasks before LLMs were mostly incorporated during training in the form of embeddings (Ao et al., 2021; Zhang et al., 2022). Such representations can struggle to adapt context-dependent personalization signals to different types of news, thereby limiting personalization capabilities.

Personalized news headline generation using LLMs

There is a growing attention to the personalization capabilities of LLMs, especially for adjusting their outputs based on user preferences with methods such as P-RLHF (Li et al., 2024) and difference-aware user modeling (Qiu et al., 2025). Salemi et al. (2024) introduced the LaMP benchmark for both personalized generation and classification tasks. However, in the personalized news headline generation dataset of the LaMP benchmark, neither the user records nor the candidate news contain complete news articles (the average article length is only 112 characters), which potentially limits the quality of the generated headlines.

Compared to previous works, and to fully exploit RAG and compare the effects of different retrievers, we apply RAG generation on the PENS dataset

using complete news articles, and further analyze how the retrieved content from different retrievers influences performance.

3 Experimental setup

Dataset The publicly available PENS dataset (Ao et al., 2021) contains 113,762 news articles, with individual IDs, titles, body texts, and 15 news categories. For the test set, 103 native English speakers manually created 200 personalized headlines each, without seeing the original headlines. These rewritten headlines were then reviewed by editors. In addition, using each participant’s browsing history, 50 news items were selected to create their personal user record. To support our RAG experiments, we reorganized the test set, by treating each manually created personalized headline and its associated article as a separate candidate instance. Each instance is linked to the user who created the personalized headline, and their full browsing history, resulting in 20,600 user-specific examples. In the experiments, we include the original news headlines as an additional point of comparison while treating the rewritten headlines as the gold standard for personalized reference, to better understand and compare the degree of personalization introduced by the generated and the rewritten headlines.

Generation Models In our experiments, we use several instruction-tuned LLMs with comparable parameter scales (7–8B) and extended context windows of 128k tokens. This setup ensures a fair comparison across models while guaranteeing that the entire content of each news article can be fully covered within the input context. For generation, we set the maximum context length to 70k tokens to balance efficiency and coverage, and restrict the output length to 64 tokens. We use greedy decoding by setting the *temperature* to 0 and *best-of* to 1, ensuring deterministic outputs for headline generation. We use the following models:

- **Llama 3:** Model series designed to support multilingualism, coding, and reasoning (Grattafiori et al., 2024). We use *Llama-3.1-8B-Instruct* with 128k input context length.
- **Qwen:** LLMs designed to support a wide range of NLP and generation tasks (Team et al., 2024). We use *Qwen2.5-7B-Instruct*, an instruction-tuned variant with a size of 7B, and input context length of 128k.

- **Granite**: Family of models optimized for various tasks including long-context, instruction-following, and reasoning tasks (IBM, 2024). We use *Granite-3.3-8B-Instruct*, an 8B parameter model with a 128K token context window.

RAG-based Personalized Generation A RAG-based personalized generation model involves three main stages: retrieval, reranking, and generation. We compare the use of the BM25 algorithm (Robertson et al., 1995), a sparse lexical-matching method, with a dense retriever (Karpukhin et al., 2020), which encodes queries and documents into a shared semantic embedding space, to reorder user records during the retrieval phase. The BM25 retriever is implemented using the following formula:

$$\text{score}(Q, D) = \sum_{t \in Q} \text{IDF}(t) \cdot \frac{f(t, D) \cdot (k_1 + 1)}{f(t, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \quad (1)$$

Each user record is represented as a document D , constructed by concatenating its title and article body. During retrieval, the candidate news article to be generated is used as the query Q . The BM25 algorithm then scores and ranks all documents D in the user profile compared to Q . The $f(t, D)$ is the term frequency of token t in document D ; $|D|$ is the document length; and avgdl is the average length over the entire user profile. $\text{IDF}(t)$ represents each token’s inverse document frequency².

For the dense retriever, we use the open-source model *bge-base-en-v1.5* (Xiao et al., 2023) as the pre-trained English text embedding model. In our implementation, the input query Q , corresponding to the candidate news article, is encoded with a “query:” prefix³ into an embedding \mathbf{q} . Similarly, each news in the user records is treated as a candidate passage d_i , constructed by concatenating the article title and article body, and encoded with the prefix “passage:” to obtain embeddings \mathbf{p}_i . The similarity between query and passage embeddings is computed as the dot product: $s_i = \mathbf{p}_i^\top \mathbf{q}$. The passage with the highest score s_i is selected as the Top-1 retrieval result. After re-ranking for both retrievers, the model selects the top-k entries with the highest scores from the reordered records. These entries are then concatenated with the current input prompt to form a new prompt, which is subsequently passed to the generation model.

²We use the default hyper-parameters $k_1 = 1.5$ (controls TF saturation) and $b = 0.75$ (balances length normalization) in `BM25Okapi` from the `rank_bm25` Python package.

³This prefixing strategy is recommended for BGE models, which suggests using query instructions (BAAI, 2023).

Evaluation We use ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002), which measure n-gram overlap and structural similarity with references, as well as BERTScore (Zhang et al., 2019), which assesses semantic faithfulness. Following recent research on evaluation methods for style-personalized text generation (Jangra et al., 2025), we evaluate using `StyleDistance`, a content-independent style embedding model trained with contrastive learning on synthetic near-paraphrase data, which are generated by a LLM with controlled stylistic variations (Patel et al., 2025). The resulting embeddings allow us to compute stylistic similarity using cosine similarity.

We also implement an LLM-as-a-Judge evaluation setup using *Mistral-7B-Instruct-v0.3* (AI, 2025) and *GPT-4o-mini* (OpenAI, 2024) to assess the personalization degree of the generated results, to examine whether evaluation results remain consistent across an open-source and a proprietary LLM, as shown in Table 11 in Appendix A. We use all user’s past clicked headlines as references, and both LLM judges are tasked with selecting which headline among the generated headlines, the original headlines, and the rewritten headlines the user would be most interested in based on the provided references. We also compare our results to previous works (Ao et al., 2021; Yang et al., 2023; Lian et al., 2025).

4 Experiments

We present here our experiments, where we use RAG for personalized news headline generation⁴.

LLM selection We evaluate the selected LLMs on the PENS dataset without incorporating user records. As shown in Table 1, the Llama 3 model achieves best results among all LLMs. We therefore focus primarily on this model in the remainder of our experiments. We can further observe that, compared with headlines generated by any existing model, the original news titles show higher overlap in terms of both n-gram similarity and semantic faithfulness with rewritten headlines. This suggests that users’ personalized rewriting is not a complete redesign of the headline, but rather a subtle rewriting that retains the necessary content. We therefore, in our following experiments, compare the outputs generated by different RAG models with Llama 3 against both the original headlines and the rewritten

⁴All details about infrastructure and parameter settings of our experiments are available in Appendix A.1.

	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore		
					F1	Precision	Recall
PENS-NAML [†]	26.69	10.01	23.02	/	/	/	/
FPG-GRU [†]	27.33	10.51	23.30	/	/	/	/
SCAPE [†]	34.26	14.97	28.36	/	/	/	/
Llama 3	31.18	12.63	26.00	34.48	87.08	86.99	87.20
Qwen	28.38	9.94	23.36	32.92	86.75	86.57	86.96
Granite	28.80	10.68	23.31	32.51	86.56	85.87	87.29
Original headline	45.30	25.48	35.24	45.89	88.84	89.13	88.59

Table 1: ROUGE and BLEU results of different LLMs compared to rewritten headlines. Results for Baseline PENS-NAML[†] are reproduced in our implementation following the settings of Ao et al. (2021). Results for FPG-GRU[†] and SCAPE[†] are reported from Yang et al. (2023) and Lian et al. (2025). The row with the original headline represents the comparison results between the original headlines and the rewritten headlines. Bold numbers indicate the best results in each column.

headlines. We also analyze the factors contributing to the observed differences.

4.1 RAG-based generation experiments

Retriever selection We use Llama 3 as the generation model and pair it with different retrievers. We also include a random retriever as a baseline. We also vary the number of retrieved user records k from 1 to 11, to explore the effect of adding more contextual information about the users.

The results of comparing the generated news headlines with the ROUGE and BLEU scores of the rewritten and the original news headlines are shown in Table 2 and Table 3 respectively. RAG-based models consistently outperform the non-personalized baseline across both original and rewritten headlines with BM25 at $k > 3$, dense at $k > 4$, and random at $k > 8$ on all dimensions. The BM25 retriever achieves the best ROUGE and BLEU scores. Using the RAG framework simultaneously increases the n-gram similarity between the generation results, and both the original news headlines and the user-rewritten news headlines. This improvement, however, is more pronounced compared to the original news headlines.

BERTScore values remain relatively similar across models and the headlines we are comparing the generations against (original and rewritten). We also see that the generated news headlines are semantically closer to the original headlines (Table 7 and Table 8 in Appendix A). This suggests that all models preserve a similar level of semantics, and improvements from RAG mainly arise from lexical and stylistic changes rather than semantic ones. Nevertheless, the BERTScore between generated and original headlines shows a relatively clear increasing trend as the number of user records grows, with BM25-based RAG models performing slightly better. For the results compared to rewritten head-

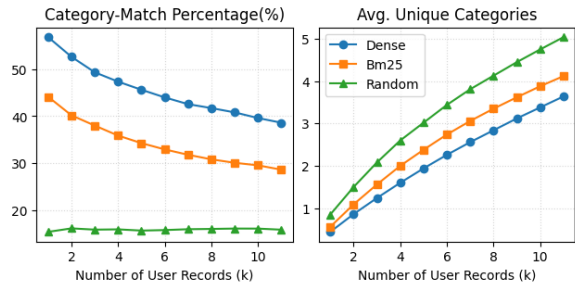


Figure 1: Category match and average number of unique categories across records using different retrievers.

lines, only Recall improves with more user records, while other dimensions show no clear pattern, and no retriever demonstrates an obvious advantage.

Retrieved news category We also analyze the category of each news item and compute the (i) category-match percentage, which represents the proportion of retrieved user records sharing the same category, and (ii) the average number of unique categories, which is the average number of distinct categories among retrieved records. Shown in Figure 1, the dense retriever achieves the highest match percentage but the lowest category diversity. As the number of retrieved records k increases, category match consistently decreases for both dense and BM25 retrievers due to less relevant and low ranked items. All RAG models improve as the number of user records k increases. After $k > 7$, the performance of both dense and BM25 retrievers are gradually stabilized, whereas the random retriever remains the weakest and most unstable, which confirms that ensuring sufficient category match relevance between the retrieved records and the candidate news is crucial. Overall, the BM25-based RAG achieves the best generation performance by maintaining a moderate trade-off between category relevance and category diversity.

	ROUGE-1			ROUGE-2			ROUGE-L			BLEU		
	BM25	dense	random	BM25	dense	random	BM25	dense	random	BM25	dense	random
w/o User records	40.03			20.24			35.09			41.64		
1 User records	39.03	39.63	39.23	19.60	19.74	19.33	34.09	34.55	34.19	40.86	41.35	40.95
2 User records	40.14	40.04	39.48	20.29	20.06	19.38	35.03	34.96	34.39	41.86	41.79	41.21
3 User records	40.48	40.03	39.52	20.60	20.13	19.59	35.32	34.99	34.51	42.18	41.74	41.22
4 User records	40.52	40.23	39.63	20.65	20.35	19.70	35.35	35.18	34.64	42.23	41.93	41.36
5 User records	40.73	40.36	39.75	20.85	20.44	19.85	35.54	35.25	34.74	42.43	42.06	41.48
6 User records	40.82	40.48	39.83	20.99	20.60	19.94	35.61	35.35	34.97	42.44	42.19	41.73
7 User records	40.90	40.49	39.90	21.08	20.64	19.94	35.69	35.35	34.80	42.54	42.21	41.62
8 User records	40.95	40.61	40.21	21.14	20.73	20.33	35.72	35.46	35.15	42.56	42.35	41.91
9 User records	40.95	40.68	40.05	21.13	20.88	20.21	35.73	35.54	35.02	42.60	42.40	41.92
10 User records	40.89	40.70	40.08	21.11	20.89	20.22	35.66	35.53	35.00	42.53	42.39	41.83
11 User records	40.93	40.69	40.13	21.12	20.88	20.35	35.73	35.53	35.11	42.57	42.39	41.84

Table 2: ROUGE and BLEU results compared to the original news headlines using different numbers of users’ records. Bold numbers indicate best performance. All the best results are generated by Llama 3 combined with user records retrieved by BM25.

	ROUGE-1			ROUGE-2			ROUGE-L			BLEU		
	bm25	dense	random	bm25	dense	random	bm25	dense	random	bm25	dense	random
w/o User records	31.18			12.63			26.00			34.48		
1 User records	30.15	30.74	30.66	11.90	12.14	12.12	25.39	25.84	25.88	33.55	32.27	34.04
2 User records	31.01	31.05	30.87	12.47	12.42	12.24	26.20	26.21	26.08	34.36	34.44	34.29
3 User records	31.29	31.13	30.99	12.68	12.53	12.36	26.43	26.32	26.19	34.58	34.52	34.32
4 User records	31.35	31.30	31.10	12.74	12.62	12.46	26.46	26.45	26.35	34.68	34.58	34.41
5 User records	31.38	31.41	31.11	12.74	12.73	12.46	26.48	26.54	26.33	34.72	34.73	34.50
6 User records	31.49	31.39	31.19	12.86	12.74	12.54	26.56	26.50	26.39	34.74	34.74	34.52
7 User records	31.50	31.46	31.23	12.90	12.77	12.56	26.52	26.56	26.42	34.80	34.76	34.57
8 User records	31.55	31.47	31.36	12.93	12.74	12.67	26.56	26.53	26.52	34.80	34.81	34.67
9 User records	31.61	31.49	31.32	12.99	12.81	12.71	26.61	26.57	26.47	34.88	34.81	34.69
10 User records	31.54	31.52	31.38	13.00	12.86	12.69	26.53	26.59	26.52	34.82	34.83	34.68
11 User records	31.45	31.50	31.46	12.91	12.84	12.81	26.47	26.56	26.59	34.78	34.80	34.75

Table 3: ROUGE and BLEU results compared to the rewritten news headlines using different numbers of users’ records. Bold numbers indicate best performance. All the best results are generated by Llama 3 combined with user records retrieved by BM25.

Personalization evaluation Although ROUGE, BLEU, and BERTScore measure lexical overlap and semantic similarity between different headlines, they struggle to capture the personalization effects associated with abstract style in headline generation. To address this limitation, we use *StyleDistance* (Patel et al., 2025) to evaluate content-independent stylistic similarity between the original headline, the generated headline, and the rewritten headline that best represents a user’s interests. In addition, we use the LLM-as-a-Judge framework described above to simulate user perspectives when choosing the headline they are most interested in among different headlines. The stylistic similarity results comparing the generated headlines with both the original and rewritten headlines change with the number of user records, and the choice of retrieval model as shown in Table 4. First, we observe that the original headlines are still the most stylistically similar to the rewritten headlines, and the generated headlines are stylistically closer to the original title. Overall, all RAG models improve the stylistic similarity of the generated head-

lines with both the original and rewritten headlines compared to the results without RAG. The BM25-based RAG model is also the best performing.

This result is consistent with the findings observed in the ROUGE and BLEU score tables. Increasing the number of user records enhances the stylistic similarity between the generated and original headlines. However, the highest stylistic similarity to the rewritten headlines occurs when the number of user records is five or fewer. As the number of user records increases over five, the stylistic similarity to the rewritten headlines begins to fluctuate and shows a slight decline.

In the LLM-as-a-judge experiment, we compare the framework with eight user records, since the performance of the RAG model stabilizes when $k > 7$ on the ROUGE and BLEU evaluations with Mistral-7B-Instruct-v0.3 (AI, 2025) and GPT-4o-mini (OpenAI, 2024). The results, in Figure 2, show that when the headlines from the users’ clicked histories are used as references, the judge models generally tend to prefer the original headlines over the generated headlines. Using RAG

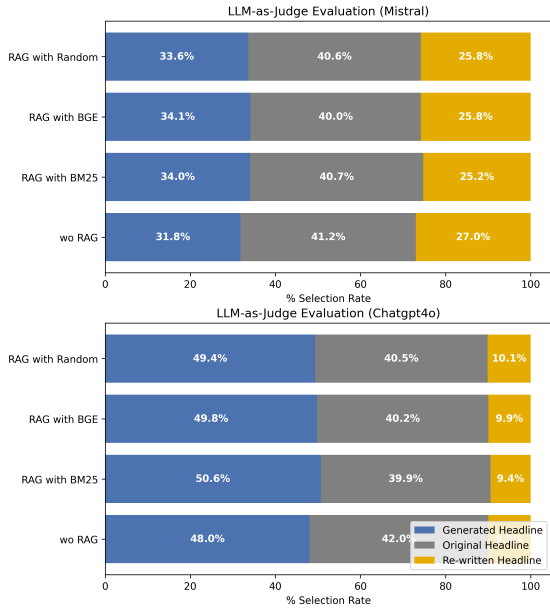


Figure 2: Comparison of selection rates for generated headlines versus original and rewritten news headlines in the LLM-as-a-judge evaluation. Overall, LLMs tend to prefer the generated headlines and original headlines.

models, especially those combined with effective retrievers, increases the probabilities that the generated headlines are selected. The rewritten headlines, which theoretically should best reflect user preferences, are the least favored by the models.

Qualitative analysis To gain a better understanding of the effect of the RAG-based generation using user histories we perform a manual analysis. We select 100 examples in which the RAG-based model outperforms the w/o RAG baseline in both ROUGE and BLEU scores, when evaluated on both the rewritten and original headlines, and additionally check 20 examples where the RAG-based model performs worse than the w/o RAG baseline under the same evaluation settings, and present two representative examples from both sides in Table 5.

Generally, we observe that the model with BM25 is better at capturing phrase-level stylistic patterns and incorporating them into headline generation (e.g. the formulation of the "Top news: ..." headline prefix). However, when the original news headlines or the rewritten news headlines employ more abstract and descriptive language, extracting phrases from the browsing history and reproducing key information from the news content in the generated headline becomes less effective. This is illustrated by the second example in Table 5, where none of the generated headlines match

	Rewritten headlines			Original headlines		
	BM25	dense	random	BM25	dense	random
Original headlines	/			/		
w/o User records	74.55			79.96		
1 User records	74.67	74.35	74.53	80.15	80.04	80.04
2 User records	74.73	74.56	74.58	80.32	80.15	80.15
3 User records	74.74	74.63	74.62	80.37	80.12	80.12
4 User records	74.76	74.65	74.68	80.33	80.21	80.21
5 User records	74.76	74.69	74.64	80.42	80.21	80.21
6 User records	74.73	74.6	74.61	80.44	80.21	80.21
7 User records	74.73	74.64	74.63	80.47	80.21	80.21
8 User records	74.74	74.57	74.58	80.48	80.23	80.23
9 User records	74.66	74.60	74.60	80.45	80.22	80.22
10 User records	74.68	74.53	74.59	80.49	80.21	80.21
11 User records	74.70	74.62	74.59	80.52	80.27	80.27

Table 4: The cosine similarity results, computed by StyleDistance. Bold numbers indicate the best performance in RAG. Both the results, which are stylistically closer to the original headline and the rewritten headline, were generated using Llama3-based RAG model with BM25.

the more abstract description of "Going Dark and Fake Voyages". To further investigate how the model extracts stylistic patterns from user histories and applies them to headline generation, we analyze another 100 examples of the model that achieve best score on rewritten headlines in terms of StyleDistance (RAG with BM25, $K = 5$) and the model that shows more stable performance on ROUGE scores (RAG with BM25, $K = 8$). From these results, we discuss two representative examples. In Table 6, we can see that as the number of user records increases, the model tends to favor a concise, entity-centered style inherited from the user's profile, which may lead it to overlook certain style-level repetitions (e.g. the use of quotation marks in "different"). To further validate this observation, we conduct a named entity recognition analysis on all generated headlines from different RAG models using DistilBERT (Sanh et al., 2019). As shown in Figure 3, the average number of entities contained in each generated headline first slightly decreases and then increases as more user records are incorporated. Moreover, the generated headlines consistently contain a higher number of entities than both the original and the rewritten headlines, and the headline produced by RAG with BM25 have the highest average number of entities. The remaining examples can be found in Table 10 and Table 9 in Appendix A.

Input news	Here's what you need to know about what's happening in Phoenix. APS worker dead after fire in underground electrical vault in downtown Phoenix According to Phoenix FD...	
Original news headline	Top Phoenix news: APS worker dead after fire in electrical vault; buildings still without power.	
Rewritten news headline	APS worker dead after fire in underground electrical vault in downtown Phoenix; high-rise power cut down possibly lasts for a few days	
Model	Generation	User profile
RAG(BM25,K=8)	Top Phoenix news: APS worker dead after fire, Phoenix buildings without power, and more.	1: Top Berkeley news... ...
RAG(Dense,K=8)	APS worker dead in downtown Phoenix fire, more	1: 1 dead,several injured... ... 7: Top Berkeley news... ...
RAG(Random,K=8)	Phoenix News Briefs: Worker Dies in Underground Fire, Harassment Allegations, and More	1: Trump raises 2020 stakes... ...
w/o RAG	Phoenix News Briefs: Fatal Fire, Harassment Allegations, and NBA Updates	/
Input news	As the U.S. looks to enforce sanctions aimed at halting trade with states such as Iran and Venezuela, there's increased scrutiny on the tricks...	
Original headline	Going Dark and Fake Voyages: The Tricks Used to Dodge Trade Sanctions	
Rewritten headline	Trade sanctions: how sanctioned countries still trade	
Model	Generation	User profile
RAG(BM25,K=8)	Sanctions Busters: How Ships Are Evading Trade Restrictions	1: America...trade deficit... ...
RAG(Dense,K=8)	Sanctions Busters: How Iran and North Korea Evade Trade Restrictions	1: America...trade deficit... 2: ...Iran strikes... ...
RAG(Random,K=8)	Sanctions Busters: How Iran and North Korea Evade Trade Restrictions	1: Trump walks back statements... ...
w/o RAG	Sanctions-Busting Tactics Used to Hide Trade with Blacklisted Countries	/

Table 5: Examples of personalized news headline generation based on different RAG frameworks. Bold text indicates the model with the best performance in each example. Blue text refers to phrase-level repetitions.

Titles in browsing history from user profile		<ul style="list-style-type: none"> - Trump Slams California On Homelessness, Threatens To 'Intercede' - Bombing Range or Nature Preserve? A Battle for Control of the Nevada Desert - Chris Watts Says He Found God After Choking the Life Out of His Wife and Daughters - Trump vows to deport 'millions' of migrants, but it's unclear if there is a plan for mass arrests and removals - Iowa weather: New record for wettest 12 months in state history - Louisville considering handing over youth detention center to state - UFC 239: Ben Askren believes he's one of the best in the world and plans on beating up Jorge Masvidal to prove it - Throwback: the secret Jaguar concept car they never made
Generated news headline	RAG(BM25,K=8)	North Korean Leader Kim Jong-un Believes US is Seeking Regime Change, But Thinks Trump is Different
	RAG(BM25,K=5)	US Intelligence Says North Korea's Kim Jong-un Believes Trump is "Different"
Original news headline		Kim Jong-un believes Trump is "different," State Dept's intel arm assesses
Rewritten news headline		North Korean Leader Kim Believes Trump to be "Different" According to State Department Sources
Titles in browsing history from user profile		<ul style="list-style-type: none"> - Mauldin PD sex scandal: SLED declines investigation request - Trump Approves Strikes on Iran, but Then Abruptly Pulls Back - Underprivileged black youth at NASA became pioneer for racial equality - At G-20, Donald Trump to talk to Saudis about Iran - not Khashoggi - Heavy Rains, Growing Sinkhole Causes Problems For Rostraver Towing Company - Cloud of Cancer-Causing Chemical Hangs Over the Houston Channel - Iowa State student charged with sexual abuse pleads to lesser charge - Automakers Send Letter to Trump about Plan to Lower Fuel-Economy Rules
Generated news headline	RAG(BM25,K=8)	North Carolina Animal Sanctuary Cited for Safety Violations After Lion Mauled Intern
	RAG(BM25,K=5)	Animal Sanctuary Cited for Safety Violations After Lion Kills Intern
Original news headline		Animal sanctuary cited for safety violations months lion mauls intern
Rewritten news headline		Safety Violation Cited When Lion Fatally Mauled Intern

Table 6: Examples of personalized news headline generation based on different numbers of records. Bold text indicates similar phrases. Blue text indicates similar style-level repetitions. In these examples, the model using 8 records achieves higher ROUGE scores in its generated outputs, while the model with 5 records produces results with better stylistic similarity compared to rewritten headlines.

5 Discussion

Our results demonstrate that LLMs achieve strong performance compared with previous models. The evaluation results for both ROUGE and style distance show that news headlines generated by the

RAG models tend to be closer to the rewritten headlines compared to non-personalized generation. However, the RAG models taking into account the user history seems to maintain the similarity to the original headline more effectively. Notably, our BM25-based RAG consistently delivers

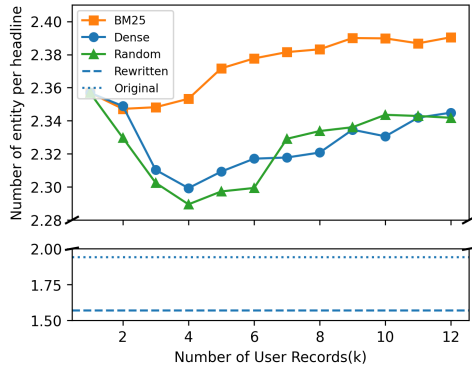


Figure 3: The average number of entities per headline across user records k .

stronger performance than dense retrievers, despite its lower retrieval accuracy. Based on our qualitative analysis, we observe that BM25’s sensitivity to features in lexical overlap beyond content relevance, can sometimes help generation by retrieving stylistically similar records. However, we still lack more precise evidence to clarify which specific properties of BM25, and under what conditions, contribute to this behavior. During our manual inspection and qualitative analysis, we found that most generated headlines follow a mainstream journalistic style: concise, and event-centered. The RAG framework tends to confuse the mainstream journalistic style present in users’ browsing histories with the users’ personal reading preferences, which may limit the adaptability of the generation model to users favoring alternative styles.

A similar discrepancy occurs in evaluation, where recurring headline styles from user history are confused with user preferences. Although the results from both the LLM-as-a-judge and `StyleDistance` evaluations indicate that RAG can improve stylistic similarity, this improvement remains limited. Across different evaluations, the headlines generated by RAG are often closer to the original news headlines. In the LLM-as-a-judge evaluation, we observe that when the clicked headlines are used as references, the model generally prefers either the original headlines, which share the general media styles with the user’s history clicks, or the generated headlines that also follow this style. This creates a disconnect from using rewritten headlines as the gold reference for personalization, raising an important question: when defining a personalization reference, should we rely on the user’s observed behavior or on their preferred linguistic expression?

Based on the current results, when user history is used as retrieval context, RAG-based models tend to reproduce stylistic patterns that are common in news media, making it difficult to disentangle general journalistic style from user-specific preferences. Existing work on personalized generation has not explicitly addressed this distinction. Moreover, this ambiguity is not only a modeling issue but also reflected in the data itself. There is no strict boundaries in the users’ personalized style and the standard news style they are frequently exposed to, which is supported by our observations that rewritten headlines remain consistently closer to original headlines in terms of both n-gram overlap and stylistic similarity than any generated outputs. Therefore, we contend that both types of headlines should jointly serve as gold references for personalization: one as a reference for the general news style alignment, and the other as a reference for the user’s style expression alignment. This dual alignment allows the model to preserve general journalistic conventions while also capturing subtle variations that reflect user preferences.

6 Conclusion and future work

The RAG-based approaches for personalized news headline generation demonstrate that using the users’ browsing history can effectively improve personalized news headline generation. We further find that in RAG systems, where retrieved records serve as user representation, quantity, diversity, and relevance of user records together play important roles for the generation. Results from our multiple experiments further reveal that, both in generation and evaluation, LLMs exhibit a strong reliance on general journalistic writing paradigms. In this context, RAG’s performance stems primarily from the model’s improved ability to learn and reproduce the styles of the news media that are embedded in users’ records, rather than from effectively capturing users’ personal reading preferences. This finding highlights a fundamental challenge in personalized text generation: distinguishing between stylistic conformity and true personalization.

Moving forward, while preserving the model’s capacity to learn news writing styles, we plan to explore how records capturing users’ reading styles can be identified and more effectively integrated into headline generation, while disentangling it from global media styles and designing evaluation protocols that better reflect personalization.

Limitations

Due to computational limitations, we were restricted during our experiments to only evaluate three open-source LLMs with comparable parameter scales. Using other, bigger or even more advanced, models could potentially further improve the baseline performance of headline generation.

Additionally, again due to computational restrictions, we did not use all 50 user records available for each news item in the dataset. It is therefore possible that the performance trends observed across different RAG experiments may change when $k > 11$. Also, we have not fully exploited the retrieved results. During our qualitative analysis, both the dense and BM25 retrievers occasionally include the same user record in the prompt. However, since all records are ranked in descending order by retrieval score, the BM25-based model produced better generation results. This suggests that the ranking order of user records may play an important role in shaping the generation outcome.

We also lack methods for detecting the factuality and consistency of the generated headlines. None of the existing evaluation methods in this research, whether n-gram-based metrics, style similarity measures, or BERTScore, are capable of addressing this issue. Based on the StyleDistance score and our qualitative analysis, we found that when using the RAG model to extract style from user records, it is difficult to separate the user's personal reading style from the general media stylistic choice. Moreover, incorporating more user records will further overshadow the user's preferred reading style.

References

- Mistral AI. 2025. Mistral-7b-instruct-v0.3. <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>. Hugging Face model card.
- Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. 2021. PENS: A dataset and generic framework for personalized news headline generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 82–92, Online. Association for Computational Linguistics.
- BAAI. 2023. Baai/bge-base-en-v1.5. <https://huggingface.co/BAAI/bge-base-en-v1.5>. Hugging Face model card.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- IBM. 2024. Granite 3.3-8b-instruct. <https://huggingface.co/ibm-granite/granite-3.3-8b-instruct>.
- Anubhav Jangra, Bahareh Sarrafzadeh, Silviu Cucerzan, Adrian de Wynter, and Sujay Kumar Jauhar. 2025. Evaluating style-personalized text generation: Challenges and directions. *arXiv preprint arXiv:2508.06374*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Pranjal Kumar. 2024. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260.
- Xinyu Li, Ruiyang Zhou, Zachary C Lipton, and Liu Leqi. 2024. Personalized language modeling from personalized human feedback. *arXiv preprint arXiv:2402.05133*.
- Junhong Lian, Xiang Ao, Xinyu Liu, Yang Liu, and Qing He. 2025. Panoramic interests: Stylistic-content aware personalized headline generation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1109–1112.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- OpenAI. 2024. Gpt-4o-mini model documentation. <https://platform.openai.com/docs/models#gpt-4o-mini>. Accessed: 2025-02-13.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2025. Styledistance: Stronger content-independent style embeddings with synthetic parallel examples. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8662–8685.

- Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. 2025. Measuring what makes you unique: Difference-aware user modeling for enhancing llm personalization. *arXiv preprint arXiv:2503.02450*.
- Han Ren, Xiaona Chang, and Xia Li. 2025. Neural headline generation: A comprehensive survey. *Neurocomputing*, page 129633.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. *Okapi at TREC-3*. British Library Research and Development Department.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. **LaMP: When large language models meet personalization**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1294–1304.
- Morten Skovsgaard and Kim Andersen. 2020. Conceptualizing news avoidance: Towards a shared understanding of different causes and potential solutions. *Journalism studies*, 21(4):459–476.
- Yun-Zhu Song, Yi-Syuan Chen, Lu Wang, and Hong-Han Shuai. 2023. General then personal: Decoupling and pre-training for personalized headline generation. *Transactions of the Association for Computational Linguistics*, 11:1588–1607.
- Xiaoyu Tan, Leijun Cheng, Xihe Qiu, Shaojie Shi, Yuan Cheng, Wei Chu, Yinghui Xu, and Yuan Qi. 2024. Enhancing personalized headline generation via offline goal-conditioned reinforcement learning with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5762–5772.
- Qwen Team et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2:3.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. **C-pack: Packaged resources to advance general chinese embedding**.
- Peng Xu, Chien-Sheng Wu, Andrea Madotto, and Pascale Fung. 2019. Clickbait? sensational headline generation with auto-tuned reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3065–3075.
- Zhao Yang, Junhong Lian, and Xiang Ao. 2023. Fact-preserved personalized news headline generation. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1493–1498. IEEE.
- Kui Zhang, Guangquan Lu, Guixian Zhang, Zhi Lei, and Lijuan Wu. 2022. Personalized headline generation with enhanced user interest perception. In *International Conference on Artificial Neural Networks*, pages 797–809. Springer.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix

A.1 Infrastructure details

All the experiments were executed on an NVIDIA A100-SXM4 GPU with 40GB memory, using the vLLM framework, with GPU memory utilization capped at 70%, enabling efficient batched inference of up to 32 sequences in parallel.

A.2 BERTScore of different RAG frameworks

In Table 7, although the difference between models is not large, we can still see that as the number of user records increases, results of all RAG models improve. Already when $k = 2$, RAG with BM25 outperforms the generation without any user records, reaching the best performance in $F1$ compared to original news headlines.

A.3 The examples of Personalized generation with RAG

From Table 9 and Table 10, we observe that headlines generated by RAG-based models are more closely aligned with the original news headlines, and the model using fewer user records tend to be relatively less similar. In Table 10, RAG with BM25 generally exhibits a stronger tendency to generate headlines with more detailed information.

	BERTScore								
	F1			PRECISION			RECALL		
	BM25	dense	random	BM25	dense	random	BM25	dense	random
w/o User records	89.21			89.46			89.00		
1 User records	89.02	89.16	89.07	89.16	89.30	89.26	88.92	89.05	88.91
2 User records	89.24	89.27	89.15	89.37	89.44	89.36	89.15	89.14	88.99
3 User records	89.30	89.28	89.17	89.43	89.47	89.40	89.21	89.13	88.97
4 User records	89.31	89.31	89.19	89.43	89.49	89.42	89.29	89.16	89.00
5 User records	89.35	89.32	89.21	89.46	89.49	89.43	89.29	89.18	89.02
6 User records	89.37	89.33	89.22	89.47	89.50	89.42	89.30	89.21	89.04
7 User records	89.37	89.33	89.21	89.47	89.49	89.41	89.32	89.22	89.04
8 User records	89.38	89.35	89.26	89.46	89.50	89.44	89.33	89.25	89.10
9 User records	89.38	89.35	89.24	89.45	89.49	89.43	89.35	89.25	89.10
10 User records	89.37	89.36	89.25	89.44	89.50	89.42	89.34	89.26	89.11
11 User records	89.38	89.36	89.26	89.45	89.50	89.44	89.35	89.26	89.12

Table 7: BERTScore of different numbers of RAG frameworks compared to original news headlines. Bold numbers indicate the best performance. The best results in F1 and Recall are generated by Llama 3 combined with user records retrieved by the BM25 retriever. The best results in Precision are generated combined with user records retrieved by the dense retriever.

A.4 The construction of LLM-as-a-Judge

The prompt used in both Mistral-7B-Instruct-v0.3 (AI, 2025) and GPT-4o-mini (OpenAI, 2024) to construct the LLM-as-a-Judge is shown in the Table 11. During the evaluation, we use headlines from the past 50 records of the current user as reference texts. For each evaluation instance, the rewritten headline, original headline, and generated headline are randomly shuffled and assigned to labels A, B, and C before being presented to the judge model, preventing the LLM from exhibiting order dependencies during evaluation.

	BERTscore								
	F1			RECISSION			RECALL		
	bm25	dense	random	bm25	dense	random	bm25	dense	random
w/o User records	87.08			86.99			87.20		
1 User records	86.90	86.97	87.01	86.75	86.79	86.89	87.09	87.20	87.16
2 User records	87.08	87.11	87.08	86.92	86.97	86.99	87.28	87.28	87.22
3 User records	87.12	87.14	87.11	86.96	87.01	87.03	87.32	87.30	87.23
4 User records	87.12	87.15	87.11	86.94	87.02	87.02	87.33	87.32	87.24
5 User records	87.13	87.16	87.12	86.94	87.02	87.03	87.36	87.33	87.25
6 User records	87.14	87.15	87.12	86.95	87.01	87.02	87.38	87.34	87.27
7 User records	87.14	87.16	87.11	86.95	87.01	87.00	87.38	87.35	87.26
8 User records	87.15	87.16	87.13	86.95	87.01	87.00	87.39	87.36	87.29
9 User records	87.15	87.16	87.12	86.94	87.00	86.99	87.39	87.36	89.30
10 User records	87.14	87.17	87.13	86.93	87.01	86.99	87.39	87.37	89.30
11 User records	87.14	87.16	87.14	86.94	87.00	87.01	87.39	87.37	87.32

Table 8: BERTScore of different numbers of RAG frameworks compared to rewritten news headlines. Bold numbers indicate the best performance in each standard. The best result in F1 is generated by Llama 3 combined with user records retrieved by the dense retriever, the best result in Precision is generated combined with random retrieved user records, and the best result in Recall are generated combined with user records retrieved by BM25.

News index: 416	Generated news headline	RAG(BM25,K=8)	More Than 50 Tips to Help You Lose Weight and Keep It Off
		RAG(BM25,K=5)	56 Weight Loss Tips to Help You Lose Weight and Keep It Off
	Original news headline		More Than 50 Tips to Help You Lose Weight and Keep It Off
	Rewritten news headline		50 Detailed Tips To Help You Melt Those Stubborn Pounds
News index: 444	Generated news headline	RAG(BM25,K=8)	Foods to Cut Out on the Keto Diet
		RAG(BM25,K=5)	Foods to Avoid on the Keto Diet
	Original news headline		These Are the Foods You Absolutely Must Cut Out While on the Keto Diet
	Rewritten news headline		The Foods The Keto Diet Absolutely Does Not Allow
News index: 658	Generated news headline	RAG(BM25,K=8)	Toronto Maple Leafs Sign Alexander Kerfoot and Cody Ceci to New Contracts
		RAG(BM25,K=5)	Maple Leafs Sign Alexander Kerfoot and Cody Ceci to New Contracts
	Original news headline		Toronto Maple Leafs to sign Alexander Kerfoot, Cody Ceci
	Rewritten news headline		Maple Leaf’s Announces New Deals

Table 9: Additional examples of personalized news headline generation based on different numbers of records. Bold text indicates similar phrases between the generated headlines, original and rewritten headlines.

News index: 79	Original news headline	Kevin Durant, Kyrie Irving to Nets 'pretty much done'?"
	Rewritten news headline	Kevin Durant and Kyrie Irving are to the Brooklyn Nets?
	RAG(BM25,K=8)	NBA Rumors: Kevin Durant and Kyrie Irving to Brooklyn Nets "Pretty Much Done"
	RAG(Dense,K=8)	NBA Rumors: Kevin Durant and Kyrie Irving to Join Brooklyn Nets
	RAG(Random,K=8)	NBA Rumors: Kevin Durant and Kyrie Irving to Join Brooklyn Nets
	w/o RAG	NBA Rumors: Durant and Irving Linked to Brooklyn Nets
News index: 213	Original headline	Kawhi Leonard had Lakers change meeting place so he could meet Paul George after?
	Rewritten headline	NBA Deals: Kawhi Leonard
	RAG(BM25,K=8)	Kawhi Leonard's Westlake Village Meeting with Lakers May Have Been a Ploy to Recruit Paul George
	RAG(Dense,K=8)	Kawhi Leonard May Have Outmaneuvered Lakers in Meeting Location Request
	RAG(Random,K=8)	Kawhi Leonard's Westlake Village Meeting with Lakers May Have Been a Power Play
	w/o RAG	Kawhi Leonard's Mysterious Meeting with Lakers Raises Questions
News index: 777	Original headline	The Best New Makeup Products to Try in 2019
	Rewritten headline	Makeup Products That You Should Definitely Add To Your Collection
	RAG(BM25,K=8)	The Best New Makeup Launches of April You Need to Know About
	RAG(Dense,K=8)	The Best New Makeup Launches This Month
	RAG(Random,K=8)	Beauty Team's Top Picks for April's Best New Makeup Launches
	w/o RAG	The Best New Makeup Launches to Add to Your Collection This Month

Table 10: Additional examples of personalized news headline generation based on different RAG frameworks. Bold text indicates similar phrases between the generated headlines, original, and rewritten headlines. In the last case, the model without RAG demonstrates better performance.

System	You are a helpful style judge that provides concise answers to style related questions based on the provided text. You output exactly one of A, B or C.
User	Choose exactly ONE among A, B, or C that BEST MATCHES the users' past preference. Focus on tone, register, vocabulary, punctuation, and phrasing. Ignore content/topic. Output EXACTLY one of: A, B, or C. No extra text. [USER STYLE EXAMPLES]: {reference_text} [Candidates]: A: {A} B: {B} C: {C} Your answer(A/B/C):

Table 11: The prompt used in LLM-as-a-Judge.

Building Multi-turn Intent Classification with LLM-based Labeling

Biancen Xie*, Kaiqi Bian*, Jai Ranjan Singh Gusain
Manikandarajan Ramanathan, Raj Maragoud

Amazon

{biancen, kbbian, jgusain, mramnat, maragoud}@amazon.com

Abstract

Intent classification is essential for customer service routing, connecting customers to the appropriate agents and reducing handling time and operational cost. Developing a real-world multi-turn intent classification system is challenging due to complex intent taxonomies, dynamic intent switching within conversations, and limited labeled training data. To address these challenges, we propose a scalable multi-turn intent classification framework for e-commerce customer service that models intent along multiple dimensions. We introduce LLM-based labeling strategies to annotate real customer transcripts at scale and augment training with LLM-simulated multi-turn dialogues that expand coverage of topic and intent switches, which are rare in existing transcripts. Through extensive experiments, we find that explanation-guided labeling with a self-critique step produces the most accurate training labels. Fine-tuned models built on a RoBERTa backbone outperform zero-shot LLM prompting while achieving substantially lower inference latency. Finally, we show that a hybrid approach that combines the fine-tuned classifier with LLM prompting further improves accuracy over either component alone. Overall, our results provide practical guidance for building and deploying high-accuracy, low-latency, large-scale multi-turn intent classification systems.

1 Introduction

Agentic customer service has become increasingly important for e-commerce platforms (Cui et al., 2017; Zhou et al., 2023). Different agents, either LLMs (Large Language Models) or workflows, are designed or trained to handle specific customer issues. Intent detection is therefore crucial for efficient routing. Intent classification failures may lead to irrelevant responses or unnecessarily escalate to human agents, increasing operational cost and degrading customer experience (Qi et al., 2021).

Both authors contributed equally.

In recent years, LLMs have demonstrated strong potential for improving intent detection due to their few-shot generalization and broad world knowledge (Zhao et al., 2023; Arora et al., 2024). However, deploying intent models that achieve both high accuracy and low latency at scale remains challenging in customer service.

First, **scalability** becomes a bottleneck as business domains or product lines expand. In real-world e-commerce systems, a high-level intent such as “Cancel” or “Refund” may be associated with multiple products. Modeling intents at a product-specific level leads to label explosion and increasing model complexity. Moreover, maintaining a consistent intent taxonomy and obtaining sufficient labeled training data becomes difficult at scale (Qi et al., 2021; Liu et al., 2024a). Additional **scalability** challenges arise from compound intents. User utterances may express multiple, non-mutually exclusive intents, such as requesting an order cancellation while reporting an unrecognized charge. Traditional flat intent classifiers are ill-suited to this.

Second, **context carryover and intent switching** increase the difficulty of intent detection in multi-turn conversations. The domain of user intent might be inferred from prior context, while user goals can shift in mid-dialogue. For example, a customer may initially seek help troubleshooting a service issue and later shift to requesting service cancellation. Without modeling conversational history, a system may misattribute the current intent or fail to link it to the relevant context. Prior work has incorporated contextual signals for intent prediction (Wu et al., 2021; Nandi et al., 2024), but typically assumes access to comprehensive labeled multi-turn data or context features. Addressing topic shifts and intent switches in real customer-service applications remains underexplored.

Finally, **low-latency requirements** constrain practical deployment in production. Although LLMs

have strong potential for intent classification, high latency makes them unsuitable for real-time inference. Production deployments must therefore balance classification accuracy with computational efficiency (Liu et al., 2024a).

To address these challenges, we (i) decompose intent understanding into three dimensions—*Domain*, *Intent*, and *Issue*—so that adding new domains or intents expands only the relevant dimension. This reduces the effective label space and supports compound intent modeling by separating actionable intent types (e.g., “Cancel”, “Informational Inquiry”) from issue attributes (e.g., “Unrecognized Charge”, “Sync/Download”). We then (ii) fine-tune lightweight models leveraging various LLMs-based methods to generate labels from customer agent transcripts at scale. The model processes the concatenation of the dialogue history and the current turn, enabling context-aware, turn-level intent detection. Finally, (iii) we augment training data with LLM-simulated multi-turn dialogues that inject topic and intent switches—patterns that are rare in transcripts but critical for robustness.

2 Related work

2.1 Multi-turn intent classification

Multi-turn intent classification incorporates dialogue context to improve intent prediction. Prior work uses contextual encoders for customer service intent detection and models cross-turn dependencies via hierarchical or graph-based structures (Wang et al., 2021; Senese et al., 2020; Liu and Chen, 2019; Qin et al., 2021). We follow this line but focus on data-driven robustness to context carryover and intent switching via controllable simulation.

LLMs have also been explored for intent detection through prompting and hybrid routing (Arora et al., 2024), as well as retrieval-augmented or demonstration-based pipelines for few-shot intent prediction (Yu et al., 2021; Zhang et al., 2025; Liu et al., 2024b). In contrast, we primarily use LLMs for automatic labeling and data generation to train lightweight models, with optional LLM fallback at inference time.

2.2 User simulation

User simulation is widely used for synthetic dialogue generation and system evaluation, including agenda-based and neural approaches (Schatzmann et al., 2007; Lin et al., 2021; Sun et al., 2022). Re-

cent work shows LLMs can act as user simulators (Balog and Zhai, 2025; Balog et al., 2025). Unlike simulators aimed at general task completion, our simulator targets real-world multi-turn behaviors—especially intent switching—using control to maintain coherence and constrain generation.

2.3 Chain-of-Thought and self-critique reasoning

Chain-of-Thought and self-refinement improve LLM outputs via intermediate reasoning and iterative revision (Wei et al., 2022; Kojima et al., 2022; Madaan et al., 2023; Shinn et al., 2023). We apply these ideas to transcript labeling (rather than inference-time in-context learning) to generate higher-quality training data for low-latency intent models. Compact transformer students further support efficient deployment (Sanh et al., 2019; Jiao et al., 2020; Wang et al., 2020).

3 Preliminary

We formulate multi-turn intent classification as a multi-class text classification problem. Given a conversation history \mathcal{C} and the customer’s current utterance q , the objective is to predict an intent label t from a predefined set $T = \{t_1, \dots, t_k\}$ using a model \mathcal{M} . The predicted intent \hat{t} is obtained by maximizing the posterior probability:

$$\hat{t} = \arg \max_{t \in T} P(t | q, \mathcal{C}; \theta), \quad (1)$$

where $P(t | q, \mathcal{C}; \theta)$ denotes the probability of intent t conditioned on (q, \mathcal{C}) , parameterized by θ of \mathcal{M} . As mentioned in Section 1, incorporating both q and \mathcal{C} in our model is essential because of context carryover and intent switching in multi-turn settings.

The number of intents k grow rapidly as product lines expand. To address the scalability issue, we propose an ontology that captures complementary aspects of intent understanding (Figure 1): **Domain**, representing product categories; **Intent**, distinguishing conversational intents from actionable intents; and **Issue**, representing slot-level attributes associated with each intent. This decomposition reduces classification complexity as well as supports **compound intent modeling** naturally by decoupling intent identification from issue detection, allowing the system to jointly predict an intent and its associated issue.

We train separate models to classify *Domain*, *Intent*, and *Issue*. Their outputs are then com-

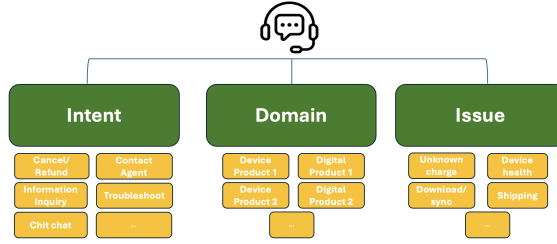


Figure 1: Intent classification ontology.

posed and mapped to downstream actions, including workflows, LLM-based agents, or human-agent hand-offs (Figure 2). Considering the latency requirement, we leverage a foundation model, Claude 3.7 Sonnet¹, to generate high-quality labeled training data. In the following sections, we introduce and evaluate multiple LLM-based labeling and data augmentation strategies that leverage existing transcript data to finetune our models under this ontology.

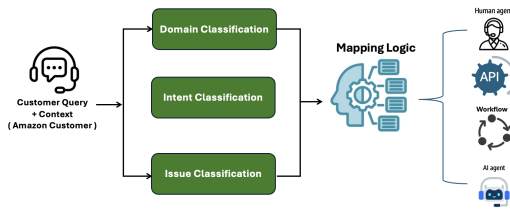


Figure 2: Intent detection system architecture

4 Method

4.1 Labeling strategy for intent classification

In real conversations, either the agent or the customer may consecutively input several utterances. However, the dialogue between a customer and a chatbot is typically conducted in an alternating manner. To construct data that aligns with the chatbot format, we merge consecutive utterances from the same role into a single unit, resulting in a dialogue $d = [q_1, a_1, \dots, q_n, a_n]$ where q_i represents the user’s current query and a_i represents the agent’s response. For a query q_i , we define its context or conversation history as $C_i = [q_1, a_1, \dots, q_{i-1}, a_{i-1}]$. In the following, we propose different label generation strategies for building our intent classification models.

Single-stage reasoning-guided labeling

¹<https://www.anthropic.com/news/claude-3-7-sonnet>

We prompt the LLM to infer the customer’s intent given the current user query q_i and its associated conversation history C_i . The prompt provides the taxonomy along with descriptions of each label, and instructs the LLM to output a single label accompanied by a free-form explanation. This explanation cites the key evidence in both the current query q_i and the context C_i that supports the model’s decision. We hypothesize that asking the LLM to articulate its reasoning and supporting evidence leads to more accurate intent classification by itself.

Two-stage labeling with self-critique

We adopt a two-stage labeling strategy that decomposes intent annotation into an *initial prediction* followed by an explicit *self-critique and revision*. **Stage 1** applies the single-stage reasoning-guided labeling procedure described in the previous section, producing an intent label with a brief explanation. In **Stage 2**, the LLM is given the original input together with the Stage-1 prediction and rationale, and is prompted to act as a critic. It assesses consistency with the dialogue evidence and intent taxonomy in Stage-1, flags failures (e.g., reliance on spurious keywords, missed contextual signals in C_i , confusion between closely related intents, or hallucinated assumptions), and then either confirms the Stage-1 label or revises it with a short justification. Stage 2 must explicitly indicate whether the label is *kept* or *revised*; when revised, it must cite minimal supporting evidence (e.g., a specific utterance in C_i or phrase in q_i) motivating the correction. We hypothesize that this two-stage self-critique improves accuracy by correcting errors introduced during initial reasoning.

To evaluate our hypothesis, we manually reviewed more than 5,500 randomly sampled test instances and report the results in Table 1. The single-stage and two-stage strategies produced identical annotations for approximately 87% of samples, and these were largely accurate. In the remaining 13% of

cases where the strategies disagreed, the two-stage approach was correct in the majority of instances, suggesting that the second-stage reassessment reduces hallucinations and overall annotation errors.

4.2 Data augmentation strategy for intent classification

Although our e-commerce platform provides access to millions of real customer service transcripts between agent and customers that can be leveraged to train models, these transcripts are typically single-topic and follow a largely linear progression. In contrast, customer–bot interactions are potentially dynamic: users may switch intents or change goals in a single session. Consequently, models trained solely on real transcripts often fail to generalize to these complex patterns.

Data generation using dialogue simulator: To bridge this gap, we develop a multi-turn dialogue generation framework that simulates realistic customer–bot interactions. Our framework uses an LLM-based user query simulator that interacts with an LLM-based response simulator, while a *Simulator Controller* orchestrates the conversation by initializing dialogues, managing turn-by-turn flow, and introducing topic/intent shifts when appropriate. The controller combines intent-level planning with controlled randomness to elicit underrepresented behaviors in real transcripts (e.g., chit-chat, intent changes, follow-up and clarification questions). At each turn, it selects the next user intent and prompts the user simulator to produce the corresponding utterance, after which the response simulator generates the bot reply. We also inject alternative seed queries sampled from a multi-domain top-query database to facilitate topic and intent transitions. The overall interaction flow is shown in Figure 3 and simulation prompt template can be found in Appendix C.3. Finally, we use the two stage labeling strategy in Section 4.1 to automatically label simulated dialogues.

4.3 Hybrid approach of LLM and fine-tuned model

To demonstrate how a fine-tuned lightweight model can be combined with an LLM to balance accuracy and latency, we introduce a *hybrid intent detection strategy* that couples fast local classification with selective LLM escalation. As shown in Appendix C.5 and Figure 4, the lightweight model first outputs an intent probability distribution along with a confidence score. If the confidence score exceeds

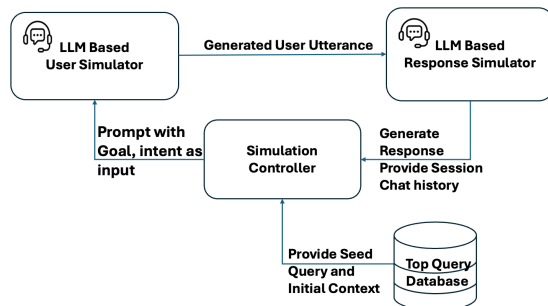


Figure 3: Multi-turn conversation simulator

a threshold τ , we accept the lightweight model’s prediction. Otherwise, we extract the model’s top K intent candidates and invoke an LLM to perform *constrained* disambiguation over this candidate set. This hybrid design routes most intent detection requests through the low-latency lightweight model, while reserving LLM calls for a small subset of ambiguous cases. The choice of τ and K is motivated by the accuracy vs coverage analyses in Appendix C.6 (Figures 5 and 6).

5 Experimental setup

5.1 Datasets

We apply the anonymization procedure described in Appendix A. Each sample consists of conversation history, the current customer utterance, and its corresponding annotations for intent, issue, and product category. To build a balanced dataset, we use stratified sampling across months and domains. Since some categories are significantly overrepresented, we further downsample high-frequency categories when forming the final training and evaluation splits. We provide additional dataset construction details are provided in Appendix B.

5.2 Evaluation benchmarks

We consider the following baselines to evaluate and benchmark our LLM-based labeling and data augmentation methods:

- **RoBERTa:** We fine-tune three separate RoBERTa-base (Liu et al., 2019) classification models (*Intent*, *Domain*, and *Issue*), enabling simultaneous prediction across multiple label spaces. To compare prompting and data-generation choices for fine-tuning (Section 4), we train variants using: (i) single-stage reasoning-guided labeling, (ii) two-stage labeling with self-critique, and (iii) two-stage labeling with

Outcome	Count
Stage 2 is correct	566
Stage 1 is correct	84
Both Stage 1 & Stage 2 are incorrect	70
Both Stage 1 & Stage 2 are correct	4802
Total	5522

Outcome	Accuracy
Stage 2 is correct	97%
Stage 1 is correct	88%

Table 1: Stage 1 and Stage 2 performance analysis

self-critique plus dialogue-simulator augmentation. We provide implementation details in appendix C.4.

- **Claude Sonnet 3.7 zero-shot:** We evaluate Claude Sonnet 3.7 in a zero-shot setting using a prompt that enumerates the intent taxonomy and provides brief definitions for each label.
- **Nova Pro zero-shot:** We mirror the **Claude Sonnet 3.7 Zero-shot** setup but replace Claude with Nova Pro, a smaller LLM that offers lower inference latency.
- **Hybrid approach (fine-tuned RoBERTa + Claude Sonnet 3.7):** Following Section 4.3, we first use the best-performing fine-tuned RoBERTa model to retrieve the top-3 candidate labels for each classifier (*Intent, Domain, Issue*). We then prompt Claude Sonnet 3.7 to select the final label conditioned on these candidates. This hybrid baseline tests whether zero-shot prompting can further improve over RoBERTa fine-tuning.

6 Results

6.1 Automatic evaluation on fine-tuned RoBERTa models

As described in Sections 4.1 and 4.2, we employ multiple data labeling and augmentation strategies to generate training data for fine-tuning RoBERTa models. Specifically, we consider three approaches for comparison: (1) a single-stage prompting strategy, (2) a two-stage prompting strategy, and (3) a two-stage prompting strategy + simulated dialogue data augmentation. We generate ground-truth labels using an LLM according to each strategy. We report automatic evaluation results in Table 2. Models trained under all three strategies achieve comparable performance across classification tasks, indicating that the fine-tuned models are able to effectively learn from their corresponding LLM teacher models. Notably, models fine-tuned using the two-stage prompting strategy exhibit substan-

Model Type	Single-Stage Labeling	Two-Stage Labeling	Two-Stage + Data Augm.
Intent Model	80.70%	80.30%	79.50%
Product Model	80.90%	80.80%	81.70%
Issue Model	76.70%	79.10%	78.20%

Table 2: Automated evaluation accuracy results

tial performance gains on the issue classification task. We hypothesize that issue classification involves greater ambiguity and is therefore more challenging. The two-stage prompting strategy, which encourages additional self-verification and refinement, produces higher-quality labels and improves fine-tuning performance.

6.2 Human evaluation

To comprehensively evaluate the effectiveness of our labeling and data augmentation strategies, as well as the benefits of the hybrid approach, we compare across several baseline approaches (see Section 5.2) using a 1,206 human-annotated dataset. The dataset is constructed by random sampling, with 50% from the evaluation dataset described in Section 5.1 and the remaining 50% from production traffic. The results, presented in Table 3, illustrate performance across all 3 types of classification models and an overall aggregate assessment based on Precision, Recall, and F1-score.

Overall, our results show that models fine-tuned with the two-stage labeling strategy and augmented with simulated dialogue data outperform alternative approaches. While training with data generated via single-stage prompting already yields strong performance, adding self-critique stage further improves labeling accuracy and model performance. Moreover, incorporating simulated multi-turn dialogues helps the model better handle intent switches, contributing to additional gains (F1 score increases by 2%-5% compared to single-stage prompting). Finally, we observe that intent classification perfor-

	Intent Model			Product Model			Issue Model		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
RoBERTa(Single-stage labeling)	73%	71%	72%	73%	65%	69%	67%	61%	64%
RoBERTa(Two-stage labeling)	74%	73%	73%	72%	69%	70%	72%	64%	68%
RoBERTa(Two-stage labeling+data augmentation)	75%	74%	74%	75%	72%	73%	73%	66%	69%
Hybrid approach	77%	75%	76%	78%	74%	76%	74%	68%	71%
Nova Pro zero-Shot	48%	42%	45%	61%	52%	56%	42%	35%	38%
Claude Sonnet 3.7 zero-shot	75%	67%	71%	61%	69%	65%	54%	51%	52%

Table 3: Performance comparison across different baselines for intent, product, and issue classification

mance is further improved by a hybrid approach that combines a fine-tuned RoBERTa model with LLM-based prompting.

Purely prompting-based methods (e.g., Claude zero-shot and Nova zero-shot) exhibit inferior performance. This degradation likely stems from the absence of domain-specific customer service training data, which increases the likelihood of hallucinations and intent misclassification. Although the hybrid approach also employs prompting when model confidence is low, it substantially outperforms standalone prompting. This gain stems from using the fine-tuned RoBERTa model to first retrieve a limited set of relevant intent candidates, thereby constraining the LLM’s decision space and enabling more accurate final predictions.

We also evaluate P50 and P90 latency across different methods, with results summarized in Table 4. The findings indicate that RoBERTa achieves the

Model	P50 Latency	P90 Latency
RoBERTa	0.08s	0.1s
Nova Pro Zero-shot	1.98s	3.97s
Claude Sonnet 3.7 Zero-Shot	5.93s	7.52s
Hybrid (RoBERTa + Zero-shot)	1.99s	3.1s

Table 4: Model latency performance comparison

lowest latency, approximately 20–50× faster than LLM zero-shot. While the hybrid approach delivers improved accuracy, it incurs higher latency than RoBERTa, suggesting a trade-off between performance gains and inference efficiency.

6.3 Online deployment performance

We deployed the intent classification model (fine-tuned RoBERTa-base) in an e-commerce customer service production systems. Compared to the previously deployed single-turn intent detection system, which could not support topic shifts or context carryover in an ongoing sessions, our model enables seamless and dynamic intent routing in multi-turn interactions. In a one-month online A/B test, our

model increased the bot automation rate by 4.91% and improved the positive customer response rate by 7.89%, demonstrating benefits for both customer experience and operational efficiency while achieving low end-to-end production latency (P50: 0.12 s; P90: 0.16 s; P99: 0.20 s).

7 Conclusion

In this paper, we address multi-turn intent detection for customer service applications. To handle the scalability and heterogeneity of intent taxonomies, we propose an ontology that captures complementary facets of user intent. To mitigate the scarcity of annotated data, we introduce LLM-based labeling methods that generate high-quality supervision from existing customer transcripts, and augment training with LLM-simulated multi-turn dialogues that explicitly model topic shifts and intent switches.

Our experiments show that two-stage labeling with self-critique, combined with simulated dialogue augmentation, consistently outperforms alternative labeling strategies. The resulting fine-tuned RoBERTa models outperform pre-trained LLMs in zero-shot settings while achieving substantially lower latency. A hybrid routing strategy that combines fine-tuned RoBERTa with an LLM further improves performance on ambiguous cases.

Our findings provide actionable guidance for practitioners building production multi-turn intent detection systems by effectively combining real transcript data with LLM-generated dialogues.

8 Limitations

Our evaluation is based on customer service conversations from a specific set of products, locales, and workflow designs, so results may not fully generalize to other domains with different intent taxonomies or dialogue patterns. In addition, model quality depends on the consistency of upstream labels (human or LLM-assisted); any ambiguity in intent definitions or noise in automatic labeling can

propagate to training and inflate offline estimates. Finally, offline metrics may not perfectly translate to end-to-end customer impact because real deployments involve additional constraints (policy, UI, latency, and fallback behavior) and are subject to distribution shift over time; broader cross-domain/locale testing and controlled online studies are needed to validate robustness and user outcomes.

References

- Gaurav Arora, Shreya Jain, and Srujana Merugu. 2024. Intent detection in the age of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*.
- Krisztian Balog, Nolwenn Bernard, Saber Zerhoubi, and ChengXiang Zhai. 2025. [Theory and toolkits for user simulation in the era of generative AI: User modeling, synthetic data generation, and system evaluation](#). In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, pages 4138–4141, Padua, Italy. Association for Computing Machinery.
- Krisztian Balog and ChengXiang Zhai. 2025. User simulation in the era of generative AI: User modeling, synthetic data generation, and system evaluation. *arXiv preprint arXiv:2501.04410*.
- Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoyun Duan, and Ming Zhou. 2017. Superagent: A customer service chatbot for e-commerce websites. In *Proceedings of ACL 2017, system demonstrations*, pages 97–102.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. [TinyBERT: Distilling BERT for natural language understanding](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.
- Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geisler, Michael Heck, Shutong Feng, and Milica Gasic. 2021. [Domain-independent user simulation with transformers for task-oriented dialogue systems](#). In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 445–456, Singapore and Online. Association for Computational Linguistics.
- Junhua Liu, Yong Keat Tan, Bin Fu, and Kwan Hui Lim. 2024a. [Balancing accuracy and efficiency in multi-turn intent classification for LLM-powered dialog systems in production](#).
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Zhengyuan Liu and Nancy Chen. 2019. Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.
- Zihan Liu, Yiming Chen, Hao Zhang, et al. 2024b. Lara: Linguistic-adaptive retrieval-augmentation for multi-turn intent classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Nanyun Raja, Shivang Gulati, Shubham Tan, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- Subhadip Nandi, Neeraj Agrawal, Anshika Singh, and Priyanka Bhatt. 2024. [Enhancing customer service chatbots with context-aware nlu through selective attention and multi-task learning](#). In *Proceedings of the 8th International Conference on Data Science and Management of Data (CODS-COMAD 2024)*, pages 220–228. Association for Computing Machinery.
- Haode Qi, Lin Pan, Atin Sood, Abhishek Shah, Ladislav Kunc, Mo Yu, and Saloni Potdar. 2021. [Benchmarking commercial intent detection services with practice-driven evaluations](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 304–310, Online. Association for Computational Linguistics.
- Libo Qin, Zhou Chen, Wanxiang Che, Hang Li, and Ting Liu. 2021. Knowing where to leverage: Context-aware graph convolutional network with an adaptive fusion layer for contextual spoken language understanding.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. [Agenda-based user simulation for bootstrapping a POMDP dialogue system](#). In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.
- Matteo Antonio Senese, Giuseppe Rizzo, Mauro Dragoni, and Maurizio Morisio. 2020. [MTSI-BERT: A session-aware knowledge-based conversational agent](#). In *Proceedings of the Twelfth Language Resources*

and Evaluation Conference, pages 717–725, Marseille, France. European Language Resources Association.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Weiwei Sun, Shuyu Guo, Shuo Zhang, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2022. Metaphorical user simulators for evaluating task-oriented dialogue systems. *arXiv preprint arXiv:2204.00763*.

Peiyao Wang, Joyce Fang, and Julia Reinspach. 2021. [CS-BERT: a pretrained model for customer service dialogues](#). In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 130–142, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Ting-Wei Wu, Ruolin Su, and Bing-Hwang Juang. 2021. [A context-aware hierarchical BERT fusion network for multi-turn dialog act detection](#). In *Proceedings of Interspeech 2021*, pages 1239–1243.

Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. [Few-shot intent classification and slot filling with retrieved examples](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 734–749, Online. Association for Computational Linguistics.

Ziji Zhang, Michael Yang, Zhiyu Chen, Yingying Zhuang, Shu-Ting Pi, Qun Liu, Rajashekar Maragoud, Vy Nguyen, and Anurag Beniwal. 2025. [REIC: RAG-enhanced intent classification at scale](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1072–1080, Suzhou (China). Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. [A survey of large language models](#).

Yunhua Zhou, Jiawei Hong, and Xipeng Qiu. 2023. [Towards open environment intent prediction](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2226–2240, Toronto, Canada. Association for Computational Linguistics.

Appendix

A Data anonymization

Due to business considerations, We manually anonymized both the labels and transcripts to ensure no personal information is included. Additionally, specific product and service names were denonymized to prevent the identification of the company from the transcript or label descriptions. Despite these modifications, the conclusions drawn from our experiments remain valid.

B Data construction

The dataset comprises both real customer service conversations and synthetic queries collected between 2024 and 2025. We apply a stratified sampling strategy across months and domains as follows: (i) to mitigate seasonality effects and prevent oversampling during peak periods, we sample an equal number of data points from each month. (ii) to reduce bias toward particular product lines or use cases, we further stratify the sampling process by selecting an equal number of examples per month across up to 30 existing skills(domains), where each skill corresponds to a specific task that an Amazon Customer Service representative can perform for a given product.

Because certain categories contain substantially more examples than others, we apply a downsampling procedure when constructing the final training and evaluation datasets. For each category, we cap the number of samples at 20,000 for training and 10,000 for testing. This downsampling strategy is applied consistently across all model types, including intent, domain, and issue classifiers. The test set was held out strictly for final evaluation.

C Implementation details

C.1 Classification prompt template

We use the following prompt template for conversation classification. It instructs the LLM to label each customer-service dialogue along three dimensions: conversational intent, issue type, and product category. For each example, the current utterance and its dialogue context are dynamically inserted into the template. The prompt provides detailed annotation guidelines, including (i) leveraging the full dialogue history for context-aware decisions and (ii) using structured reasoning to resolve ambiguous cases. We define three separate prompt templates corresponding to Stage 1, Stage 2 annotation and LLM zero-shot. Note LLM zero-shot does not ask LLM to provide justification explanation.

Classification prompt template for stage 1:

```
## Dialogue Context
{dialogue_context}

## Turn T - Current Customer Utterance
{turn_T_utterance}

## Conversational/Actional Intent Options (Choose Exactly One)

1. CANCEL - Request to cancel a subscription or service or product
2. REFUND - Request for a refund related to a service or subscription
3. CANCEL&REFUND - Explicitly state desire to cancel and request a refund
4. RETURN - Request for a return related to a service or product
5. REPLACEMENT - Request for a replacement related to a service or product
6. RETAIN/SUBSCRIBE - Request to retain/subscribe subscription and NOT to cancel
7. TRADE-IN - Customer is requesting to trade-in devices
8. ADS-FREE/ADS-REMOVAL - Customer is requesting to remove ads or subscribe ads free
9. TROUBLESHOOT - Customer describes an issue requiring troubleshooting
10. INFORMATIONAL_INQUIRY - General "how-to" or "what-is" questions
11. ISSUE_DEPENDENT_INQUIRY - Inquiries tied to a specific issue or service/product
12. Yes - The customer responded affirmatively
13. No - Customer generally respond with "No" or "Nope"
14. TRANSACTIONAL_OTHER - Other action-oriented or transactional inquiries
15. CHIT_CHAT/CONVERSATION_FILLER - Polite fillers, navigation phrases
16. FRUSTRATION/COMPLAINT - General dissatisfaction expressed
17. REQUEST_HUMAN_AGENT - Customer explicitly asks for a human agent
18. END_CONVERSATION - Customer clearly states issue is resolved
```

19. NON_AMAZON_TOPIC - Completely unrelated to Amazon products or services

Issue Options (Choose Exactly One)

1. SETUP/REGISTRATION/ACTIVATION/INSTALLATION
2. CANCEL/REFUND/RETURN/REPLACEMENT/RETAIN/SUBSCRIBE/TRADE-IN/TRANSACTIONAL
3. CONNECT/PAIR
4. WIFI/NETWORK
5. SYNC/DOWNLOAD
6. PLAY/STREAMING/DISPLAY
7. AUDIO/SOUND
8. DAMAGE/REPAIR
9. WARRANTY
10. DEVICE_HEALTH
11. DEFECT
12. BATTERY
13. SOFTWARE/OTA_UPDATE
14. LOST/STOLEN
15. PROMOTIONS/CREDITS/LOOT
16. RESTRICTIONS/PARENTAL_CONTROL
17. HOUSEHOLD
18. PAYMENT_ISSUE/PURCHASE/COINS
19. PRICING
20. UNRECOGNIZED_CHARGES/UNKNOWN_CHARGES/FRAUD_CHARGES
21. CONTENT_AVAILABILITY
22. SHIPPING/DELIVERY
23. OTHER_ISSUE
24. NONE

Product Options (Choose Exactly One)

1. Music
2. Video
3. SmartTV

[... additional product categories ...]

Output Format (JSON)

```
{
  "predicted_conversational_intent": "One of the 19 intent labels",
  "predicted_issue": "One of the 24 issue labels",
  "predicted_product": "One of the 31 product labels",
  "reason": "Brief explanation justifying the choice"
}
```

Classification prompt for stage 2:

Taxonomy Intents

1. CANCEL - Request to cancel subscription/service/product or turn off auto renewal. Excludes refunds.
2. REFUND - Request for refund related to service/subscription.
3. CANCEL&REFUND - Explicitly cancel and request refund.
4. RETURN - Request for return related to service/product.
5. REPLACEMENT - Request for replacement related to service/product.
6. RETAIN/SUBSCRIBE - Request to retain/subscribe, NOT cancel.
7. TRADE-IN - Customer requesting to trade-in devices.
8. ADS-FREE/ADS-REMOVAL - Request to remove ads or subscribe ads free.
9. TROUBLESHOOT - Customer describes issue requiring troubleshooting.
10. INFORMATIONAL_INQUIRY - General "how-to" or "what-is" questions answerable from help pages. Excludes vague utterances like "I need help".
11. ISSUE_DEPENDENT_INQUIRY - Inquiries tied to specific issue/service/product requiring customer context. Exclude transactional intents.
12. Yes - Customer responded affirmatively. Don't confuse with END_CONVERSATION.
13. No - Customer responds "No" or "Nope". Don't confuse with END_CONVERSATION.
14. CHIT_CHAT/CONVERSATION_FILLER - Polite fillers (hi, thanks),

- navigation phrases. Exclude frustration/complaint and Yes/No.
15. FRUSTRATION/COMPLAINT - General dissatisfaction without clear request.
 16. REQUEST_HUMAN_AGENT - Customer explicitly asks for human agent.
 17. END_CONVERSATION - Customer states issue resolved. Don't confuse with polite chit-chat.
 18. NON_RELATED_TOPIC - Completely unrelated to the e-commerce. Must be completely off-topic.

Taxonomy Issues

1. SETUP/REGISTRATION/ACTIVATION/INSTALLATION - Issues registering, setting up, activating, or installing device/subscription/service/app.
2. CANCEL/REFUND/RETURN/REPLACEMENT/RETAIN/SUBSCRIBE/TRADE-IN/TRANSACTIONAL - Issue with transaction already occurred or attempted.
3. CONNECT/PAIR - Issues connecting or bluetooth pairing.
4. WIFI/NETWORK - Issues with networking or Wifi.
5. SYNC/DOWNLOAD - Trouble syncing/downloading (often eBook/App related).
6. PLAY/STREAMING/DISPLAY - Trouble playing content or visualizing.
7. AUDIO/SOUND - Audio issues (no audio, low audio, etc.).
8. DAMAGE/REPAIR - Device damage, inquiring about repair.
9. WARRANTY - Inquiring about or checking warranty.
10. DEVICE_HEALTH - Device performance/stability issues (crashes, reboots, bricked, frozen, responsiveness).
11. DEFECT - General mention of device defect.
12. BATTERY - Battery issues.
13. SOFTWARE/OTA_UPDATE - Software update issues (stuck on update screen, no update available).
14. LOST/STOLEN - Reporting lost/stolen device.
15. PROMOTIONS/CREDITS/LOOT - Issues with promotions, bundles, credits, loot.
16. RESTRICTIONS/PARENTAL_CONTROL - Trouble with Parental Controls, Pins, child purchases.
17. HOUSEHOLD - Issues with household (sharing content, adding members). If PINS/Child Controls, select Parental Controls instead.
18. PAYMENT_ISSUE/PURCHASE/COINS - Issue/inquiry related to payment, gift cards, or coins.
19. PRICING - Price related issues (price match, price adjustment).
20. UNRECOGNIZED_CHARGES/UNKNOWN_CHARGES/FRAUD_CHARGES - Generic unrecognized charges customer not aware of or deems fraud.
21. CONTENT_AVAILABILITY - Issues with content availability.
22. SHIPPING/DELIVERY - Shipping or delivery related issues.
23. OTHER_ISSUE - Other action-oriented/transactional inquiries not mentioned above.
24. NONE - No specific issue, purely chit-chat, complaint or frustration.

Taxonomy Products

1. Music.
 2. Video
 3. Video Channels
 4. SmartTV Cube
- [... additional product categories ...]

Classification prompt template for LLM zero-shot:

Dialogue Context
{dialogue_context}

Turn T - Current Customer Utterance
{turn_T_utterance}

Conversational/Actional Intent Options (Choose Exactly One)

1. CANCEL - Request to cancel a subscription or service or product
2. REFUND - Request for a refund related to a service or subscription
3. CANCEL&REFUND - Explicitly state desire to cancel and request a refund
4. RETURN - Request for a return related to a service or product
5. REPLACEMENT - Request for a replacement related to a service or product
6. RETAIN/SUBSCRIBE - Request to retain/subscribe subscription and NOT to cancel

7. TRADE-IN - Customer is requesting to trade-in devices
8. ADS-FREE/ADS-REMOVAL - Customer is requesting to remove ads or subscribe ads free
9. TROUBLESHOOT - Customer describes an issue requiring troubleshooting
10. INFORMATIONAL_INQUIRY - General "how-to" or "what-is" questions
11. ISSUE_DEPENDENT_INQUIRY - Inquiries tied to a specific issue or service/product
12. Yes - The customer responded affirmatively
13. No - Customer generally respond with "No" or "Nope"
14. TRANSACTIONAL_OTHER - Other action-oriented or transactional inquiries
15. CHIT_CHAT/CONVERSATION_FILLER - Polite fillers, navigation phrases
16. FRUSTRATION/COMPLAINT - General dissatisfaction expressed
17. REQUEST_HUMAN_AGENT - Customer explicitly asks for a human agent
18. END_CONVERSATION - Customer clearly states issue is resolved
19. NON_AMAZON_TOPIC - Completely unrelated to Amazon products or services

Issue Options (Choose Exactly One)

1. SETUP/REGISTRATION/ACTIVATION/INSTALLATION
2. CANCEL/REFUND/RETURN/REPLACEMENT/RETAIN/SUBSCRIBE/TRADE-IN/TRANSACTIONAL
3. CONNECT/PAIR
4. WIFI/NETWORK
5. SYNC/DOWNLOAD
6. PLAY/STREAMING/DISPLAY
7. AUDIO/SOUND
8. DAMAGE/REPAIR
9. WARRANTY
10. DEVICE_HEALTH
11. DEFECT
12. BATTERY
13. SOFTWARE/OTA_UPDATE
14. LOST/STOLEN
15. PROMOTIONS/CREDITS/LOOT
16. RESTRICTIONS/PARENTAL_CONTROL
17. HOUSEHOLD
18. PAYMENT_ISSUE/PURCHASE/COINS
19. PRICING
20. UNRECOGNIZED_CHARGES/UNKNOWN_CHARGES/FRAUD_CHARGES
21. CONTENT_AVAILABILITY
22. SHIPPING/DELIVERY
23. OTHER_ISSUE
24. NONE

Product Options (Choose Exactly One)

1. Music
2. Video
3. SmartTV

[... additional product categories ...]

Output Format (JSON)

```
{
  "predicted_conversational_intent": "One of the 19 intent labels",
  "predicted_issue": "One of the 24 issue labels",
  "predicted_product": "One of the 31 product labels",
}
```

C.2 System prompt

We define three separate prompt templates corresponding to Stage 1, Stage 2 annotation and LLM zero-shot. Note LLM zero-shot and Stage 1 annotation shares the same system prompt.

Stage 1 or LLM zero-shot system prompt:

You are an expert annotation assistant specializing in analyzing conversations between customers and bots/agents. Your task is to classify each customer message (Turn T) into its primary intent and the most relevant Amazon product or service discussed.

Use the provided dialogue history for context, and ensure that classifications adhere strictly to the predefined categories. Always select exactly one intent

and one product for each message, even if the product is inferred from the context.

If the product is ambiguous but likely Amazon-related, choose 'Other'. If the message is unrelated to Amazon, select 'NON_AMAZON_TOPIC' as the intent. Provide clear reasoning for your classifications, referencing specific dialogue cues and your decision-making process.

Stage 2 system prompt:

You are a strict annotation reviewer. Your job is to AUDIT a prior classification (Stage 1) for a customer's Turn T, using the SAME taxonomy as Stage 1.

Goals:

- Verify that the predicted intent, issue, and product each match their definitions.
 - Challenge the original reasoning (try to find contradictions or missing evidence).
 - Correct any mistakes; otherwise confirm the original labels.
 - Be conservative with ambiguous cases: only use NON_AMAZON_TOPIC when clearly unrelated to Amazon; do not confuse CHIT_CHAT with END_CONVERSATION; do not confuse YES/NO with functional intents.
 - Always choose EXACTLY ONE intent, ONE issue, and ONE product from the Stage-1 taxonomy (no new labels).
 - Keep reasoning concise and reference concrete spans from the dialogue (short quotes). IMPORTANT: Do NOT repeat the Stage-1 JSON.
- Produce the reviewer JSON ONLY using the final_* keys.

C.3 Dynamic conversation simulation prompt templates

The following prompt templates were used in this study to serve a two-stage approach for generating realistic customer service conversations. The `system_prompt_dialogue_helper` and `user_intent_helper` work together to analyze existing conversation history and identify all possible customer intents (such as ChitChat, Frustration, IntentChange, or FollowUpQuestion) that could naturally occur next in the dialogue flow. Once potential intents are identified, the system randomly selects one and employs the `system_prompt_talk_to_bot` and `user_prompt_turn_helper` templates to generate authentic customer responses that align with the chosen intent. This dual-phase prompting strategy ensures that simulated conversations maintain conversational coherence while introducing realistic variability in customer behavior, enabling comprehensive testing of chatbot performance across diverse interaction scenarios.

`system_prompt_dialogue_helper`:

ROLE:

You are the dialogue helper for a user simulator helping find the intent for the user given a conversation history.

TASK:

Select out all POSSIBLE intent from CANDIDATE LIST to carry on the conversation given the previous Conversation history.

GUIDELINES:

1. Read through the whole conversation and identify the subset of POSSIBLE INTENTS

CANDIDATE LIST:

- 1.ChitChat: Small talk loosely related to previous chat history
- 2.Frustration: Expression of frustration in the middle of conversation
- 3.intentChage: The user changes their request midway through a conversation, the request can be related to previous Conversation history
- 4.FollowUpQuestion: The user asks follow-up questions that are related to previous Conversation history
- 5.Clarification: The user asks for clarification for certain points in previous Conversation history
- 6.Rambling: Speaker(s) ramble and repeat themselves. They may paraphrase themselves
- 7.ContactRealAgent: Request to speak to a real agent
- 8.EndConversation: End Conversation naturally when the issue or problem is resolved

Here is the input format
<Conversation history>
[provide chat_history here]
</Conversation history>

Here is the output format
<Possible Intents>
[provide possible intents here]
</Possible Intents>

user_intent_helper:

Here is your input:

<Conversation history>
{chat_history}
</Conversation history>

Now respond with what the customer would say next:

system_prompt_talk_to_bot:

ROLE:

You are a user engaging in a natural conversation with a customer service bot or agent. Your goal is to generate the next user turn based on the conversation history and the intent provided below.

OBJECTIVES

Conversation-Level Goal:

Seek a resolution (e.g., HOW-TO answer) to the seed query provided below.

Current Turn Goal:

Generate a user response that aligns with the current intent described below.

INTENT DEFINITIONS

ChitChat - Casual or light-hearted comments loosely related to the conversation.

Frustration - Expressions of annoyance or dissatisfaction.

IntentChange - The user changes their goal mid-conversation, potentially related to prior turns.

FollowUpQuestion - User asks a question that builds directly on prior discussion.

Clarification - User requests clarification about something mentioned previously.

Rambling - User paraphrases, repeats, or meanders while staying within the topic.

ContactRealAgent: Request to speak to a real agent

EndConversation: End Conversation naturally when the issue or problem is resolved

GUIDELINES:

-If no chat history exists, begin with the seed query.

-Respond naturally: ask relevant questions, express preferences, or make decisions as needed.

-If the bot successfully resolves the task and provides a reference number, reply only with: "I'm all set" (no additional text).

-If the bot is repetitive or unhelpful across multiple turns, escalate by using Contact Real Agent intent and say "talk to a real agent".

-Do not impersonate a bot or break character.

-Be concise and speak like a real customer in real life. Each response should be less than 25 words.

-If the intent is either ContactRealAgent or EndConversation. Be concise and the response should be straight forward.

Here is the input format

<seed_query>
[provide seed_query here]
</seed_query>
<Provided Intent>
[provide intent here]
</Provided Intent>
<Conversation history>
[provide chat_history here]
</Conversation history>
Here is the output format
<Current Turn>

```
[provide the current turn]
</Current Turn>
"""
```

user_prompt_turn_helper:

Here is your input:

```
<Conversation history>
{chat_history}
```

```
</Conversation history>
```

```
<Provided Intent>
{provided_intent}
</Provided Intent>
```

Now respond with what the customer would say next:

C.4 RoBERTa model fine-tuning implementation details

We use RoBERTa-base for intent detection model, optimizing with cross-entropy and the Adam optimizer. Models are trained for 10 epochs on 8 NVIDIA A10 GPUs, with learning rates of $1e-5$, batch sizes of 32 and early stopping(patience = 5) to prevent overfitting.

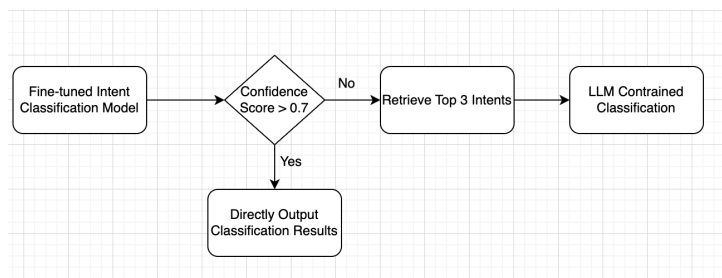


Figure 4: Hybrid deployment approach

C.5 Hybrid approach work flow implementation Details

Figure 4 illustrates the workflow of our hybrid approach. In our implementation, we set the confidence threshold to $\tau = 0.7$ and use $K = 3$ candidate intents for LLM disambiguation (see Appendix C.6). We use a fine-tuned RoBERTa model as the lightweight classifier, and invoke Claude Sonnet 3.7 for intent classification when the confidence score falls below the threshold.

C.6 Analysis of τ and K in hybrid approach

We select the confidence threshold by balancing accuracy and coverage. As shown in Figure 5, setting the threshold to 0.7 allows our fine-tuned model to cover roughly 80% of intent-detection requests while maintaining about 85% accuracy. We therefore choose 0.7 as the operating point because it offers a practical tradeoff: the low-latency model can handle the majority of traffic with sufficiently high accuracy for direct deployment. For the remaining low-confidence cases (confidence ≤ 0.7), we defer intent detection to an LLM. Importantly, for these instances, the ground-truth intent appears in our fine-tuned model’s top-3 predictions nearly 90% of the time (Figure 6). This suggests that low confidence typically reflects ambiguity among a small set of plausible intents rather than a complete failure. Accordingly, we ask the LLM to select among the top-3 candidate intents produced by the fine-tuned model.

This yields an effective hybrid intent detection strategy: high-confidence requests (confidence ≥ 0.7) are handled directly by the lightweight model, while only a small fraction of ambiguous cases trigger a secondary LLM call over a constrained top-3 label set. This design improves accuracy on difficult utterances while keeping overall latency under control.

This is a section in the appendix.

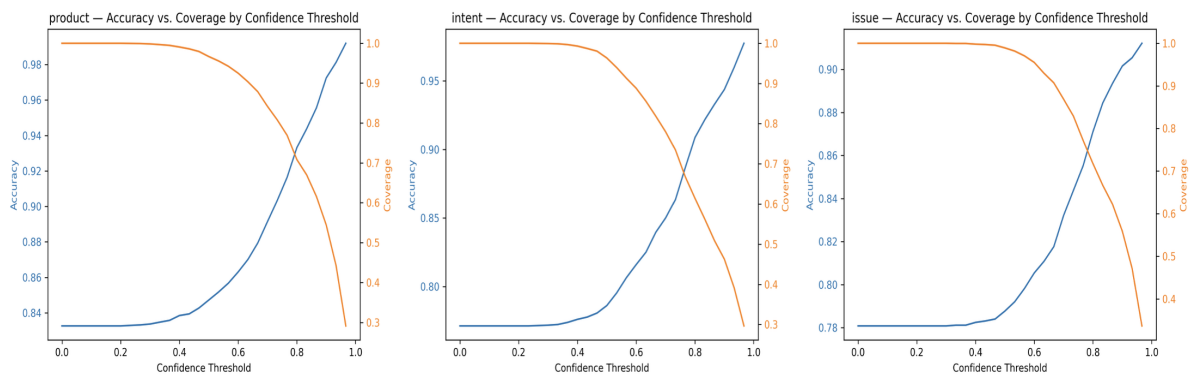


Figure 5: Accuracy–coverage tradeoff under different confidence thresholds

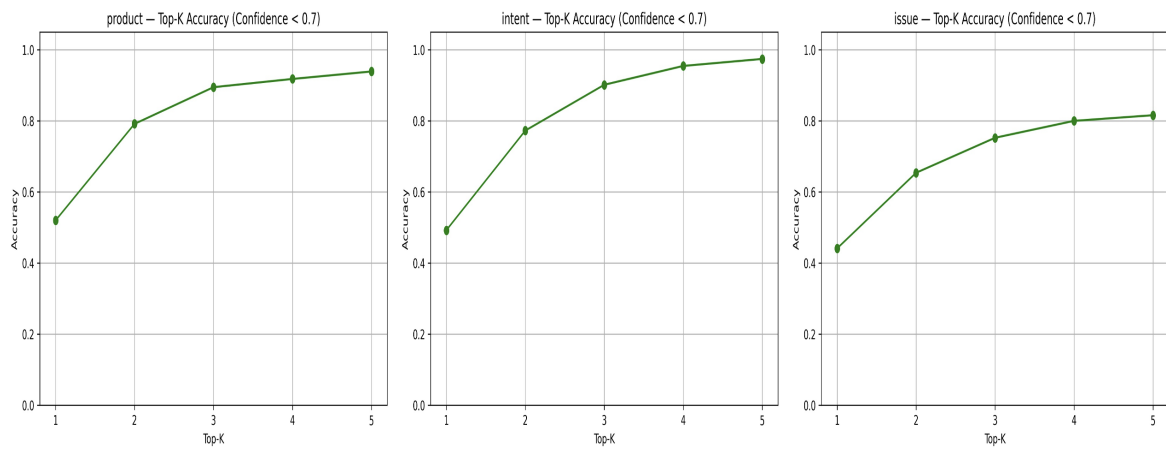


Figure 6: Top- K accuracy for low-confidence predictions

Cross-Tokenizer LLM Distillation through a Byte-Level Interface

Avyav Kumar Singh^{1*†}, Yen-Chen Wu^{1*}, Alexandru Cioba^{1‡},
Alberto Bernacchia¹, Davide Buffelli¹,

¹MediaTek Research, Cambridge (United Kingdom)

Correspondence: davide.buffelli@mtkresearch.com

Abstract

Cross-tokenizer distillation (CTD), the transfer of knowledge from a teacher to a student language model when the two use different tokenizers, remains a largely unsolved problem. Existing approaches rely on heuristic strategies to align mismatched vocabularies, introducing considerable complexity. In this paper, we propose a simple but effective baseline called **Byte-Level Distillation (BLD)** which enables CTD by operating at a common interface across tokenizers: the byte level. In more detail, we convert the teacher’s output distribution to byte-level probabilities, attach a lightweight byte-level decoder head to the student, and distill through this shared byte-level interface. Despite its simplicity, BLD performs competitively with—and on several benchmarks surpasses—significantly more sophisticated CTD methods, across a range of distillation tasks with models from 1B to 8B parameters. Our results suggest that the byte level is a natural common ground for cross-tokenizer knowledge transfer, while also highlighting that consistent improvements across all tasks and benchmarks remain elusive, underscoring that CTD is still an open problem.

1 Introduction

Large Language Models (LLMs) demonstrated unprecedented capabilities in natural language understanding, generation, and reasoning. Their applications are becoming ubiquitous, from conversational agents (e.g., (Guo et al., 2025; Yang et al., 2025; OpenAI, 2025)) and next-generation search engines (Xi et al., 2025) to tools that assist in scientific discovery (Zhang et al., 2024b) and software development (Dong et al., 2025). The re-

markable performance of these models, however, is intrinsically linked to their scale with state-of-the-art LLMs often comprising billions of parameters. This size renders their training prohibitively expensive for most research institutes, and often inference becomes prohibitively slow for real-time or on-device applications.

To bridge the gap between the capabilities of large frontier models and the practical constraints of real-world systems, knowledge distillation has emerged as a seminal technique (Hinton et al., 2015). Distillation is a process in which a compact *student* model is trained to mimic the behavior of a larger, more powerful *teacher* model. Instead of learning solely from hard labels in a dataset, the student learns from the rich, dense output distribution produced by the teacher. This allows the student to inherit the teacher’s sophisticated reasoning patterns while operating with a fraction of the computational footprint. The impact of distillation is already evident across the research environment and the industry, e.g., it enables to speedup the training of small specialized models, and to “compress” models and lower costs when serving them at scale (Xu et al., 2024).

Despite its success, the standard framework for knowledge distillation is built on a fundamental, yet restrictive, assumption: the teacher and student models must share an identical tokenizer and vocabulary. This is because the most common form of distillation operates at the *logit* level, where the student is trained to match the teacher’s probability distribution over a fixed set of vocabulary tokens. If the tokenizers differ, their corresponding vocabularies lead to distinct output spaces. A logit vector of size 50,000 from the teacher cannot be directly compared to a logit vector of size 32,000 from the student. Consequently, performing *cross-tokenizer distillation* (CTD) has been considered infeasible without resorting to approximations or heuristics. These workarounds, such as distilling

*Equal contribution.

†Work done during an internship at MediaTek Research. Avyav is now at King’s College London, London (United Kingdom).

‡Work done while at MediaTek Research. Alexandru is now at Orbital Materials, London (United Kingdom).

from generated text samples (Kim and Rush, 2016) or attempting to create ad-hoc mappings between vocabularies or hidden states (Boizard et al., 2025; Wan et al., 2024; Zhang et al., 2024a; Minixhofer et al., 2025), are either computationally inefficient, suffer from significant information loss, or lack a principled theoretical foundation.

The ability to perform principled CTD would unlock powerful new paradigms. First, it would allow us to combine the distinct strengths of diverse models. For instance, one could distill the broad world knowledge of a general-purpose model (e.g., trained with a large, multilingual tokenizer) into a specialized student model equipped with a domain-specific tokenizer optimized for medicine, law, or finance. This would create highly efficient and accurate expert models. Second, it would enable distillation from ensembles of heterogeneous models. For example, training a single student by distilling the collective intelligence of several top-tier open-source models (e.g., DeepSeek (Guo et al., 2025), Qwen (Yang et al., 2025), GPT-OSS (OpenAI, 2025), etc.), each with its own tokenizer. This would allow the student to learn a consensus of knowledge that potentially surpasses any individual teacher.

In this paper we introduce Byte-Level Distillation (BLD), which sidesteps the vocabulary mismatch in cross-tokenizer distillation by operating at the byte level—a representation shared by all tokenizers. Our method (i) converts the teacher’s token-level output distribution to byte-level probabilities using a fast approximation (Vieira et al., 2025), (ii) attaches a lightweight, learnable byte-level decoder head to the student in parallel with its original token-level head, and (iii) performs distillation through this shared byte-level interface. After distillation, the byte-level head is simply removed, leaving a standard token-level model. This approach enables direct and effective knowledge transfer between models with different tokenizers.

Despite its simplicity, BLD performs competitively with—and on several benchmarks surpasses—substantially more complex CTD methods across tokenizer transfer and cross-model distillation tasks with models ranging from 1B to 8B parameters. At the same time, no method, including ours, achieves consistent gains across all benchmarks, suggesting that CTD remains an open and challenging problem. In summary, our contributions are:

- We propose BLD, a simple and alignment-free baseline for CTD that operates through a shared byte-level interface.
- We empirically show that this simple approach performs competitively with significantly more complex state-of-the-art CTD methods across a range of tasks.
- Through our analysis of the results, we highlight that no existing method—including ours—consistently dominates across benchmarks, and argue that CTD remains a largely open problem deserving further investigation.

2 Related Work

Our work is positioned at the intersection of three active areas of research: cross-tokenizer knowledge distillation, byte-level language modeling, and methods for converting token-level probability distributions to the byte level.

Cross-Tokenizer Distillation The challenge of transferring knowledge between models with different tokenizers is a significant hurdle for standard distillation techniques. Several recent works have proposed approximate or heuristic methods to bridge this gap. For instance, some approaches focus on aligning the vocabularies of the teacher and student models through various mapping strategies. Boizard et al. (2025) introduce a Universal Logit Distillation (ULD) loss based on optimal transport theory, which allows for distillation across different architectures and tokenizers without requiring them to share the same vocabulary. Other works, like Wan et al. (2024) and Zhang et al. (2024a), explore knowledge fusion and dual-space distillation, respectively, to enable knowledge transfer between heterogeneous models. Similarly, Minixhofer et al. (2025) propose a method for universal cross-tokenizer distillation through approximate likelihood matching. These methods often introduce additional complexity and rely on approximations to align the output spaces of the models. In contrast, our proposed BLD method circumvents this issue by operating at the byte level, a universal interface shared by all tokenizers.

Byte-Level Probability Estimation A core component of our BLD method is the ability to obtain a byte-level probability distribution from a standard token-based language model. This has been the focus of a number of recent studies. Vieira et al.

(2025) present algorithms for converting token-level language models into character-level ones. Phan et al. (2025) introduce the Byte-Token Representation Lemma, a framework that provides a formal mapping between a model’s learned token distribution and its equivalent byte-level distribution. Our work leverages the insights from these works to create a shared byte-level space for distillation.

Byte-Level Language Models Our work is also related to the growing body of research on byte-level language models, which can be broadly categorized by how they process raw byte sequences. First are the pure byte-level models, which operate directly on sequences of bytes without any explicit grouping. Xue et al. (2022), with their ByT5 model, demonstrated that a standard Transformer architecture can be adapted to process byte sequences effectively, achieving competitive performance with token-level models while being more robust to noise. More recently, Wang et al. (2024) proposed MambaByte, a token-free model based on the selective state space architecture. Second are models that use fixed chunking to group bytes into patches. YU et al. (2023) introduced MEGABYTE, a multi-scale architecture that segments long byte sequences into fixed-size patches, using a local model within patches and a global model across them. Slagle (2024) proposed SpaceByte, which uses larger Transformer blocks after specific bytes (like spaces) to more efficiently model byte sequences. The autoregressive U-Net (AU-Net) of Videau et al. (2025) also falls into this category, as it pools bytes into a multi-scale representation based on fixed rules. Third are models that employ learned chunking to dynamically group bytes. Hierarchical Transformers like the Hourglass model from Nawrot et al. (2021) and the dynamic pooling mechanism from Nawrot et al. (2023) laid the groundwork for more flexible byte-level processing. More recent works have built on this, such as the Byte Latent Transformer (BLT) from Pagnoni et al. (2025), which encodes bytes into dynamically sized patches based on next-byte entropy, and MrT5 from Kallini et al. (2025), which uses dynamic token merging. The H-Net model from (Hwang et al., 2025) takes this a step further with a dynamic chunking mechanism that learns content- and context-dependent segmentation directly from the data, effectively creating an end-to-end, tokenizer-free model. While our method does

not involve using byte-level models, it can be used to distill information from token based model into byte-level ones.

3 Our Method

3.1 Preliminaries

Let Σ be the alphabet containing all bytes, i.e., $\{1, 2, \dots, 256\}$, and let Σ^* be the set of all sequences over the alphabet. Given a vocabulary $V \subseteq \Sigma^*$, which determines all the possible tokens, a tokenizer is a deterministic function that maps sequences of bytes to sequences of tokens: $\mathcal{T} : \Sigma^* \rightarrow V^*$, where V^* indicates the set of all sequences composed of tokens from the vocabulary V . We also define a decoder function $\mathcal{D} : V^* \rightarrow \Sigma^*$ as the function that “maps back” from a sequence of tokens to a sequence of bytes. We can assume that the decoder function is the inverse of the tokenizer, i.e., $\mathcal{D}(\mathcal{T}(\{b_1, b_2, \dots, b_{N_b}\})) = \{b_1, b_2, \dots, b_{N_b}\}$, with N_b indicating the length of the byte sequence, though this is not always the case in practice¹.

When performing distillation, the goal is to transfer knowledge from a teacher model to a student model. The teacher model has an associated vocabulary V_T , tokenizer \mathcal{T}_T , and decoder \mathcal{D}_T . The teacher model can be seen as a function mapping a given tokenized input sequence into a probability distribution over its vocabulary indicating the probability of the next token, $f_T : \mathcal{T}_T(\Sigma_T^*) \rightarrow \Delta(V_T)$, where $\Delta(V_T)$ is the probability simplex over the vocabulary. Similarly, the student model also has a vocabulary V_S , tokenizer \mathcal{T}_S , and decoder \mathcal{D}_S , which may differ from those of the teacher.

In standard distillation approaches, given a dataset of tokenized sequences $\mathcal{Z} = \{s_1, s_2, \dots\}$, each one composed of multiple tokens $s_i = \{t_1, t_2, \dots, t_{|s_i|}\}$, the student model parameters are updated by minimizing the following loss function

$$\mathcal{L} = \sum_{s_i \in \mathcal{Z}} \frac{1}{|s_i|} \left(\sum_{t_j \in s_i} \text{CE}(\delta(t_j), f_S(t_{<j})) + \text{KL}(f_T(t_{<j}), f_S(t_{<j})) \right) \quad (1)$$

where $\delta(t_j)$ is the delta function which is zero everywhere except at the index of token t_j for which it is equal to 1, $t_{<j}$ indicates the sequence

¹This is because in practice tokenizers involve some pre-tokenization steps which are not reversible, like for example normalizing Unicode characters.

of tokens up to the j -th token excluded, CE indicates cross-entropy, and KL indicates the Kullback–Leibler divergence. The first term in equation 1, the cross entropy, is the standard next token prediction loss, while the second term, the KL divergence, is responsible for transferring knowledge from the teacher to the student. Notice however that for the latter to be well defined, it requires teacher and student to have the same vocabulary, which in practice usually leads to sharing also the same tokenizer, although in theory it could be different between the two. Recently, several works have introduced heuristic or approximate strategies to overcome this issue (Boizard et al., 2025; Wan et al., 2024; Zhang et al., 2024a; Minixhofer et al., 2025). These approaches require identifying some form of alignment between tokenizations and introducing additional heuristic losses. Our approach instead overcomes these challenges by performing distillation at the byte level.

From BPE-level to Byte-Level Probabilities.

Given a sequence of bytes $\{b_1, b_2, \dots, b_{N_b}\}$ and a teacher model f_T with vocabulary V_T and tokenizer \mathcal{T}_T , Phan et al. (2025) and Vieira et al. (2025) show that it is possible to compute the probability of generating a sequence of bytes using the model f_T by summing the probabilities that the model assigns to all the *coverings* of the byte sequence. Let us define a *covering*, associated to the teacher model, for a byte sequence $\{b_1, b_2, \dots, b_{N_b}\}$ as the set containing all the sequences of tokens that “cover” the sequence of bytes when decoded, i.e.,

$$\begin{aligned} \text{cover}_T(b = \{b_1, b_2, \dots, b_{N_b}\}) = \\ \{ \{t_1, t_2, \dots, t_m\} \in V_T^* \mid \exists i \in \mathbb{Z}^{>0} \text{ s.t.} \\ \mathcal{D}_T(\{t_1, t_2, \dots, t_{m-1}\}) = b_{<i} \text{ and} \\ b_{\geq i} \text{ is a prefix of } \mathcal{D}(t_m) \} \end{aligned} \quad (2)$$

We can now compute the probability assigned by the teacher to a byte sequence $b = \{b_1, b_2, \dots, b_{N_b}\}$ as

$$P_T(b) = \sum_{y_i \in \text{cover}_T(b)} \prod_{t_j^{(i)} \in s_i} f_T(t_j^{(i)} | t_{<j}^{(i)})$$

From this we can straightforwardly obtain the conditional probabilities for each single byte in the sequence as

$$P_T(b_i | b_{<i}) = \frac{P_T(\{b_1, b_2, \dots, b_i\})}{P_T(\{b_1, b_2, \dots, b_{i-1}\})} \quad (3)$$

The above procedure can be quite expensive computationally, but Vieira et al. (2025) provide a fast approximation, which we use for our method. More details are provided in Appendix C.

A naive approach to byte level CTD. Given that we can extract the probabilities at the byte level from any token based model, one might think of “going back” from byte level to a different token level to perform CTD. In fact, a naive approach for byte-level CTD, once the probabilities $P_T(b_i | b_{<i})$ at the byte level are extracted from the teacher for a given sequence, could be to use them to construct the probabilities of a tokenized version of the sequence in which the student’s tokenizer is used instead. In more detail, given a sequence $b = \{b_1, b_2, \dots\}$, we can tokenize it using the student’s tokenizer into a sequence of tokens $\{y_1, y_2, \dots\} = \mathcal{T}_S(b)$, and then compute the probability of each possible token (as this is needed for the KL term in the distillation loss) in V_S as follows

$$\begin{aligned} \forall t = \{b_1^{(t)}, \dots, b_k^{(t)}\} \in V_S, \\ P(y_i = t | y_{<i}) = \prod_{b_j^{(t)} \in t} P_T(b_j^{(t)} | b_{<j}^{(t)}, y_{<i}) \end{aligned} \quad (4)$$

where, with a slight abuse of notation, we use $P_T(b_j^{(t)} | b_{<j}^{(t)}, y_{<i})$ to indicate the probability assigned by the teacher to the j -th byte of token t given all previous bytes in the whole sequence. This quantity is computed using the equations presented above. The advantage of this approach is that there is no need to add any module to the original architecture of the student (which instead is required in our method). On the other side, this approach has several issues that make it impractical. First, equation (4) requires the computation of $|V_S|$ probabilities – which in practice is between 30000 and 250000 – for each token in the sequence (where the sequence is tokenized according to the student’s tokenizer \mathcal{T}_S), which would be computationally prohibitive. Second, if the byte level probabilities are computed with an approximate method, the errors will compound when computing equation (4).

3.2 Byte-Level Interface for Distillation

Our method, called Byte Level Distillation (BLD), can be divided into two steps which we present below. A schematization of BLD can be found in Figure 1.

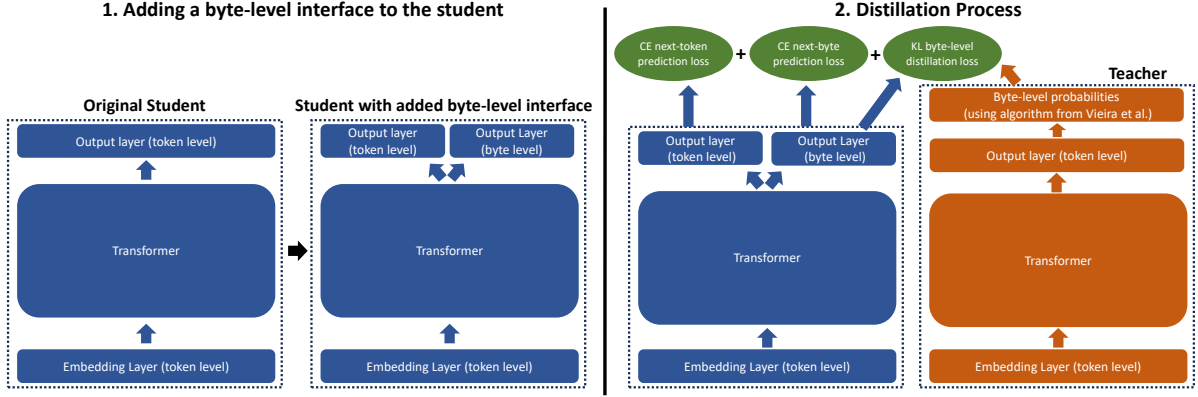


Figure 1: Representation of our Byte-Level Distillation (BLD) method composed of two steps. Step 1 adds a byte-level interface to the student model. Step 2 performs distillation by transferring knowledge from the teacher to the student using the shared byte-level interface. Additional next-token prediction and next-byte prediction losses are also used following standard distillation approaches. The byte-level interface can be removed at the end of the process.

Step 1: byte-level interface. The first step is to enable teacher and student models to share knowledge through the byte level. For the teacher we can use the approach presented in Section 3.1 to compute byte-level probabilities, but for enabling training of the student we need to introduce a new module to it. We start from a pretrained student model. The model is composed of a tokenizer $\mathcal{T}_S : \Sigma^* \rightarrow V_S^*$ with a respective decoder $\mathcal{D}_S : V_S^* \rightarrow \Sigma^*$, an encoder $E : V_S^* \rightarrow \mathbb{R}^{N \times d}$ (typically a learnable embedding matrix with one row for each element of the vocabulary V_S), a transformer $H : \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{N \times d}$, and an output layer $O : \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{N \times |V_S|}$. Here N is the input sequence length (in terms of numbers of tokens from the vocabulary V_S), and d is the dimension of token embeddings and hidden representations (we assume they are the same for simplicity of presentation but in practice hidden dimensions at every layer of the transformer can be different from the dimensions of token embeddings). We now add a new learnable module to the student model. In more detail, in parallel to the existing token-level decoder O , we add byte-level decoder: $O_b : \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{N_b \times |\Sigma|}$, where N_b is the length of the input sequence in terms of bytes. With this we have effectively added a *byte-level interface* to the output of the student model².

Step 2: distillation. Given a teacher model, we use the method of Vieira et al. (2025) to obtain

²The byte-level decoder can be pre-trained while keeping the rest of the weights fixed for additional stability, but in our experiments we found that it is not necessary.

$P_T(b_i | b_{<i})$ for each sequence $x_i = \{b_1, b_2, \dots\}$ in a given dataset \mathcal{D} . We can now perform distillation without requiring any specific alignment or heuristic as we have the probabilities at the byte-level obtained from the teacher model, and our student model has an output interface at the byte level. During distillation the loss is a combination of next-byte cross entropy loss, KL divergence at the byte level, and next-token cross entropy loss³. Formally, let $f_S : \mathcal{T}_S(V_S^*) \rightarrow \Delta(V_S)$ be the function at the token level for the student model obtained by composing $f_S(t) = O(H(E(t)))$ and let $f_S^{(b)} : \mathcal{T}_S(V_S^*) \times \mathbb{Z}^{>0} \rightarrow \Delta(\Sigma)$ be the function with the byte-level interface for the student model, i.e., $f_S^{(b)}(t, j) = O_b(H(E(t)))[j]$ (where “[j]” indicates selecting the j -th byte of the output), then the full loss for distillation is:

$$\mathcal{L} = \sum_{\substack{x_i \in \mathcal{Z}, \\ \{t_1, t_2, \dots, t_k\} = \mathcal{T}_S(x_i), \\ t_i = \{b_1^{(i)}, \dots, b_{n_i}^{(i)}\}}} \frac{1}{k} \sum_{\ell=1}^k \left[\text{CE}(\delta(t_\ell), f_S(t_{<\ell})) + \frac{1}{n_\ell} \sum_{j=1}^{n_\ell} \text{CE}(\delta(b_j^{(\ell)}), f_S^{(b)}(t_{<\ell}, j)) + \text{KL}(P_T(b_j^{(\ell)} | b_{<j}^{(\ell)}, t_{<\ell}), f_S^{(b)}(t_{<\ell}, j)) \right] \quad (5)$$

where $P_T(b_j^{(\ell)} | b_{<j}^{(\ell)}, t_{<\ell})$ indicates the probability assigned by the teacher to the j -th byte in the ℓ -th token given all bytes in the sequence (including those from tokens prior to the ℓ -th) up to the $(j-1)$ -

³The next token cross entropy loss is added to ensure the weights of the token-level decoder O get updated.

th byte of the ℓ -th token. All or a subset of the parameters of the model can be updated during distillation, except from the byte-level output layer which must be updated if not pre-trained first.

After distillation, we remove the byte-level interface O_b , and thus keeping only the token-level output layer O . It is also possible to instead keep the byte level output layer if one is interested in generating outputs in terms of bytes or combinations of tokens and bytes. In our experiments, as byte-level decoder O_b , we use a simple linear projection for O_b with N_b fixed to 10, which means that for tokens that span more than 10 bytes, supervision signal will be provided only for the first ten. We validate our choice experimentally as shown in Appendix A. A different approach could be to have a small autoregressive layer to accommodate different values of N_b . We leave these directions for future work.

4 Experiments

To evaluate our approach, we follow the experimental procedure of [Minixhofer et al. \(2025\)](#) which considers three tasks: tokenizer transfer across different BPE tokenizers, tokenizer transfer from BPE to byte, and cross-tokenizer distillation.

Training setup. We fine-tune the student backbone with LoRA ([Hu et al., 2022](#)), applying rank $r = 64$ updates to the query and value projection matrices while keeping all other backbone weights frozen. For tokenizer transfer experiments, the embedding matrix and LM head are re-initialised using Fast Vocabulary Transfer (FVT) ([Gee et al., 2022](#)): tokens present in both vocabularies are initialised by directly copying the corresponding source embedding; tokens absent from the source vocabulary are initialised as the mean of their constituent sub-token embeddings, falling back to a random Gaussian sample drawn from the source embedding distribution when no decomposition is available. The byte-level decoder head O_b is a lightweight module consisting of 10 parallel linear projections from the model’s hidden dimension to the byte vocabulary (260 tokens representing the 256 bytes and 4 special tokens for: beginning of sequence, end of sequence, padding, and out-of-vocabulary), enabling each token position to predict up to 10 bytes simultaneously (see Appendix A for a validation of this approach). We optimize with AdamW ([Loshchilov and Hutter, 2019](#)) using a cosine learning rate schedule with linear warm-

up. Full hyperparameter details are provided in Appendix B. Importantly, we use the same SFT backbone for all considered distillation methods.

Training datasets For the BPE tokenizer transfer and byte tokenizer transfer experiments, we train on the Tulu-3 SFT mixture ([Lambert et al., 2024](#)). Byte-level teacher probabilities are pre-computed offline for this dataset using the fast approximation of [Vieira et al. \(2025\)](#), as described in Appendix C. For the cross-tokenizer distillation experiment (OpenMath2-Llama3.1-8B \rightarrow Gemma2 2B), we train on the OpenMathInstruct-2 dataset ([Toshniwal et al., 2024](#)).

Validation datasets For the tokenizer transfer experiments, we use the no-robots split of Tulu-3 ([Rajani et al., 2023](#)) as a held-out validation set; this subset spans a diverse range of tasks—including coding, mathematics, and general reasoning—making it a representative signal for general-purpose capability. For the cross-tokenizer distillation experiment, we randomly sample approximately 1,000 examples from OpenMathInstruct-2 as a held-out validation set.

4.1 BPE Tokenizer Transfer

We first evaluate our method on the task of *tokenizer transfer* between two different BPE tokenizers. This involves selecting a pre-trained model, in our case LLama 3.2 3B ([Meta, 2024](#)) and replacing its tokenizer with the BPE tokenizer from Qwen 2 ([Yang et al., 2024](#)). The procedure involves replacing embedding and output projection layers with uninitialized layers in accordance (in terms of dimensionalities) with the new tokenizer, and distilling from the original model to the modified one. We present results in Table 1.

Table 1 shows that BLD performs competitively but does not uniformly dominate. It achieves the highest scores on PiQA (75.68) and AGI-ZH (35.97), and recovers performance close to the original model on PiQA, MMLU, and BoolQ, demonstrating that distillation through the byte-level interface successfully transfers general knowledge after tokenizer replacement. ALM + SFT is the strongest overall competitor, leading on four of seven benchmarks (ARC-C, BoolQ, MMLU, AGI-EN). The most notable weakness of BLD is instruction following: its IFEval score (30.58) lags far behind MinED (62.83) and ALM + SFT (58.51), both of which retain near-original IFEval performance.

Model	Method	Benchmark						
		PiQA	ARC-C	BoolQ	MMLU	AGI-EN	AGI-ZH	IFEval
<i>original (Llama3.2 3B IT)</i>		75.46	45.73	78.41	60.50	35.27	42.93	66.31
→ Qwen2	SFT	74.54	41.89	76.48	57.11	30.47	34.30	26.74
	DSKD	62.95	28.84	71.80	50.48	26.12	34.18	28.13
	MinED	75.35	42.58	78.65	58.20	34.68	34.76	62.83
	ALM + SFT	75.46	45.82	79.36	58.86	36.64	35.27	58.51
	BLCTD (Ours)	75.68	43.26	77.34	58.29	31.98	35.97	30.58

Table 1: Results of transferring Llama3.2 3B (Meta, 2024) to the Qwen2 tokenizer (Yang et al., 2024). *original* denotes the original model without transfer. *ARC-C* refers to Arc-Challenge. *AGI-EN* and *AGI-ZH* refer to the English and Chinese splits of AGIEval.

Model	Method	Benchmark						
		PiQA	ARC-C	BoolQ	MMLU	AGI-EN	AGI-ZH	IFEval
<i>original (Llama3.2 3B IT)</i>		75.46	45.73	78.41	60.50	35.27	42.93	66.31
→ Byte	SFT	67.30	31.57	73.00	38.95	26.05	35.18	24.70
	DSKD	64.47	31.31	60.34	37.62	23.74	33.36	23.98
	MinED	67.41	32.94	65.32	39.84	27.52	33.90	31.89
	ALM + SFT	66.32	31.57	71.41	39.15	27.66	35.39	29.74
	BLCTD (Ours)	67.52	30.89	69.85	39.06	26.44	34.57	25.43

Table 2: Results of transferring Llama3.2 3B (Meta, 2024) to byte-level tokenization. *original* denotes the original model without transfer. *ARC-C* refers to Arc-Challenge. *AGI-EN* and *AGI-ZH* refer to the English and Chinese splits of AGIEval.

This suggests that the byte-level distillation objective does not sufficiently preserve the structured output behaviour required for instruction following. DSKD performs worst across all benchmarks, confirming that direct distribution alignment without vocabulary alignment is ineffective in this setting.

4.2 BPE-to-byte Tokenizer Transfer

We now repeat the same *tokenizer transfer* task as the previous section, but this time we transfer from a BPE tokenizer to byte-level. This can be seen as adapting Llama 3.2 3B (Meta, 2024) to be a byte-level model. The procedure involves replacing embedding and output projection layers with uninitialized layers compatible with a byte-level tokenizer, and distilling from the original model to the modified one. We present results in Table 2.

Results in Table 2 show that transferring to byte-level tokenization is substantially harder than BPE-to-BPE transfer: all methods suffer large degradations across every benchmark (e.g., MMLU drops approximately 21 points and ARC-C approximately 13 points relative to the original model), reflecting the challenge of adapting a model trained on subword tokens to a much finer-grained representation. In this setting, BLD ranks first on PiQA (67.52), though the margin over MinED (67.41) is negligible. Performance leadership is fragmented across methods: SFT leads on BoolQ (73.00), MinED on ARC-C (32.94) and MMLU

(39.84), and ALM + SFT on AGI-EN (27.66) and AGI-ZH (35.39). The spread between methods is noticeably narrower than in Table 1, suggesting that in this harder regime all approaches converge to a similar performance ceiling. DSKD again performs worst across most benchmarks. Overall, no method establishes a clear advantage, and the collective degradation relative to the original underscores that byte-level tokenizer transfer remains an unsolved challenge.

4.3 Cross-Tokenizer Distillation

Finally, we perform CTD across different models with different tokenizers. In more detail, we distill the maths-specialised OpenMath2-Llama3.1-8B (Toshniwal et al., 2024) into Gemma2 2B (Deepmind, 2024). Results are shown in Table 3.

Table 3 shows that BLD achieves the highest GSM8K score (62.55), modestly outperforming ALM + SFT (61.56) and SFT (59.29), and represents a meaningful gain over the uninitialised Gemma2 2B IT baseline (51.48). However, SFT leads on MATH (22.40 vs. 20.08 for BLD), suggesting that BLD’s advantage over SFT is task-dependent and does not generalise uniformly across mathematical reasoning benchmarks. Despite BLD’s result, the gap to the teacher (87.26 GSM8K, 37.60 MATH) remains very large, highlighting that effective cross-tokenizer knowledge transfer across heterogeneous models is still an open problem.

Model	Method	GSM8K	MATH
OpenMath2-Llama3.1-8B		87.26 ± 0.92	37.60 ± 2.16
Gemma2 2B IT		51.48 ± 1.38	10.60 ± 1.38
Gemma2 2B	SFT	59.29 ± 1.35	22.40 ± 1.87
	ALM + SFT	61.56 ± 1.34	19.00 ± 1.76
	Ours	62.55 ± 1.33	20.08 ± 1.82

Table 3: Results of cross-tokenizer distilling the large math-specialized OpenMath2-Llama3.1-8B (Toshniwal et al., 2024) into the small Gemma2 2B (Deepmind, 2024) language model. All results are zero-shot CoT.

5 Limitations

Due to computational constraints, our work explores the task of tokenizer transfer with 3 billion parameter models, and the task of CTD between an 8 billion parameter teacher and a 2 billion parameter student. While these are practical sizes for models that are destined to run on-device, the behavior of CTD methods at larger scales remains underexplored.

Similarly, our distillation makes use of LORA to reduce the computational requirements, and performing full-parameter optimization may lead to higher performance.

6 Conclusions

In this paper we introduced BLD, a simple baseline for cross-tokenizer knowledge distillation that operates through a shared byte-level interface. By converting the teacher’s output distribution to byte-level probabilities and attaching a lightweight byte-level decoder head to the student, our method avoids the complex vocabulary alignment procedures required by existing approaches. Despite this simplicity, BLD performs competitively with—and on several benchmarks outperforms—substantially more sophisticated methods across both tokenizer transfer and cross-model distillation settings. The effectiveness of this approach can be enhanced much further, for example, one can use a byte-level transformer architecture as opposed to MLP byte-level heads to capture sequential dependencies at the byte level.

Nevertheless, our experiments reveal a sobering finding: no method, including ours, achieves consistent improvements across all benchmarks and tasks. Performance leadership shifts depending on the benchmark, the transfer target, and the specific model pair. This inconsistency suggests that cross-tokenizer distillation remains a fundamentally open problem. We thus encourage the community to continue pursuing this line of research which has strong practical implications.

References

- Nicolas Boizard, Kevin El Haddad, CELINE HUDELOT, and Pierre Colombo. 2025. [Towards cross-tokenizer distillation: the universal logit distillation loss for LLMs](#). *Transactions on Machine Learning Research*.
- Google Deepmind. 2024. [Gemma 2: Improving open language models at a practical size](#). *Preprint*, arXiv:2408.00118.
- Yihong Dong, Xue Jiang, Jiaru Qian, Tian Wang, Kechi Zhang, Zhi Jin, and Ge Li. 2025. [A survey on code generation with llm-based agents](#). *Preprint*, arXiv:2508.00083.
- Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torrioni. 2022. [Fast vocabulary transfer for language model compression](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 409–416, Abu Dhabi, UAE. Association for Computational Linguistics.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *Preprint*, arXiv:1503.02531.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*.
- Sukjun Hwang, Brandon Wang, and Albert Gu. 2025. Dynamic chunking for end-to-end hierarchical sequence modeling. *arXiv preprint arXiv:2507.07955*.
- Julie Kallini, Shikhar Murty, Christopher D Manning, Christopher Potts, and Róbert Csordás. 2025. [Mrt5: Dynamic token merging for efficient byte-level language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327.

- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengding Hu, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. [Tülu 3: Pushing frontiers in open language model post-training](#). *Preprint*, arXiv:2411.15124.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Meta. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. 2025. Universal cross-tokenizer distillation via approximate likelihood matching. In *The Thirty-Ninth Conference on Neural Information Processing Systems*.
- Piotr Nawrot, Jan Chorowski, Adrian Lancucki, and Edoardo Maria Ponti. 2023. [Efficient transformers with dynamic token pooling](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6403–6417, Toronto, Canada. Association for Computational Linguistics.
- Piotr Nawrot, Szymon Tworowski, Michal Tyrolski, Lukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. 2021. [Hierarchical transformers are more efficient language models](#). *CoRR*, abs/2110.13711.
- OpenAI. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. [Byte latent transformer: Patches scale better than tokens](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.
- Buu Phan, Brandon Amos, Itai Gat, Marton Havasi, Matthew J. Muckley, and Karen Ullrich. 2025. [Exact byte-level probabilities from tokenized language models for FIM-tasks and model ensembles](#). In *The Thirteenth International Conference on Learning Representations*.
- Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. 2023. No robots.
- Kevin Slagle. 2024. [Spacebyte: Towards deleting tokenization from large language modeling](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisanin, Alexan Ayrapetyan, and Igor Gitman. 2024. [Openmathinstruct-2: Accelerating AI for math with massive open-source instruction data](#). In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.
- Mathurin Videau, Badr Youbi Idrissi, Alessandro Leite, Marc Schoenauer, Olivier Teytaud, and David Lopez-Paz. 2025. [From bytes to ideas: Language modeling with autoregressive u-nets](#). *Preprint*, arXiv:2506.14761.
- Tim Vieira, Benjamin LeBrun, Mario Giulianelli, Juan Luis Gastaldi, Brian DuSell, John Terilla, Timothy J. O'Donnell, and Ryan Cotterell. 2025. [From language models over tokens to language models over characters](#). In *Forty-second International Conference on Machine Learning*.
- Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. [Knowledge fusion of large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Junxiong Wang, Tushaar Gangavarapu, Jing Nathan Yan, and Alexander M Rush. 2024. [Mambabyte: Token-free selective state space model](#). In *First Conference on Language Modeling*.
- Yunjia Xi, Jianghao Lin, Yongzhao Xiao, Zheli Zhou, Rong Shan, Te Gao, Jiachen Zhu, Weiwen Liu, Yong Yu, and Weinan Zhang. 2025. [A survey of llm-based deep search agents: Paradigm, optimization, evaluation, and challenges](#). *Preprint*, arXiv:2508.05668.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. [A survey on knowledge distillation of large language models](#). *Preprint*, arXiv:2402.13116.
- Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. [ByT5: Towards a token-free future with pre-trained byte-to-byte models](#). *Transactions of the Association for Computational Linguistics*, 10:291–306.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.

LILI YU, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. [MEGABYTE: Predicting million-byte sequences with multiscale transformers](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024a. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18164–18181.

Yu Zhang, Xiushi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024b. [A comprehensive survey of scientific large language models and their applications in scientific discovery](#). Preprint, arXiv:2406.10833.

A Evaluating the use of Linear Layers as Byte Level Heads

To test the effectiveness of a simple linear layer for each byte level head, we performed SFT *only at the byte level* on the Llama3.2 1B model (Meta, 2024) on a subset the TULU-3 dataset (Lambert et al., 2024). We then looked at training and validation losses over both bytes and tokens. We report the plots in Figure 2. We observe that, not only do the training and validation losses decrease smoothly for the byte level, but, surprisingly, they decrease also for the token level, demonstrating the effectiveness of adding even simple linear layers as heads for the byte level interface. This also indicates that a byte-level probability distribution can be effectively used for knowledge distillation – thus bridging a gap between different tokenizers with a common byte-level interface.

B Training Hyperparameters

We provide the values for the main hyperparameters used in our experiments, together with the respective search space for the tuning procedure in Table 4. The values for the baselines follow the optimized setup of Minixhofer et al. (2025), and only the learning rate has been further tuned due to computational constraints. For our method we tested different values of the weights for the loss functions.

C Approximation Settings for Byte-Probability Computations

The algorithm proposed by Vieira et al. (2025) provides an efficient approximation for computing byte-level probabilities from a token-level language

model. In this section we describe the approximation parameters used in our implementation and the empirical procedure used to select them.

C.1 Approximation Parameters

The algorithm introduces two parameters, K and ϵ , that control the trade-off between computational efficiency and approximation accuracy when estimating the byte-level probability

$$P_T(b_1, b_2, \dots, b_{N_b})$$

from a teacher model f_t operating over a token vocabulary (see Section 3.1).

Beam width (K). The algorithm performs a beam search over token sequences that are compatible with a given byte prefix. The beam width K specifies the maximum number of hypotheses retained during the search. Larger values of K allow more tokenization paths to be explored, which improves approximation accuracy but increases computational cost.

Pruning threshold (ϵ). During beam search, hypotheses with very small probability mass are removed. Specifically, beams whose probability falls below a threshold ϵ relative to the highest-probability beam are pruned. This pruning step eliminates tokenization paths that contribute negligibly to the final byte probability distribution.

Together, K and ϵ determine the number of tokenization paths considered during the computation.

C.2 Algorithm for Byte Probability Computation

The byte-level probability distribution at each position is computed using the following procedure:

1. **Initialization:** Create a beam state with parameters K (beam width) and ϵ (pruning threshold). The beam maintains a set of candidate tokenization paths, each with an associated probability weight.
2. **For each byte position i :**
 - **Compute distribution:** Call `logp_next()` to obtain the log probability distribution over the next 256 possible byte values. This operation marginalizes over all tokenization paths

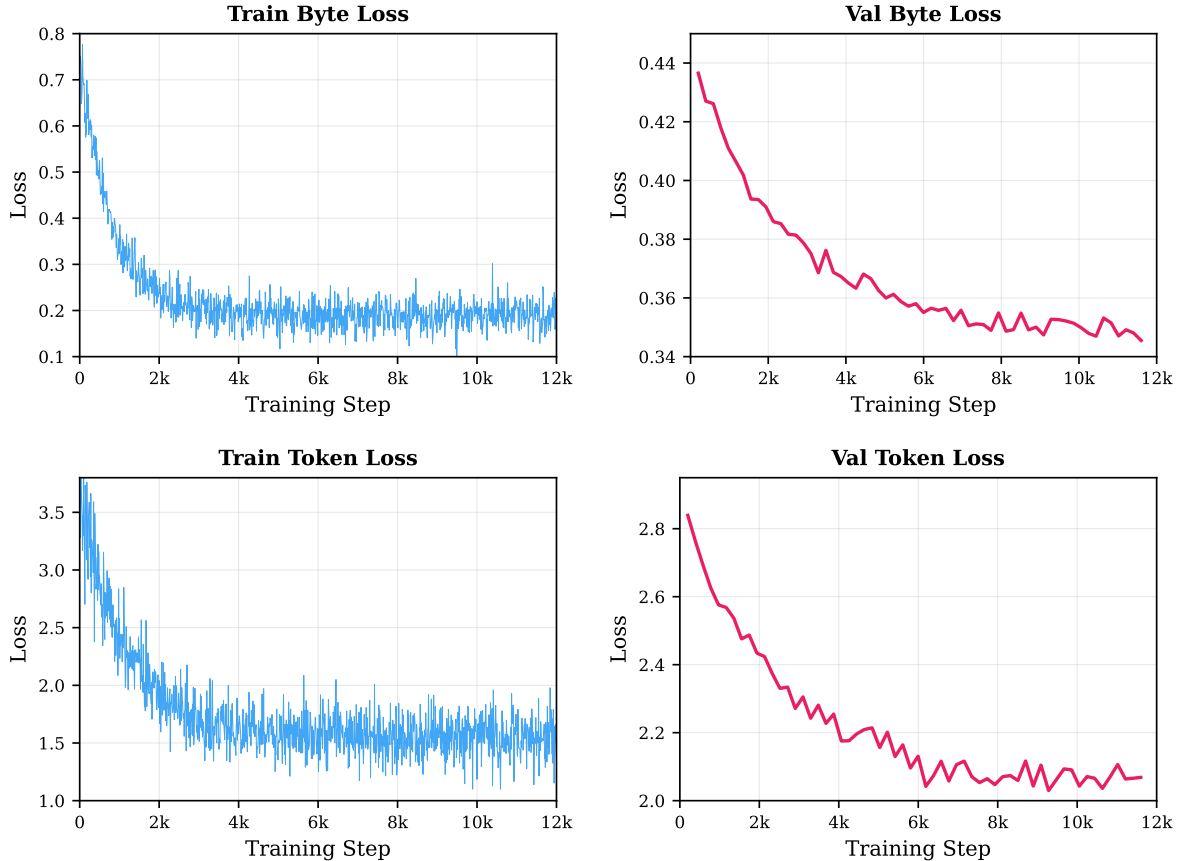


Figure 2: Training and validation losses for a Llama3.2-1B model with added byte-level head trained on a subset of the TULU-3 dataset with supervised fine-tuning only at the byte level. The top row plots are for training curves, while bottom row ones are for validation.

in the current beam:

$$\log P(b_i | b_{<i}) = \log \sum_{t \in \text{Beam}} P(t) \cdot P(b_i | t) \quad (6)$$

where t represents a tokenization path and $P(t)$ is its weight.

- **Advance beam:** Incorporate the observed byte b_i into the beam using the operation `beam.prune() << byte`. This extends each candidate path by consuming the byte.
- **Prune paths:** Remove tokenization paths with probability below the threshold ϵ relative to the highest-probability path. Retain at most K paths.
- **Handle token boundaries:** When a path completes a token, extend the beam by starting a new token using the teacher model’s next-token probabilities.

The key computational bottleneck is the teacher model inference at token boundaries. The beam

parameters K and ϵ control how many tokenization alternatives are maintained, which determines both accuracy and computational cost.

C.3 Evaluating Approximation Quality

We measure the Jensen–Shannon divergence (JSD) between the approximated byte probability distribution and a high-precision reference distribution computed using $K = 100, \epsilon = 10^{-6}$. Figure 3 shows the resulting approximation error for different combinations of K and ϵ . We observe that the setting $K = 10, \epsilon = 0.01$ achieves a Jensen–Shannon divergence of 0.0045. Figure 4 shows that runtime is primarily affected by the pruning threshold ϵ (lower values retain more beams), while beam width K has minimal impact due to efficient GPU batching of token queries. We also evaluate the effect of the approximation on downstream distillation performance by measuring the distilled model’s perplexity and task accuracy, confirming that configurations with $\text{JSD} < 0.005$ produce negligible performance degradation.

Hyperparameter	Value	Search Space
<i>LoRA</i>		
Rank (r)	64	—
Alpha (α)	64	—
Dropout	0.05	—
Target modules	q_proj, v_proj, k_proj, o_proj, gate_proj, up_proj, down_proj	—
<i>Optimiser</i>		
Algorithm	AdamW	—
Learning rate	2×10^{-5}	{5e-6, 1e-5, 2e-5, 3e-5, 5e-5, 1e-4}
Weight decay	0.01	—
(β_1, β_2)	(0.9, 0.95)	—
Gradient clipping (norm)	1.0	—
<i>Learning rate schedule</i>		
Scheduler	Cosine + linear warm-up	—
Warm-up steps	1,000	—
<i>Training</i>		
Epochs	5	—
Batch size (per device)	2	—
Gradient accumulation steps	4	—
Max sequence length	512	—
Precision	bf16-mixed	—
<i>Loss coefficients</i>		
KL divergence (λ_{KL})	0.1	{0.1, 0.2, 0.5, 0.8, 1.0}
Byte SFT (λ_b)	1.0	{0.5, 1.0}
<i>Byte-level decoder head</i>		
Parallel heads	10	—
Byte vocabulary size	261	—

Table 4: Training hyperparameters used in all experiments. The *Search Space* column lists the values explored during hyperparameter tuning; a dash indicates the value was fixed without search.

C.4 Experimental Setup

We conduct experiments using Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct as teacher models, with the Tulu-3 dataset for distillation. We test beam widths $K \in \{2, 5, 10, 20, 50, 100\}$ and pruning thresholds $\epsilon \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$, measuring runtime and JSD relative to the reference configuration for each setting.

C.5 Parallel Implementation

Our implementation achieves efficient throughput through multi-level parallelization. The dataset is partitioned into shards, with each shard processed by an independent worker (one per GPU). Within each worker, we use a process pool with

`n_sample_worker=15` to parallelize across samples, and the underlying trie operations batch up to 1000 token probability queries per forward pass. We use Python’s `asyncio` framework to overlap CPU preprocessing with GPU computation.

In our experiments using four NVIDIA RTX 3090 GPUs, the configuration $K = 10, \epsilon = 0.01$ achieves approximately 10.4 seconds per sample for 100–150 byte sequences. We choose this configuration because it provides excellent approximation accuracy ($\text{JSD} < 0.005$) while using $10\times$ less memory than the reference configuration ($K = 100$), enabling higher sample-level parallelism. The lower memory footprint allows us to process more samples concurrently, and the balanced pruning threshold $\epsilon = 0.01$ avoids both

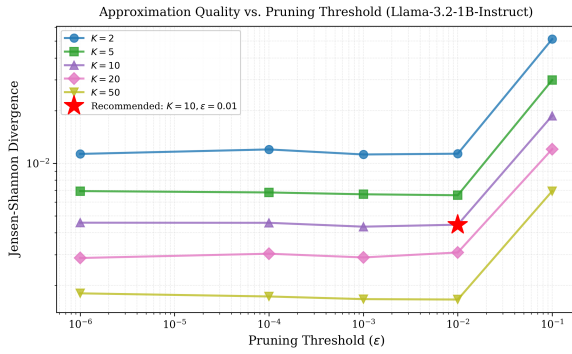


Figure 3: Jensen–Shannon divergence between approximated byte distributions and the reference distribution under different approximation settings.

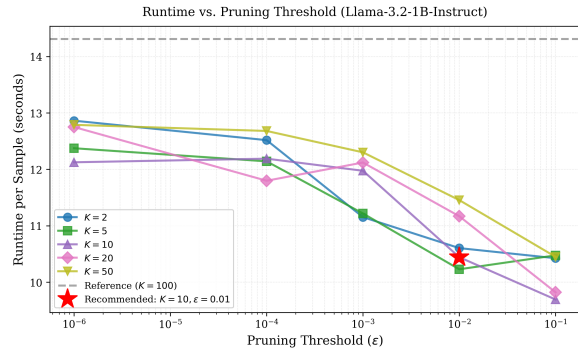


Figure 4: Runtime of byte-probability computation under different beam search configurations.

overly aggressive pruning (which degrades accuracy) and overly conservative retention (which increases memory usage). With this configuration and parallelism, computing byte probabilities for the entire Tulu-3 dataset requires approximately 2 days.

Fine-grained Readability Controlled Summarization of Scientific Documents via Control Vectors

Isabel Cachola

Johns Hopkins University
icachola@cs.jhu.edu

Kuleen Sasse

Johns Hopkins University
ksasse1@jh.edu

Mark Dredze

Johns Hopkins University
mdredze@cs.jhu.edu

Abstract

Plain Language Summarization (PLS) generates summaries of technical documents accessible to non-expert audiences. Readability – commonly used to evaluate PLS – has often been treated coarsely (expert vs. lay) although it exists on a spectrum with different levels for different readers. We propose a light weight control vector method for fine-grained readability control in scientific summarization along with a requirements-based framework for data selection. Our framework enforces: (1) readability levels differ substantially, and (2) paired examples share comparable content. Under this, control vectors enable more precise readability control than other popular methods.

1 Introduction

Effective scientific summarization must consider multiple factors, including both form and audience (Cohan et al., 2022; Yasunaga et al., 2019; Stefanou et al., 2024). When targeting non-expert readers, Plain Language Summarization (PLS) seeks to make complex scientific documents more accessible. This goal is particularly important given the significant public investment in research—approximately 40% of basic research in the United States is government-funded (Christopher V. Pece, 2024). To promote equitable access to publicly funded research, the U.S. National Institutes of Health now requires all NIH-funded papers to be open access.¹ However, while open access broadens availability, PLS ensures research is understandable to the general public.

Past work has found that users with higher, but non-expert, familiarity with a topic prefer more technical summaries and found lower complexity summaries less useful (August et al., 2024). However, most prior work in PLS has treated readability as a binary (plain versus technical), aiming for maximum readability, as measured by readability met-

¹NIH Public Access Policy

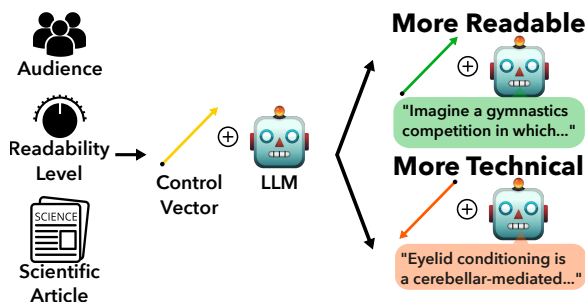


Figure 1: Visual Representation of our task.

rics such as Dale-Chall and Flesch-Kincaid Grade Level (Guo et al., 2021; Goldsack et al., 2022; Dale and Chall, 1948; Tanprasert and Kauchak, 2021). To address this gap, we explore fine-grained controls of readability for scientific document summarization using control vectors (Zou et al., 2025; Rimsky et al., 2024), a lightweight controllability method by modifying the model’s hidden states. Control vectors are derived from paired examples along a spectrum, capturing the difference in hidden states between the paired examples. During inference, they steer the model towards either end of the spectrum by applying a specified weight.

Control vectors can effectively steer complex attributes like honesty, making them well suited for readability control (Zou et al., 2025; Rimsky et al., 2024) and are less compute and data intensive than past work in readability controlled summarization (Ribeiro et al., 2023). However, despite their appeal, prior work has noted challenges in reliability of these vectors (von Rütte et al., 2024; Stickland et al., 2024; Tan et al., 2025; Braun et al., 2025). In this paper, we propose a requirements based system for data selection that improves on these failure points.

Our goal is to apply control vectors to the task of readability controlled summarization. We hypothesize that, in order for them to be effective, the extraction data needs to meet two Requirements (Req.): (1) separability of the paired examples and

(2) paired examples contain comparable content. Regarding Req. 1, if the paired examples are too similar in readability, the control vectors will not have enough information, and will therefore be ineffective. Regarding Req. 2, control vectors work by taking the difference between the hidden states of the positive and negative examples. If the content of the summaries differ, the resulting vector will encode factors beyond readability, reducing its effectiveness. Under these simple requirements, control vectors extracted from compliant data can interpolate between plain and technical summaries and generalize to other datasets.

We conduct an analysis of PLS datasets, and find that many popular datasets do not satisfy our proposed requirements. While we show that non-compliant data yields poor control vectors, such issues likely affect other summarization methods as well. Because control vectors is a low data method, applying our approach to a new tasks require collecting a small curated dataset.

Our work clarifies the data requirements for control vectors, increasing the chance of future success for new tasks. We release our inference code and dataset splits to support future work.²

2 Related Works

PLS has typically treated readability as a binary feature - plain or technical (Guo et al., 2021; Goldsack et al., 2022; Zaman et al., 2020). However, recent work has shown that preferred readability levels vary across different backgrounds (August et al., 2024). Current methods are limited, highlighting the need for additional research (Luo et al., 2022). Past work in controllable summarization has focused primarily on attributes other than readability, such as topic and aspect (Urlana et al., 2024; Zhang et al., 2022; He et al., 2022) and on the news domain, rather than scientific (Retkowski and Waibel, 2025; Chan et al., 2021). Finally, prior work in readability controllable summarization relies on compute and data intensive methods, such as reinforcement learning (Ribeiro et al., 2023).

To address this, we use control vectors, a representation-engineering method that manipulates hidden states during inference. Zou et al. (2025) introduced it for steering concepts like honesty. Several studies have identified limitations in control vectors (Bartoszcze et al., 2025; Wehner et al., 2025; von Rütte et al., 2024; Korznikov et al.,

2025; Stickland et al., 2024). Only a few works (Tan et al., 2025; Braun et al., 2025) have analyzed the underlying causes. None have proposed systematic methods for improving data quality, a gap our work addresses through our framework.

3 Method

Extracting control vectors. We begin with a small extraction dataset of paired examples. Each example contains a lay summary s^+ and technical summary s^- of the same source paper. There are 3 steps to retrieve the control vectors: (1) Extraction, (2) Contrastive Combination, and (3) Reduction. For step (1) Extraction, we pass each example pair (s_i^+, s_i^-) through the model and retrieve the last non-padding hidden state $([h_i^+, h_i^-])$. Step (2) Contrastive Combination requires us to combine the paired hidden states to a single vector. We experiment taking the difference $c_i := h_i^+ - h_i^-$ and the mean $c_i := (h_i^+ + h_i^-)/2$ of the 2 vectors, as in Luo et al. (2022). We refer to these options as Diff and Center, respectively. Step (3) Reduction requires we then combine the vectors of each paired example in the extraction dataset. Luo et al. (2022) uses PCA $c := \text{PCA}(c_i)$, for this step. We also experiment with simply averaging the vectors $c := \text{mean}(c_i)$. At inference time, we add these control vectors multiplied by a scalar strength to the hidden states of the model at a certain layer(s).

Requirements of extraction data. We hypothesize that extracting effective readability control vectors requires the dataset to satisfy two requirements (Req.): (1) positive and negative examples must differ sufficiently in readability and (2) they must share roughly the same content. For Req. 1, if the examples are too close in readability (i.e. the lay summary is not significantly more readable than the technical summary), their hidden states will not differ enough, resulting in low controllability. For Req. 2, when we combine the hidden states of the positive and negative examples, we are subtracting (or averaging) the content of the summary and the resulting vector represents the readability. If the content of the paired summary is too different (e.g. the lay summary contains quotes from the authors that are not present in the technical summary see Table 12) the vector will capture information beyond readability and fail for fine-grained control. We use these requirements to select the extraction dataset in §5 for the main results (§6) and explore non-compliant datasets in §6.

²ANONYMOUS REPO FOR REVIEW

4 Experimental Setup

We use Llama-3.1-8B-Instruct, google/gemma-7b-it, and mistralai/Mistral-7B-Instruct-v0.3 as they are open-source instruction tuned models (Grattafiori et al., 2024; Team et al., 2024). We opt to use smaller models for broader accessibility and lower computational costs. We compare the ability of the control vector methods to In-Context Learning (ICL), Supervised Fine Tuning (SFT), LoRa/QLoRa SFT and DSPy Prompt Optimization. ICL has shown great abilities in instruction following, and is a similarly low data, lightweight method (Brown et al., 2020). For ICL, we take a positive and negative example from the extraction set (§5) as examples of a lay summary and technical summary, respectively. We then provide the document to the model and ask it to summarize the document with a specified readability level. We repeat this with 5 levels of specified readability. SFT and QLoRa/LoRa SFT based methods (Hu et al., 2021; Dettmers et al., 2023). Prompt Optimization has shown promise as a way to control language models without expensive training (Khatab et al., 2023). See §A and §B for additional details.

Past work has shown that the Coleman-Liau readability index (CLI) (Coleman and Liau, 1975) and Dale-Chall readability scores (DCRS) (Dale and Chall, 1948) correlate the highest with human judgments of readability (Cachola et al., 2025). To avoid overfitting to our evaluation, we use DCRS for our data selection (§5) and CLI for our system evaluation (§6). To evaluate system performance, we report the Pearson (Pearson, 1895) and Kendall-Tau (Kendall, 1938) correlation of the specific strength scalar with CLI. CLI provides a lower score for higher readability, while our setup treats the lay summary as the positive example. In other words, we expect higher control vector strengths to produce lower CLI scores. Therefore, we multiply the CLI scores by -1 so the resulting correlations are positive. We also use BERTScore (Zhang et al., 2020) with the abstract as the reference, since some dataset lay summaries contain extraneous information not present in the original document and thus make poor evaluation references.

5 Meeting Data Requirements

We begin with four different scientific summarization datasets: eLife (Goldsack et al., 2022), Eureka (Zaman et al., 2020), PLOS (Goldsack et al., 2022), and SciNews (Liu et al., 2024). These

datasets are designed for scientific PLS, allowing us to use the lay summary as the positive example and the abstract as the technical, negative example. Additional dataset descriptions in §D.

We choose an extraction dataset that follows the requirements outlined in §3. Req. 1 states that the paired examples must be sufficiently different in readability. We plot the Dale-Chall scores of the positive and negative examples for each dataset in a histogram. We additionally calculate the Bhattacharyya distance, which measures the overlap between two distributions (Bhattacharyya, 1943). A dataset that best meets this requirement will have a large visual separation between the two distributions, as well as a high Bhattacharyya distance. Figure 2 contains both the histograms and the Bhattacharyya distance for each dataset. eLife has both the highest visual separation and highest Bhattacharyya distance. PLOS has a particularly low separation, indicating that the PLOS lay summaries are not significantly more readable than the technical summaries.

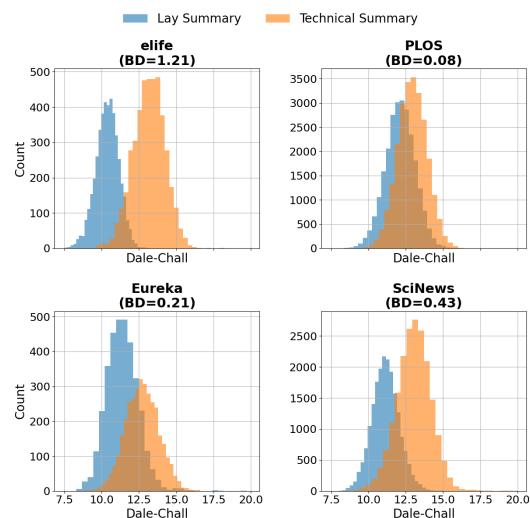


Figure 2: Histogram of the Dale-Chall readability scores for the lay and technical summaries and the Bhattacharyya distance (BD) of each dataset.

Req. 2 states that the positive and negative examples must have approximately the same content, only differing in their readability level. In order to measure overlap in content, we use Sentence Transformers to retrieve an embedding matrix for the example pairs, then compute the cosine similarity (Reimers and Gurevych, 2019). We use Specter for the embeddings, a model trained on scientific data (Cohan et al., 2020). The resulting mean cosine similarity scores are as follows: 0.631 for eLife, 0.604 for PLOS, 0.575 for Eureka, and 0.590 for SciNews. Based on these scores,

we conclude that eLife and PLOS have the highest content similarity while Eureka and SciNews have the lowest. This aligns from the construction of the datasets: eLife and PLOS’s lay summaries written by the original paper authors to be accessible explanations of their papers while Eureka and SciNews’s summaries are news reports discussing papers. Upon inspection, we find that Eureka and SciNews often contain information not present in the abstract, such as interviews with the authors or information on a study’s funding source. We provide examples from each dataset in §E. Based on this analysis, we conclude that eLife best meets the requirements. As the control vector method does not require a large extraction dataset, we select 128 of the most readable examples (lowest Dale-Chall) from the eLife training set. Choosing the most readable examples gives the control vectors the best examples of readability. We use this setup to report our main results in §6.

6 Results

Dataset	Model	Method	CC	Red.	PC	KT	BertS
eLife	LLaMa	BL			0.015	0.017	0.840
		CV	Center	Avg	0.730	0.563	0.803
	Gemma	BL			0.000	0.000	0.820
		CV	Center	PCA	0.063	0.023	0.813
	Mistral	BL			0.000	0.000	0.820
		CV	Diff	PCA	0.315	0.223	0.821
PLOS	LLaMa	BL			0.055	0.019	0.831
		CV	Center	Avg	0.725	0.553	0.805
	Gemma	BL			-0.002	-0.003	0.833
		CV	Center	PCA	0.106	0.065	0.821
	Mistral	BL			0.000	0.000	0.833
		CV	Diff	PCA	0.213	0.189	0.830
Eureka	LLaMa	BL			0.000	-0.003	0.848
		CV	Center	Avg	0.770	0.607	0.810
	Gemma	BL			0.000	0.000	0.874
		CV	Center	PCA	0.075	0.042	0.895
	Mistral	BL			0.000	0.000	0.874
		CV	Center	PCA	0.121	0.098	0.899
SciNews	LLaMa	BL			0.030	-0.004	0.840
		CV	Center	Avg	0.674	0.573	0.801
	Gemma	BL			0.002	0.001	0.820
		CV	Center	Avg	0.129	0.110	0.821
	Mistral	BL			0.002	0.001	0.820
		CV	Center	Avg	0.141	0.177	0.809

Table 1: Pearson Correlation (PC) and Kendall-Tau (KT) correlation of the specified level of readability with the Coleman-Liau readability index, and BertScore F1 (BertS). Reported is the best baseline (BL). We report the best performing CV setting for Contrastive Combination (CC) and Reduction (Red.)

We report best settings per dataset in Table 1 and full ablations in §F. ICL, SFT, QLoRA/LoRA SFT, and prompt optimization perform poorly, with ICL and prompt optimization aligning with prior work showing they are insufficient for readability control (Ribeiro et al., 2023). Training-based methods also underperform, due to dataset differences (scientific texts vs. shorter news and fewer examples). Ap-

pendix F for more analysis and example outputs in Tables 13a–13c.

For most settings, using Center for the Contrastive Combination and Avg for the Reduction step performs best. Although the control vectors use eLife for the extraction dataset, the best control vectors generalize well to the other datasets, achieving similarly high correlations. This suggests that the resulting control vectors are in fact representing readability, rather than dataset specific information. See Appendix F for full results, and Appendices F.1 and G for resource usage and latency overhead. Example outputs in Appendix H.

Analysis of extraction dataset requirements.

We conduct experiments using the non-compliant datasets. We extract control vectors using the 128 most readable summaries from each train dataset, similar to the process in §5. For evaluation, we randomly sample 248 examples from each test set. At inference, we use the control vectors extracted from each datasets’ respective extraction set. This is an easier task than that presented above as it does not test generalization to other datasets. We use Center as our Contrastive Combination method and Avg as our reduction method, the best performing setting above. Results are below:

Dataset	Pearson	KT
eLife	0.302	0.205
PLOS	0.099	0.112
Eureka	-0.195	-0.137
SciNews	0.083	0.051

Table 2: Correlation of non-compliant datasets with specified readability level and CLI score.

We find that the non-compliant datasets perform poorly. Eureka results in a moderate negative correlation, likely a result of the extraneous information present in the lay summaries, which can increase the readability scores. SciNews and PLOS have low correlation scores, indicating poor controllability. This contrasted with the high performance presented in Table 1 provides evidence for the validity of our requirement framework.

7 Conclusion

We propose control vectors as a lightweight, low-data, resource-efficient method for readability-controlled summarization of scientific articles. We also introduce a requirement-based data selection framework and show that control vectors extracted from compliant data effectively control readability, providing guidance for future tasks and data collection.

8 Limitations

We identify a few key limitations in our work. Our work does not verify that control vectors are latent representations of readability, rather we focus on the downstream effects of using control vectors. Further research is required to understand what information is encoded in control vectors. Our experiments are only tested on English-language data in the science domain. We focus our experimentation on smaller models and results may not generalize to larger models. Although our method is not specific to these facts, further analysis is needed to test for generalizability across domains and languages. Finally, our evaluation relies on automatic readability metrics, which are imperfect measures of readability.

9 Ethics

Our work focuses on better controlling the outputs of language models. Although the majority of research in this area focuses on general controllability (Keskar et al., 2019; Chen et al., 2024; Yang and Klein, 2021) and AI Safety features (Zou et al., 2025; Rinsky et al., 2024), the same methods could be used for nefarious tasks, such as generating purposely dishonest language (Barman et al., 2024). However, we believe that better controllability of language models will ultimately lead to safer models as we can better prevent undesired outputs. Additionally, our work specifically focuses on improving readability controlled summarization of scientific documents, with the goal of improving access to scientific discovery for the general public. Therefore, we believe the benefits of our work outweigh the risk.

References

- Tal August, Kyle Lo, Noah A. Smith, and Katharina Reinecke. 2024. [Know your audience: The benefits and pitfalls of generating plain language summaries beyond the "general" audience](#). *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- Dipto Barman, Ziyi Guo, and Owen Conlan. 2024. [The dark side of language models: Exploring the potential of llms in multimedia disinformation generation and dissemination](#). *Machine Learning with Applications*, 16:100545.
- Lukasz Bartoszcze, Sarthak Munshi, Bryan Sukidi, Jennifer Yen, Zejia Yang, David Williams-King, Linh Le, Kosi Asuzu, and Carsten Maple. 2025. [Representation engineering for large-language models: Survey and research challenges](#).
- A. Bhattacharyya. 1943. [On a measure of divergence between two statistical populations defined by their probability distributions](#). In *Bulletin of the Calcutta Mathematical Society*.
- Joschka Braun, Carsten Eickhoff, David Krueger, Seyed Ali Bahrainian, and Dmitrii Krasheninnikov. 2025. [Understanding \(un\)reliability of steering vectors in language models](#).
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). *Advances in Neural Information Processing Systems*, 33:1877–1901.
- Isabel Cachola, Daniel Khashabi, and Mark Dredze. 2025. [Evaluating the evaluators: Are readability metrics good measures of readability?](#) *ArXiv*, abs/2508.19221.
- Hou Pong Chan, Lu Wang, and Irwin King. 2021. [Controllable summarization with constrained markov decision process](#). *Transactions of the Association for Computational Linguistics*, 9:1213–1232.
- Yihan Chen, Benfeng Xu, Quan Wang, Yi Liu, and Zhendong Mao. 2024. [Benchmarking large language models on controllable generation under diversified instructions](#). In *AAAI Conference on Artificial Intelligence*.
- Gary W. Anderson Christopher V. Pece. 2024. [Analysis of federal funding for research and development in 2022](#).
- Arman Cohan, Guy Feigenblat, Dayne Freitag, Tirthankar Ghosal, Drahomira Herrmannova, Petr Knoth, Kyle Lo, Philipp Mayr, Michal Shmueli-Scheuer, Anita de Waard, and Lucy Lu Wang. 2022. [Overview of the third workshop on scholarly document processing](#). In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 1–6, Gyeongju, Republic of Korea. Association for Computational Linguistics.
- Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. [SPECTER: Document-level representation learning using citation-informed transformers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.
- Meri Coleman and Ta Lin Liau. 1975. [A computer readability formula designed for machine scoring](#). *Journal of Applied Psychology*, 60:283–284.

- Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [Qlora: Efficient finetuning of quantized llms](#).
- Hantian Ding, Zijian Wang, Giovanni Paolini, Varun Kumar, Anoop Deoras, Dan Roth, and Stefano Soatto. 2024. [Fewer truncations improve language modeling](#).
- Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. 2022. [Making science simple: Corpora for the lay summarisation of scientific literature](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10589–10604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#).
- Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. 2021. Automated lay language summarization of biomedical scientific reviews. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 160–168.
- Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2022. [CTRLsum: Towards generic controllable text summarization](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5879–5915, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#).
- M. G. Kendall. 1938. [A new measure of rank correlation](#). *Biometrika*, 30:81–93.
- Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. [Ctrl: A conditional transformer language model for controllable generation](#). *ArXiv*, abs/1909.05858.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. [Dspy: Compiling declarative language model calls into self-improving pipelines](#).
- Anton Korznikov, Andrey Galichin, Alexey Dontsov, Oleg Y. Rogov, Ivan Oseledets, and Elena Tutubalina. 2025. [The rogue scalpel: Activation steering compromises llm safety](#).
- Dongqi Liu, Yifan Wang, Jia Loy, and Vera Demberg. 2024. [SciNews: From scholarly complexities to public narratives – a dataset for scientific news report generation](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14429–14444, Torino, Italia. ELRA and ICCL.
- Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2022. [Readability controllable biomedical document summarization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4667–4680, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. [Optimizing instructions and demonstrations for multi-stage language model programs](#).
- Karl Pearson. 1895. [VII. note on regression and inheritance in the case of two parents](#). *Proceedings of the Royal Society of London*, 58:240 – 242.
- Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. [Zero: Memory optimizations toward training trillion parameter models](#).
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Fabian Retkowsky and Alexander Waibel. 2025. [Zero-shot strategies for length-controllable summarization](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 551–572, Albuquerque, New Mexico. Association for Computational Linguistics.
- Leonardo F. R. Ribeiro, Mohit Bansal, and Markus Dreyer. 2023. [Generating summaries with controllable readability levels](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11669–11687, Singapore. Association for Computational Linguistics.
- Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. [Steering llama 2 via contrastive activation addition](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

- Loukritia Stefanou, Tatiana Passali, and Grigorios Tsoumakos. 2024. Auth at biolaysumm 2024: Bringing scientific content to kids. In *Proceedings of the ACL 2024 BioNLP Workshop*, Bangkok, Thailand. A paper presented at the BioLaySumm 2024 shared task on lay summarization of biomedical research articles.
- Asa Cooper Stickland, Alexander Lyzhov, Jacob Pfau, Salsabila Mahdi, and Samuel R. Bowman. 2024. [Steering without side effects: Improving post-deployment control of language models](#).
- Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. 2025. Analysing the generalisation and reliability of steering vectors. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA. Curran Associates Inc.
- Teerapaun Tanprasert and David Kauchak. 2021. [Flesch-kincaid is not a text simplification evaluation metric](#). In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 1–14, Online. Association for Computational Linguistics.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. [Gemma: Open models based on gemini research and technology](#).
- Ashok URLana, Pruthwik Mishra, Tathagato Roy, and Rahul Mishra. 2024. [Controllable text summarization: Unraveling challenges, approaches, and prospects - a survey](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1603–1623, Bangkok, Thailand. Association for Computational Linguistics.
- Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. 2024. [A language model’s guide through latent space](#).
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Galouédec. 2020. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>.
- Jan Wehner, Sahar Abdelnabi, Daniel Tan, David Krueger, and Mario Fritz. 2025. [Taxonomy, opportunities, and challenges of representation engineering for large language models](#).
- Chenxi Whitehouse, Fantine Huot, Jasmijn Bastings, Mostafa Dehghani, Chu-Cheng Lin, and Mirella Lapata. 2024. [Low-rank adaptation for multilingual summarization: An empirical study](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1202–1228, Mexico City, Mexico. Association for Computational Linguistics.
- Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. 2025. [Axbench: Steering llms? even simple baselines outperform sparse autoencoders](#).
- Kevin Yang and Dan Klein. 2021. [FUDGE: Controlled text generation with future discriminators](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.
- Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. [Scisummnet: a large annotated corpus and content-impact models for scientific paper summarization with citation networks](#). In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI’19/IAAI’19/EAAI’19*. AAAI Press.
- Farooq Zaman, Matthew Shardlow, Saeed-Ul Hassan, Naif R. Aljohani, and Raheel Nawaz. 2020. [HTSS: A novel hybrid text summarisation and simplification architecture](#). *Inf. Process. Manag.*, 57:102351.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#).
- Yusen Zhang, Yang Liu, Ziyi Yang, Yuwei Fang, Yulong Chen, Dragomir R. Radev, Chenguang Zhu, Michael Zeng, and Rui Zhang. 2022. [Macsum: Controllable summarization with mixed attributes](#). *Transactions of the Association for Computational Linguistics*, 11:787–803.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2025. [Representation engineering: A top-down approach to ai transparency](#).

A Control Vector Implementation Details

We extract the control vectors for layers 16, 17, 18 of the model, as middle layers were shown to have the best effect on the downstream output (Rimsky et al., 2024). We use $\{-1, -0.5, 0, 0.5, 1\}$ as the set of strengths and set the temperature to 0.0. In preliminary experiments, we found that using a control vector strength greater than $|1|$ caused degeneration.

For the main results reported in Table 5, Table 6 and Table 7, we randomly sample 1024 examples from the PLOS and SciNews test sets for evaluation. We use the full eLife and Eureka test datasets for evaluation, as they contain less than 1024 examples (241 and 1010 respectively). We ran all of our experiments on a grid with NVIDIA A100 and H200 GPUs available. Each experiment only requires a single A100 GPU.

B Baselines Implementation Details

We compared our control-vector method against commonly used alternatives, including In Context Learning (ICL) (Brown et al., 2020), Supervised Fine-Tuning (SFT), LoRA/QLoRA based Supervised Fine-Tuning (Hu et al., 2021; Dettmers et al., 2023), and Prompt Optimization.

B.1 ICL Prompts

We use the following prompt for our ICL baseline, based on the prompts used in (Ribeiro et al., 2023):

```
Summary 0 properties: 0% readability \n
Summary 0: <TECHNICAL SUMMARY> \n \n
Summary 1 properties: 100% readability \n \n
Summary 1: <LAY SUMMARY> \n \n
Summarize the input document with the following properties: <0,25,50,75,100>% readability \n
Input document: <SOURCE> \n
Summary:
```

We use the following prompt during extraction of the control vectors:

```
Document: <SOURCE> \n Summarize: <TARGET>
```

We use the following prompt during inference using the control vectors:

```
Document: <SOURCE> \n Summarize:
```

B.2 ICL Shot Selection

We experiment with the effect of number of examples provided to the model and its effect on control-

ability. In our setup, a “shot” is a pair of positive and negative summaries, so for each setting a N -shot experiment contains $2 * N$ total summaries. We experiment with $N = [1, 8, 64, 128]$ shots to match up to the total number of example pair provided to the control vector experiments. The shots are randomly sampled from the training dataset and we randomly sample 100 examples from the test set for inference. We use LLaMa 8b, the best performing model in Table 1. The results are in Table 3.

Dataset	N	Pearson	Kendall	BertS F1
Elife	1	-0.066	-0.039	0.845
	8	-0.023	-0.013	0.841
	64	0.016	0.006	0.838
	128	0.027	0.022	0.841
Eureka	1	-0.062	-0.035	0.836
	8	-0.018	-0.017	0.831
	64	-0.001	0.001	0.834
	128	0.038	0.035	0.844
PLOS	1	-0.030	-0.021	0.847
	8	-0.023	-0.014	0.838
	64	0.011	0.004	0.837
	128	-0.019	-0.007	0.843
SciNews	1	-0.028	-0.024	0.835
	8	-0.009	-0.009	0.831
	64	-0.001	-0.011	0.830
	128	-0.013	-0.007	0.834

Table 3: Results using In-Context Learning, varying the number of shots. Additional shots do not generally increase performance.

The number of shots provided to for In-Context Learning does not have a significant effect on the overall controllability or quality of the summaries. Therefore, to save compute by number of tokens, we use $N = 1$ for the ICL experiments in the main body of results, reported in Section 6.

B.3 SFT

To train our model using supervised fine-tuning (SFT), we leveraged the datasets originally generated for learning control vectors. For each source document, we included both the positive and negative examples as separate training instances paired with the same reference summary, effectively doubling the size of the training set.

```
Document: <SOURCE>\n Summarize with readability level <READABILITY>.\n Summary: <SUMMARY>\n \n
```

We used the prompt shown above for training. For each example, we computed the Dale–Chall readability score of the reference summary and used this value as the target readability level specified in the prompt. The model was trained using

a causal language modeling loss applied only to the summary portion of the output, implemented using the TRL framework (von Werra et al., 2020). During training, we did not apply output truncation and instead relied on the maximum input length supported by the base model.

B.4 LoRA/QLoRA SFT

Our second baseline uses LoRA and QLoRA adapters for parameter-efficient fine-tuning. The training procedure is identical to SFT, except that only adapter parameters are updated. We primarily use LoRA; when memory constraints prevent LoRA training, we instead employ 4-bit QLoRA to enable training under reduced memory usage.

B.5 Prompt Optimization

For prompt optimization, we used the DSPy (Khattab et al., 2023) framework to automatically select an optimized prompt. We provided DSPy with the same training examples and initial prompt used for SFT. The optimization objective was the mean squared error between the Dale–Chall readability score of the generated summary and the target readability specified in the prompt. We performed optimization using the MiPROv2 (Opsahl-Ong et al., 2024) optimizer with default hyperparameters.

C Baseline Experimental Procedure

C.1 Datasets

We attempt use the sub-selected the 128 most readable summaries dataset described in the main paper.

C.2 Hyperparameters Used

We took hyperparameters from the AxBench Paper (Wu et al., 2025) for the LoRa-SFT and SFT as they provided a fair comparison between Control Vectors, SFT, and Prompting-based Methods. However, we modified the number of epochs and batch size as our datasets were much longer in both number of items and length of inputs. In Table 4, we show the hyperparameters used for each baseline method.

C.3 Inference and Evaluation

For SFT, QLoRa/LoRa SFT and DSPy Prompt Optimization, we tested the models ability by telling the model to provide summaries at readability scores of 1,3,5,7,9 mirroring the range of Dale Chall, evaluate the models ability to change the readability similarly to the control vectors.

D Dataset Details

The eLife dataset consists of 5,000 full-text biomedical research articles paired with non-technical lay summaries published by the eLife scientific journal. The Eureka dataset is derived from the EurekaAlert corpus provided by the HTSS project. It contains 5,000 general scientific articles along with corresponding simplified lay summaries written as news reports for the public. The PLoS dataset comprises of around 40,000 full-length biomedical research articles paired with expert-written lay summaries, sourced from various journals published by the Public Library of Science (PLOS). The SciNews dataset includes over 40,000 scientific articles from nine distinct domains paired with a corresponding news report written for a lay audience.

E Dataset Examples

We include examples of each dataset in Table 9 (eLife), Table 10 (PLOS), Table 11 (Eureka), and Table 12. In the eLife example, the lay summary is significantly more readable than the technical summary, using less technical language, while only focusing on the content of the paper. This is an example of a pair of summaries that are compliant with the requirements outlined in §3. In the PLOS example, although it is clear that both summaries cover the same material, the lay summary is not significantly more readable than the technical summary, including highly complex terms such as “glycosaminoglycans.” This example meets Req. 2 but not Req. 1. The Eureka and SciNews examples contain extraneous information not present in the technical summary. Both example lay summaries contain interviews with the papers’ authors, meaning these summaries do not meet Req. 2. We note that although this extraneous information is problematic for control vectors, it is also likely problematic for any method. At best, the extraneous information will make a method ineffective. At worst, it could teach a model to hallucinate interviews that did not happen. For this reason, we urge future researchers to use caution using these datasets for summarization training.

F Full Results

We report the results of all the ablations in Table Table 5, Table 6 and Table 7. Overall, we see that the baselines do not perform as well as the control vectors. We also see that some of the methods

Hyperparameter	Baseline Method	LoRa	QLoRa	SFT	DSPy
Learning rate		5.00E-03	5.00E-03	4.00E-05	–
Epochs		3	3	3	–
Quantization		–	4-bit	–	–
Batch size		1	1	1	–
LoRA rank		4	4	–	–
LoRA alpha		32	32	–	–
LoRA dropout		0.1	0.1	–	–
Target modules		o_proj	o_proj	–	–
Layers to transform		12, 20, 31, 39	12, 20, 31, 39	–	–
Optimizer		AdamW	AdamW	AdamW	MIPROv2
Max bootstrapped demos		–	–	–	4
Max labeled demos		–	–	–	4
Auto		–	–	–	medium

Table 4: Hyperparameters for each baseline method.

are either unsupported or cannot run due to out of memory errors.

We see that Avg works the best as a reduction setting for all datasets except SciNews. This holds for both Diff and Center as the Contrastive Combination setting. For eLife, PLOS, and Eureka, Center achieves slightly better performance, although the difference is small. In general, the choice of reduction method makes the largest difference in terms of performance. While the Center-Avg settings perform best for 3 of the 4 datasets, SciNews achieving better performance with Diff-PCA indicates this setting is not universally optimal. Some tuning of the control vector settings may be necessary, although the overhead of this tuning is minimal compared to that of higher compute methods, such as finetuning, prompt optimization, and reinforcement-learning.

The failure of the training methods does not mean they did not learn how to summarize but rather they were unable to learn to control the readability of the summaries.

F.1 Discussion

The baselines were less simple to run than the control vectors leading to many four major issues: Resource Usage, Out of Memory Errors, and Lack of Support Out of the Box, Small Training Set.

Resource Usage For DSPy, it also used way more resources than control vectors while performing worse. The DSPy method we use uses in context learning which forces the prompt to include extremely long examples. This could lead to some of the sources not being able to be summarized as the total length of the prompt was more than the max context length of the model. This was the

main issue with DSPy for these smaller model.

As for SFT and QLoRa/LoRa-SFT, it requires the more forward passes as finetuning is multiple epochs with the same amount of tokens. In addition, it requires expensive backward passes that require more memory. Even with this more computation, our method performs better while keeping training costs down.

Out of Memory Errors Many of the training based methods required hefty amounts of resources and very long source documents led to out of memory errors with even QLoRa. While truncation could have prevented these errors, truncating at an arbitrary length could harm the overall plain language summary as this increases problems with reading comprehension, adds more hallucinations, and more (Ding et al., 2024). While smarter engineering could have performed with other training optimizations, the amount of work to prevent these errors would have dwarfed the time it takes to set up the control vectors and could have led to even slower training times with the increase in communication time between devices and loading from RAM (Rajbhandari et al., 2020). In addition, Gemma experienced more out of memory errors with less parameters due to a different less memory efficient architecture.

Lack of Support Out of the Box We could not run DSPy prompt optimization on the Gemma model due to an incompatibility between DSPy and the Gemma system prompt. It would have required modifying DSPy for specific set of models by adding additional adapters not available out of the box.

Small Training Set We hypothesize that SFT performed worse in controllability than LoRA due to the number of trainable parameters relative to the size of the training dataset. Our SFT model was trained on only the 128 most readable examples, which likely provided insufficient signal to effectively update the large number of parameters involved in full fine-tuning. In contrast, LoRA introduces far fewer trainable parameters, making it better suited for small-data regimes. This is consistent with findings found in (Whitehouse et al., 2024) where they find that "in low-data scenarios, LoRA is a better alternative to full fine-tuning."

At first glance, this result may appear to contradict the findings of (Ribeiro et al., 2023). However, their experiments were conducted with a substantially larger training dataset. Additionally, their source documents consisted primarily of news articles, which are typically much shorter and less structurally complex than scientific articles, reducing the difficulty of the summarization task.

G Latency testing

We conduct latency tests on the added overhead of using control vectors over standard LM inference. We randomly generate input IDs and measure the latency of the forward pass, as this is the portion of the generation pipeline where control vectors are used and have the potential to add overhead. We experiment with 3 batch sizes (1, 4, 16) and 3 sequence lengths (512, 1024, 2048) for a total of 9 ablations. For each setting, we run 32 warm-up runs then measure the latency for 1024 inference calls. We report the mean latency in milliseconds per sample and per token. We additionally report the Δ latency and percent overhead. All the latency tests are run using Llama 3.1 8b Instruct (meta-llama/Llama-3.1-8B-Instruct) on a single NVIDIA A100 GPU, using half-precision floating-point. The results are reported in Table 8. We find that using control vectors adds approximately a 5% latency overhead when measured per sample or per token. We believe this difference is negligible when compared to other controllable generation methods, that require expensive training or hyper-parameter searches, or the additional token processing required for In-Context Learning.

H Example Generations

We present example outputs for the SFT, LoRA, and DSPy baselines in Table 13. Example outputs

for the ICL baseline in Table 14 and example outputs for CV method in Table 15. From the outputs, SFT and DSPy generally fail to produce reasonable generations, as discussed in Appendix F.1. LoRA produces reasonable summaries, but the readability of the summaries does not vary with the desired readability level, generally always producing technical summaries. We see a similar pattern for the ICL baseline; the summaries are reasonable but there is minimal controllability for the readability levels. The Control Vector outputs indicate higher controllability. The more technical summary references specific technical details from the paper while the lay summary focuses on the bigger picture of "The researchers are looking at how to live longer by looking at what is in the gut."

	Method	Contrastive Combination	Reduction	Pearson	Kendall Tau	BertScore F1
eLife	ICL	-	-	-0.025	-0.017	0.847
	DSPy	-	-	0.000	0.000	0.000
	QLoRa SFT	-	-	-0.115	-0.164	0.847
	SFT	-	-	0.033	0.035	0.780
		Diff	Avg	0.302	0.205	0.837
	Control	Center	Avg	0.302	0.205	0.803
	Vectors	Diff	PCA	0.101	0.075	0.836
		Center	PCA	0.101	0.075	0.835
PLOS	ICL	-	-	0.00024	-0.003	0.848
	DSPy	-	-	0.000	0.000	0.000
	LoRa SFT	-	-	-0.084	-0.126	0.819
	SFT	-	-	0.0074	0.004	0.719
		Diff	Avg	0.416	0.286	0.805
	Control	Center	Avg	0.417	0.287	0.805
	Vectors	Diff	PCA	0.365	0.249	0.838
		Center	PCA	0.144	0.107	0.837
Eureka	ICL	-	-	0.055	0.0194	0.853
	DSPy	-	-	0.000	0.000	0.000
	LoRa SFT	-	-	-0.034	-0.036	0.846
	SFT	-	-	0.0078	0.0095	0.764
		Diff	Avg	0.407	0.298	0.840
	Control	Center	Avg	0.408	0.298	0.810
	Vectors	Diff	PCA	0.062	0.043	0.839
		Center	PCA	0.168	0.1212	0.838
SciNews	ICL	-	-	0.0302	-0.004	0.840
	DSPy	-	-	0.000	0.000	0.000
	LoRa SFT	-	-	-0.048	-0.053	0.480
	SFT	-	-	-0.0062	-0.0059	0.44
		Diff	Avg	0.075	0.000	0.827
	Control	Center	Avg	0.268	0.216	0.801
	Vectors	Diff	PCA	0.395	0.315	0.826
		Center	PCA	0.081	0.074	0.826

Table 5: Full set of results for all the Control Vector settings we tested for meta-llama/Llama-3.1-8B-Instruct. We report the Pearson and Kendall-Tau correlation of the specified level of readability with the Coleman-Liau readability index. We experiment with In-Context Learning (ICL), Supervised Finetuning (SFT), QLoRa/LoRa SFT, and DSPy Prompt Optimization (DSPy) as our baselines. For the Control Vectors, we experiment with 2 Contrastive Combination methods (Diff and Center) and 2 Reduction methods (Avg. and PCA).

	Method	Contrastive Combination	Reduction	Pearson	Kendall Tau	BertScore F1
eLife	DSPy	-	-	CONTEXT ERROR		
	LoRa SFT	-	-	OOM		
	SFT	-	-	OOM		
		Diff	Avg	0.000	0.000	0.822
	Control	Center	Avg	0.132	0.206	0.808
	Vectors	Diff	PCA	0.315	0.223	0.821
		Center	PCA	0.280	0.207	0.823
PLOS	DSPy	-	-	CONTEXT ERROR		
	LoRa SFT	-	-	0.000	0.000	0.000
	SFT	-	-	OOM		
		Diff	Avg	0.000	0.000	0.832
	Control	Center	Avg	0.085	0.112	0.815
	Vectors	Diff	PCA	0.213	0.189	0.830
		Center	PCA	0.100	0.120	0.831
Eureka	DSPy	-	-	CONTEXT ERROR		
	LoRa SFT	-	-	-0.002	0.009	0.813
	SFT	-	-	0.003	-0.011	0.798
		Diff	Avg	0.000	0.000	0.906
	Control	Center	Avg	0.117	0.147	0.870
	Vectors	Diff	PCA	0.137	0.112	0.891
		Center	PCA	0.121	0.098	0.899
SciNews	DSPy	-	-	CONTEXT ERROR		
	LoRa SFT	-	-	-0.034	-0.040	0.483
	SFT	-	-	OOM		
		Diff	Avg	0.000	0.000	0.824
	Control	Center	Avg	0.141	0.177	0.809
	Vectors	Diff	PCA	0.116	0.110	0.822
		Center	PCA	0.120	0.114	0.824

Table 6: Full set of results for all the Control Vector settings we tested for google/gemma-7b-it. We report the Pearson and Kendall-Tau correlation of the specified level of readability with the Coleman-Liau readability index. We experiment with In-Context Learning (ICL), Supervised Finetuning (SFT), QLoRa/LoRa SFT, and DSPy Prompt Optimization (DSPy) as our baselines. For the Control Vectors, we experiment with 2 Contrastive Combination methods (Diff and Center) and 2 Reduction methods (Avg. and PCA). OOM means the baseline could not be run due to an out of memory error on our setup. UNSUPPORTED means that out of the box the baseline does not support google/gemma-7b-it.

	Method	Contrastive Combination	Reduction	Pearson	Kendall Tau	BertScore F1
eLife	DSPy	-	-		CONTEXT ERROR	
	LoRa SFT	-	-	0.001	0.004	0.850
	SFT	-	-	0.001	0.000	0.724
		Diff	Avg	0.000	0.000	0.822
	Control	Center	Avg	0.132	0.206	0.808
	Vectors	Diff	PCA	0.315	0.223	0.821
		Center	PCA	0.280	0.207	0.823
PLOS	DSPy	-	-		CONTEXT ERROR	
	LoRa SFT	-	-	0.000	0.001	0.857
	SFT	-	-	0.0027	0.005	0.424
		Diff	Avg	0.000	0.000	0.832
	Control	Center	Avg	0.085	0.112	0.815
	Vectors	Diff	PCA	0.213	0.189	0.830
		Center	PCA	0.100	0.120	0.831
Eureka	DSPy	-	-		CONTEXT ERROR	
	LoRa SFT	-	-	0.027	0.035	0.766
	SFT	-	-	0.000	0.000	0.696
		Diff	Avg	0.000	0.000	0.906
	Control	Center	Avg	0.117	0.147	0.870
	Vectors	Diff	PCA	0.137	0.112	0.891
		Center	PCA	0.121	0.098	0.899
SciNews	DSPy	-	-		CONTEXT ERROR	
	LoRa SFT	-	-	0.000	0.000	0.104
	SFT	-	-	0.000	0.000	0.361
		Diff	Avg	0.000	0.000	0.824
	Control	Center	Avg	0.141	0.177	0.809
	Vectors	Diff	PCA	0.116	0.110	0.822
		Center	PCA	0.120	0.114	0.824

Table 7: Full set of results for all the Control Vector settings we tested for mistralai/Mistral-7B-Instruct-v0.3. We report the Pearson and Kendall-Tau correlation of the specified level of readability with the Coleman-Liau readability index. We experiment with In-Context Learning (ICL), Supervised Finetuning (SFT), QLoRa/LoRa SFT, and DSPy Prompt Optimization (DSPy) as our baselines. For the Control Vectors, we experiment with 2 Contrastive Combination methods (Diff and Center) and 2 Reduction methods (Avg. and PCA). OOM means the baseline could not be run due to an out of memory error on our setup. LENGTH ERROR means the baseline could not be run due to the context window being too short.

Settings		Per Sample Latency (ms)				Per Token Latency (ms)			
BSZ	Seq Len	Std	CV	Δ	% OH	Std	CV	Δ	% OH
1	512	46.123	48.434	2.311	5.01	0.090	0.095	0.005	5.01
1	1024	85.334	89.729	4.395	5.15	0.083	0.088	0.004	5.15
1	2048	163.423	172.033	8.610	5.27	0.080	0.084	0.004	5.27
4	512	39.693	41.864	2.171	5.47	0.078	0.082	0.004	5.47
4	1024	78.111	82.373	4.262	5.46	0.076	0.080	0.004	5.46
4	2048	156.692	165.435	8.743	5.58	0.077	0.081	0.004	5.58
16	512	38.098	40.223	2.125	5.58	0.074	0.079	0.004	5.58
16	1024	75.600	79.912	4.312	5.70	0.074	0.078	0.004	5.70
16	2048	155.383	163.905	8.522	5.48	0.076	0.080	0.004	5.48

Table 8: Latency comparison between inference on a Standard (Std) LLM and an LLM with added Control Vectors (CV). We report the per sample and per token latency in milliseconds (ms), the Δ latency from using control vectors, and the % overhead (OH).

Technical

Whether complement dysregulation directly contributes to the pathogenesis of peripheral nervous system diseases, including sensory neuropathies, is unclear. We addressed this important question in a mouse model of ocular HSV-1 infection, where sensory nerve damage is a common clinical problem. Through genetic and pharmacologic targeting, we uncovered a central role for C3 in sensory nerve damage at the morphological and functional levels. Interestingly, CD4 T cells were central in facilitating this complement-mediated damage. This same C3/CD4 T cell axis triggered corneal sensory nerve damage in a mouse model of ocular graft-versus-host disease (GVHD). However, this was not the case in a T-dependent allergic eye disease (AED) model, suggesting that this inflammatory neuroimmune pathology is specific to certain disease etiologies. Collectively, these findings uncover a central role for complement in CD4 T cell-dependent corneal nerve damage in multiple disease settings and indicate the possibility for complement-targeted therapeutics to mitigate sensory neuropathies.

Lay

Most people have likely experienced the discomfort of an eyelash falling onto the surface of their eye. Or that gritty sensation when dust blows into the eye and irritates the surface. These sensations are warnings from sensory nerves in the cornea, the transparent tissue that covers the iris and pupil. Corneal nerves help regulate blinking, and control production of the tear fluid that protects and lubricates the eye. But if the cornea suffers damage or infection, it can become inflamed. Long-lasting inflammation can damage the corneal nerves, leading to pain and vision loss. If scientists can identify how this happens, they may ultimately be able to prevent it. To this end, Royer et al. have used mice to study three causes of hard-to-treat corneal inflammation. The first is infection with herpes simplex virus (HSV-1), which also causes cold sores. The second is eye allergy, where the immune system overreacts to substances like pollen or pet dander. And the third is graft-versus-host disease (GVHD), an immune disorder that can affect people who receive a bone marrow transplant. Royer et al. showed that HSV-1 infection and GVHD – but not allergies – made the mouse cornea less sensitive to touch. Consistent with this, microscopy revealed damage to corneal nerves in the mice with HSV-1 infection and those with GVHD. [...]

Table 9: Example technical and lay summaries from eLife.

Technical

The 3-O-sulfotransferase (3-OST) family catalyzes rare modifications of glycosaminoglycan chains on heparan sulfate proteoglycans, yet their biological functions are largely unknown. Knockdown of 3-OST-7 in zebrafish uncouples cardiac ventricular contraction from normal calcium cycling and electrophysiology by reducing tropomyosin4 (tpm4) expression. Normal 3-OST-7 activity prevents the expansion of BMP signaling into ventricular myocytes, and ectopic activation of BMP mimics the ventricular noncontraction phenotype seen in 3-OST-7 depleted embryos. In 3-OST-7 morphants, ventricular contraction can be rescued by overexpression of tropomyosin tpm4 but not by troponin tnt2, indicating that tpm4 serves as a lynchpin for ventricular sarcomere organization downstream of 3-OST-7. Contraction can be rescued by expression of 3-OST-7 in endocardium, or by genetic loss of bmp4. Strikingly, BMP misregulation seen in 3-OST-7 morphants also occurs in multiple cardiac noncontraction models, including potassium voltage-gated channel gene *kcnh2* affected in Romano-Ward syndrome and long-QT syndrome. [...]

Lay

A highly complex environment at the cell surface and in the space between cells is thought to modulate cell behavior. Heparan sulfate proteoglycans are cell surface and extracellular matrix molecules that are covalently linked to long chains of repeating sugar units called glycosaminoglycan chains. These chains can be subjected to rare modifications and they are believed to influence specific cell signaling events in a lineage-specific fashion in what is called the “glycocode.” Here we explore the functions of one member of a family of enzymes, 3-O-sulfotransferases (3-OSTs), that catalyzes a rare modification (3-O-sulfation) of glycosaminoglycans in zebrafish. We show that knockdown of 3-OST-7 results in a very specific phenotype, including loss of cardiac ventricle contraction. Knockdown of other 3-OST family members did not result in the same phenotype, suggesting that distinct 3-OST family members have distinct functions in vertebrates and lending in vivo evidence for the glycocode hypothesis. [...]

Table 10: Example technical and lay summaries from PLOS.

Technical

Extracellular vesicles (EVs) are small vesicles released by cells to aid cell–cell communication and tissue homeostasis. Human islet amyloid polypeptide (IAPP) is the major component of amyloid deposits found in pancreatic islets of patients with type 2 diabetes (T2D). IAPP is secreted in conjunction with insulin from pancreatic cells to regulate glucose metabolism. Here, using a combination of analytical and biophysical methods in vitro, we tested whether EVs isolated from pancreatic islets of healthy patients and patients with T2D modulate IAPP amyloid formation. We discovered that pancreatic EVs from healthy patients reduce IAPP amyloid formation by peptide scavenging, but T2D pancreatic and human serum EVs have no effect. In accordance with these differential effects, the insulin:C-peptide ratio and lipid composition differ between EVs from healthy pancreas and EVs from T2D pancreas and serum. It appears that healthy pancreatic EVs limit IAPP amyloid formation via direct binding as a tissue-specific control mechanism.

Lay

A step closer to a cure for adult-onset diabetes. Professor and head of the Chemical Biology division at the Department of Biology and Biological Engineering, she leads a research team focusing on metalloproteins and proteins that fold incorrectly. Exosomes in patients with the disease don't have the same ability. This discovery, by a research collaboration between Chalmers University of Technology and AstraZeneca, takes us a step closer to a cure for type 2 diabetes. Proteins are the body's workhorses, carrying out all the tasks in our cells. A protein is a long chain of amino acids that must be folded into a specific three-dimensional structure to function properly. Sometimes, however, they misbehave and aggregate—clump together—into long fibers called amyloids, which can cause diseases. It was previously known that type 2 diabetes is caused by a protein aggregating in the pancreas. "What we've found is that exosomes secreted by the cells in the pancreas stop that process in healthy people and protect them from type 2 diabetes, while the exosomes of diabetes patients do not," says Professor Pernilla Wittung-Stafshede, who headed the study. [...]

Table 11: Example technical and lay summaries from Eureka.

Technical

Observational studies suggest that lower educational attainment (EA) may be associated with risky alcohol use behaviors; however, these findings may be biased by confounding and reverse causality. We performed two-sample Mendelian randomization (MR) using summary statistics from recent genome-wide association studies (GWAS) with <780,000 participants to assess the causal effects of EA on alcohol use behaviors and alcohol dependence (AD). Fifty-three independent genome-wide significant SNPs previously associated with EA were tested for association with alcohol use behaviors. We show that while genetic instruments associated with increased EA are not associated with total amount of weekly drinks, they are associated with reduced frequency of binge drinking ≥ 6 drinks ($\beta_{IVW} = -0.198$, 95% CI, -0.297 to -0.099 , $P_{IVW} = 9.14 \times 10^{-5}$) [...]

Lay

Higher educational attainment—spending more years in education—may impact people's drinking behaviour and reduce their risk of alcohol dependence, according to a study published in *Molecular Psychiatry*. Alcohol consumption is a major risk factor for death and disability worldwide. Identifying factors associated with how much, how often and what people drink may be important for developing and improving intervention and treatment strategies. Previous studies have suggested that educational attainment may influence drinking, but with conflicting results. To assess the possible effects of educational attainment on alcohol use behaviours and alcohol dependence, a team of researchers at the National Institutes of Health, USA used two-sample Mendelian randomisation statistical methods. [...] Dr. Lohoff said: "It is important to understand that while these genetic variants allow us to investigate the possible effect of educational attainment on alcohol consumption and alcohol dependence, this doesn't mean that educational attainment can't be modified. The possible effect of educational attainment on drinking that we show in this study suggests that increasing educational attainment may be a useful target for prevention programs against problematic alcohol use, alcohol dependence, and their consequences." [...]

Table 12: Example technical and lay summaries from SciNews.

RL	SFT Output
1	. These have to brain with for to. The, the brain, and the. This and to. These and, and the cells. The to a and to the heart that cells to is cells, are the their their. In, it the heart of the brain this. The. Next of are cells the brain the heart the and by. But the that the cells to to this the et al to the cells. These, and to the environment, a. We the mechanism. The, and in,. In the body and are is. This cells, the body, and of the heart to the body to of a of, the brain. The, the.
3	brain a a of to the to, they. However.
5	, the that the cells to the of the the of of the visual in and of the brain are, the the body. The. In, but the time to the eyes the brain and the brain cells. The their are to the brain. Here, and the cells.
7	enerative their body to, the two the, in the body of, the their into. We from the cell of, the into the by their cells.
9	brain. In.

(a) Outputs generated by SFT model.

RL	LoRA Output
1	Small open reading frames (smORFs) are small stretches of DNA that code for proteins, but their translation into proteins has been difficult to detect. This is because they are too short to produce the high conservation scores that are used to predict which stretches of DNA code for proteins. To overcome this problem, Kondziella et al. have developed a new method called Poly-Ribo-Seq that uses a combination of biochemistry and sequencing to detect smORFs that are being translated into proteins. Using this method, they were able to detect smORFs in <i>Drosophila</i> cells that were not detected by other methods. These smORFs were found to be translated at similar levels to canonical proteins and had the same characteristics as canonical proteins. These findings suggest that many smORFs in <i>Drosophila</i> are being translated into proteins and may have important functions in the cell. Future experiments will be needed to confirm this and to determine the functions of these proteins.
3	The discoidin domain receptors (DDRs) are a family of receptor tyrosine kinases (RTKs) that bind to collagen and play a key role in tissue development, cell migration, and disease progression. DDRs are composed of an extracellular region that binds to collagen, a transmembrane domain and a cytoplasmic region that contains the catalytic kinase domain. Unlike other RTKs, DDRs are constitutively dimerised in the absence of ligand and activate downstream signalling through a mechanism that is not well understood. Here, we show that DDR1 activation by collagen involves phosphorylation of the juxtamembrane and activation loop tyrosines in trans between dimers. Phosphorylation between DDR1 dimers requires specific contacts between the transmembrane domains but not between the extracellular domains. These findings define a unique activation mechanism for DDR1.
5	Stem cell biology is a rapidly evolving field that is critical for our understanding of how tissues and organs form during development and how they are maintained in the adult. In the fruit fly <i>Drosophila melanogaster</i> , the lymph gland is a multi-lobed organ that produces blood cells throughout the larval stages. This organ contains a stem cell niche that maintains a pool of undifferentiated blood cell progenitors. While the adult hematopoietic hub is known to be the final site of hematopoiesis, the mechanism of blood cell formation in larvae has been largely unexplored. Here we identify a population of cells in the early larval lymph gland that express the Notch signaling pathway. These cells are multipotent, dividing and give rise to the domeless expressing blood cell progenitors that populate the lymph gland. Notch signaling is required for their maintenance and relies on Dpp signaling from the niche. These results reveal the presence of hematopoietic stem cells in the early larval lymph gland and establish <i>Drosophila</i> as a model for studying hematopoietic stem cell biology. The findings also shed light on the evolutionary conservation of hematopoietic stem cell biology between <i>Drosophila</i> and vertebrates.
7	The malaria parasite <i>Plasmodium</i> causes a disease that kills more than a million people every year. To infect the human body, the parasite must first invade the liver cells. How this invasion takes place is still not well understood. Silvie et al. have now investigated how <i>Plasmodium</i> parasites enter human liver cells. The experiments show that two host proteins, CD81 and SR-BI, can be used by the parasite to invade the liver cells. The parasite uses different proteins to enter the liver cells of different species. The experiments also show that a parasite protein called P36 can bind to the host proteins and trigger the invasion of the liver cells. The experiments provide new insights into the invasion process and could ultimately help to develop new treatments for malaria.
9	A large family of proteins called TRIM5 is thought to be a key part of the body's natural defense against viruses. These proteins prevent certain viruses from multiplying by binding to the proteins that form the outer shell of the virus. To find out how TRIM5 proteins work, researchers have made a detailed 3D image of a TRIM5 protein bound to a virus. The image shows that the protein forms a hexagonal pattern around the virus, with each hexagon made up of 12 smaller protein molecules. The next step will be to find out how the proteins interact with the virus to prevent it from multiplying. These findings could also help to develop new treatments for viruses.

(b) Outputs generated by LoRA model.

RL	DSPy Output
1	""
3	""
5	""
7	""
9	""

(c) Outputs generated by DSPy (empty because of length failure).

Table 13: Model outputs at each requested Readability Level (RL) for SFT, LoRA, and DSPy. Readability level is on a scale of 1 to 9, with 9 being the most readable.

RL	ICL Output
0	The study aimed to investigate the early development of the zebra finch (<i>Taeniopygia guttata</i>) and its potential use as a model organism for studying embryogenesis and stem cell biology. The researchers characterized the morphology and molecular features of finch embryos at oviposition, which were found to be equivalent to the blastocyst stage mouse embryo. They also identified the expression of pluripotency-associated genes, including <i>Nanog</i> , <i>PouV</i> , and <i>Dnmt3b</i> , in the finch blastoderm at oviposition. The study used quantitative PCR (Q-PCR) to compare the expression levels of various genes associated with early embryonic development and pluripotency in finch and chick blastoderms at oviposition. The results showed that finch blastoderms expressed higher levels of naive pluripotency markers, such as <i>Fbxo15</i> , <i>Prdm14</i> , and <i>Tbx3</i> , compared to chick blastoderms. [...] The findings have implications for our understanding of developmental biology and may contribute to the development of new technologies for understanding and manipulating avian development.
25	The study aimed to investigate the early development of the zebra finch (<i>Taeniopygia guttata</i>) and its potential use as a model organism for studying embryogenesis and stem cell biology. The researchers characterized the morphology and molecular features of finch embryos at oviposition, which were found to be equivalent to the blastocyst stage mouse embryo. They also identified the expression of pluripotency-associated genes, including <i>Nanog</i> , <i>PouV</i> , and <i>Dnmt3b</i> , in the finch blastoderm at oviposition. The study used quantitative PCR (Q-PCR) to compare the expression levels of various genes associated with early embryonic development and pluripotency in finch and chick embryos. The results showed that finch blastoderms at oviposition expressed higher levels of naive pluripotency markers, such as <i>Fbxo15</i> , <i>Prdm14</i> , and <i>Tbx3</i> , compared to chick blastoderms. [...] Overall, the study provides new insights into the early development of the zebra finch and its potential use as a model organism for studying embryogenesis and stem cell biology. The findings suggest that the finch blastoderm at oviposition is a valuable resource for studying the naive stage of embryogenesis and the derivation of ES cells.
50	The study aimed to investigate the early development of the zebra finch embryo and its potential use for comparative stem cell studies and genome engineering. The researchers used a combination of morphological and molecular characterization to study the finch embryo at oviposition, which is equivalent to the blastocyst stage in mouse embryos. They found that the finch embryo at oviposition is similar to the mouse embryo at E4-E4.5, a stage that has been termed the naive state of embryogenesis. The researchers used quantitative PCR (Q-PCR) to compare the expression levels of various genes associated with early embryonic development and pluripotency in the finch and chick blastoderms at oviposition. They found that the finch blastoderm at oviposition showed higher levels of expression of naive pluripotency markers, such as <i>Fbxo15</i> , <i>Prdm14</i> , and <i>Tbx3</i> , compared to the chick blastoderm. They also found that the finch blastoderm at oviposition showed higher levels of expression of genes associated with the naive state of pluripotency, such as <i>Nrobl</i> and <i>Esrrb</i> . [...] Overall, the study provides new insights into the early development of the zebra finch embryo and its potential use for comparative stem cell studies and genome engineering. The findings suggest that the finch blastoderm at oviposition is a valuable model system for studying the naive state of embryogenesis and the mechanisms of ES cell derivation.
25	The study aimed to investigate the early development of the zebra finch embryo and its potential use for comparative stem cell studies and genome engineering. The researchers used a combination of morphological and molecular characterization to study the finch embryo at oviposition, which is equivalent to the blastocyst stage mouse embryo. They found that the finch blastoderm at oviposition was similar to the mouse embryo at the same stage, with a multi-layered structure and the presence of epiblast and hypoblast precursors. The researchers also found that the finch blastoderm at oviposition expressed markers of pluripotency, including <i>Nanog</i> , <i>PouV</i> , and <i>Dnmt3b</i> , which are associated with the naive state of pluripotency. They also found that the finch blastoderm at oviposition showed higher levels of expression of naive pluripotency markers, such as <i>Fbxo15</i> , <i>Prdm14</i> , and <i>Tbx3</i> , compared to the chick blastoderm at oviposition. [...] Overall, the study provides new insights into the early development of the zebra finch embryo and its potential use for comparative stem cell studies and genome engineering. The study also highlights the importance of the finch embryo as a model system for studying the mechanisms of pluripotency and stem cell maintenance. [...]
100	The study aimed to investigate the early development of the zebra finch (<i>Taeniopygia guttata</i>) and its potential use as a model organism for studying embryogenesis and stem cell biology. The researchers used a combination of morphological and molecular characterization techniques to study the finch embryo at oviposition, including quantitative RT-PCR, immunohistochemical staining, and in situ hybridization. The results showed that finch embryos are laid at a stage equivalent to the blastocyst stage in mouse embryos, with a multi-layered blastoderm that has not yet epithelialized. The epiblast and hypoblast precursors were already molecularly specified at this stage, and the primordial germ cell fate was marked by immunoreactivity to SSEA-1 and EMA-1 antibodies. [...] They also suggested that the finch embryo may be a useful model organism for studying embryogenesis and stem cell biology, particularly in the context of avian development. Overall, the study provides new insights into the early development of the zebra finch and its potential use as a model organism for studying embryogenesis and stem cell biology. The findings have implications for our understanding of the evolution of germ layer formation in amniotes and the development of new technologies for manipulating avian development.

Table 14: Example Outputs from In-Context Learning at each requested Readability Level (RL). Readability level is on a scale of 0 to 100, with 100 being the most readable.

RL	CV Output
-1.0	The wild fish samples were collected in 2015 in the Gonarezhou National Population in Zimbabwe and Mozambique. Intestines were collected at each location and preserved in pure ethanol. Sampling locations coordinates are listed in Figure 2—source data 1. Fish (GRZ strain) used for microbiota analysis and scored for survival were individually housed from week 4 post-hatching in single 2.8L tanks connected to a water recirculation system receiving 12 hr of light and 12 hr of dark every day. Water temperature was set to 28°C and fish were fed blood worm larvae and brine shrimp nauplii twice a day during the week and once a life span in the wild fish populations, we identified a core microbiota that is conserved between wild and captive TK, with a significant reduction in gut bacterial richness in old fish.
-0.5	The turquoise killifish (TK) is a naturally short-lived vertebrate model system, characterized by a broad spectrum of aging phenotypes, including cancer, neurodegeneration, and behavioral decline. We show that TK have a complex gut microbial taxonomic diversity similar to other vertebrate aging model organisms, including zebrafish, mice, and humans. The core microbiota is conserved between wild and captive TK, with a significant reduction in gut bacterial richness during aging. Acute gut microbiota transfer in the context of normal aging significantly prolongs life span in a vertebrate, becoming a novel candidate life span enhancing intervention. Our results indicate that improving the ecological diversity of the GM in old individuals helps to restore health and prolongs life span. Our approach could provide a key to slowing aging and retarding the onset of age-associated diseases by specifically targeting the GM. The wild fish samples were collected in 2015 during an expedition in the Gonarezhou National Park in Zimbabwe and Mozambique. Intestines were collected at each location and preserved in pure ethanol. Sampling locations coordinates are listed in Figure 2—source data 1.
0.0	The study investigates the relationship between gut microbiota and aging in the turquoise killifish, a short-lived vertebrate model organism. The researchers found that the gut microbiota of young fish is more diverse and complex than that of old fish, with a higher abundance of beneficial bacteria such as Firmicutes and Bacteroidetes. In contrast, old fish have a less diverse gut microbiota with a higher abundance of pathogenic bacteria such as Proteobacteria. The researchers also found that the gut microbiota of old fish is associated with inflammation and immune responses, while the gut microbiota of young fish is associated with healthy gut function and cell proliferation. The study suggests that the gut microbiota plays a key role in aging and that manipulating the gut microbiota may be a potential strategy for promoting healthy aging. The researchers also found that transferring young gut microbiota to middle-aged fish can prolong their lifespan and improve their health, suggesting that the gut microbiota may be a key factor in determining lifespan. Overall, the study provides new insights into the relationship between gut microbiota and aging and highlights the potential of manipulating the gut microbiota as a strategy for promoting healthy aging. The study used a combination of 16S rRNA gene amplicon sequencing and RNA sequencing to analyze the gut microbiota and transcriptome of young and old turquoise killifish. The researchers found that the gut microbiota of young fish is more diverse and complex than that of old fish, with a higher abundance of beneficial bacteria such as Firmicutes and Bacteroidetes. In contrast, old fish have a less diverse gut microbiota with a higher abundance of pathogenic bacteria such as Proteobacteria.
0.5	The study looked at the relationship between the gut microbiome and aging in the turquoise killifish. The researchers found that as the fish get older, their gut microbiome changes and becomes less diverse. They also found that if they took the gut microbiome from a younger fish and put it into an older fish, it could help to extend the older fish's lifespan. The researchers also looked at the genes that are expressed in the gut of older fish and found that they are different from the genes that are expressed in the gut of younger fish. They also found that if they took the gut microbiome from a younger fish and put it into an older fish, it could help to change the way that the older fish's genes are expressed. The study suggests that the gut microbiome could be an important factor in how long we live, and that it could be possible to use it to help to extend our lifespan. The researchers also found that the gut microbiome of the turquoise killifish is similar to the gut microbiome of humans, and that it could be possible to use the turquoise killifish as a model to study how the gut microbiome affects our health and how it could be used to help to extend our lifespan. The study was conducted by a team of researchers from the University of Bologna and the University of California, San Francisco. The researchers used a technique called 16S rRNA gene sequencing to study the gut microbiome of the turquoise killifish. They also used a technique called RNA sequencing to study the genes that are expressed in the gut of the fish. The researchers also looked at the lifespan of the fish and how it was affected by the gut microbiome.
1.0	The researchers are looking at how to live longer by looking at what is in the gut. They are looking at the gut of a fish that lives for about 10 years. They want to see if they can help the fish live longer by changing what is in the gut. They are also looking at what is in the gut of people. They want to see if they can help people live longer by changing what is in their gut. They are also looking at what is in the gut of mice. They want to see if they can help mice live longer by changing what is in their gut.

Table 15: Example outputs from Control Vector method at each requested Readability Level (RL). Readability level is on a scale of -1.0 to 1.0 , with 1.0 being the most readable.

Building a Custom Taxonomy of AI Skills and Tasks from the Ground Up with Job Postings

Stephen Meisenbacher

Technical University of Munich
Garching, Germany
stephen.meisenbacher@tum.de

Peter Norlander

Loyola University Chicago
Chicago, IL, USA
pnorlander@luc.edu

Abstract

Utilizing LLMs for automated taxonomy construction presents a clear opportunity for the comprehensive, yet efficient mapping of potentially complex domains. When contending with high volumes of rapidly growing corpora, however, it becomes unclear how to best leverage such data for optimal taxonomy construction. Taking the case of systematizing *AI skills in the workplace*, we use two large-scale job postings corpora to investigate key design decisions for the inclusion (or exclusion) of data points for taxonomy construction. We propose TAXONOMYBUILDER as a blueprint for our systematic study, with which we evaluate various configurations of custom, data-informed, and hierarchical taxonomies. We demonstrate that *less* data can provide more clarity: filtering inputs to TAXONOMYBUILDER provides better domain-specific coverage than offering unfiltered inputs to clustering and LLM-enhanced hierarchical taxonomy labeling tools.

1 Introduction

What AI Skills are in demand in recent years? To answer this and other timely questions about rapid change in labor markets, novel, open taxonomies of tasks and skills are needed to enable tracking of granular changes in work (Frank et al., 2019; Margaryan, 2023). When built from high-volume, real-time sources of labor market text such as job postings, such taxonomies are valuable sources of data to inform individual career development, educational program designs, workforce development interventions, and firm strategy (Zweig, 2026).

However, proprietary taxonomies, data access restrictions, and lack of standardization limit the development and use of automated tools to harness general information from job postings at scale (National Academies of Sciences, 2025). Open data, including novel taxonomies and data generated by them, are potential public goods to support

broad skills development, training needs for workforce development, growth, democratization, and entrepreneurship (Lerner and Tirole, 2002; Nagaraj et al., 2020; Nagaraj, 2022). From a research standpoint, to assess whether AI is complementing or replacing human skills (Acemoglu et al., 2026), social scientists need detailed and up-to-date taxonomic information on the human tasks and skills of workers building and using AI.

When faced with such problems, researchers have often turned to *taxonomy construction*. The creation of taxonomies has spanned the decades, and established methodologies are both accessible and widely used (Michalski and Stepp, 1983; Nickerson et al., 2013). The shortcomings of taxonomy creation mirror those of many traditionally manual tasks, such as the expense of large-scale manual labor, researcher fatigue and bias, and the difficulty in empirically validating created artifacts with real-world data (Vu et al., 2025). The creation of representative taxonomies in custom, complex, and domain-specific fields remains challenging.

In the field of Natural Language Processing and especially in the era of LLMs, researchers have begun to explore the capabilities of automating the process of taxonomy construction. The promise of such approaches is the ability to sift through and categorize massive amounts of data, a step that would be highly expensive in the traditional manner of hand-crafted taxonomies (Vu et al., 2025). Improving initial approaches leveraging clustering algorithms (Gordon, 1996; Cimiano et al., 2004; Ienco and Meo, 2008), recent state-of-the-art methods have leveraged LLMs and their generative capabilities (Moskvoretskii et al., 2026).

As a motivation for our work, when confronted with the need to develop a taxonomy of AI skills from job postings, the literature offers prompt engineering guidance (Chen et al., 2023; Vu et al., 2025) and handles situations where *seed* information, or *seed taxonomies* can guide the LLM-assisted tax-

onomy building process (Zhang et al., 2025; Motamedi et al., 2026). However, in the situation at present, questions posed to us include: *which data* should be used (among all data from a massive text corpus), *how much* of this data should be used, and *in what way* can the built taxonomies be evaluated in turn on real-world datasets? Thus, while it has been established that leveraging LLMs can be helpful for taxonomy building, there is a need for systematic experimentation with key design parameters for best practices in custom domains.

To guide our experiments, we leverage two large corpora of job postings in the United States, spanning the past decade. To address the needs identified above, we build taxonomies that efficiently and comprehensively organize AI in the workplace, particularly skills, tasks, experience, and qualifications. To accomplish this, we design a simple, yet powerful pipeline that transforms mined candidate contexts into hierarchical taxonomies. We evaluate 12 configurations of our pipeline on automatic and LLM-based metrics, leading to clear recommendations for automatic taxonomy building.

Our findings reveal that when handling massive data corpora for automated taxonomy construction, *less is more*. Although the tendency to include as much data as possible, including via data augmentation, might plausibly increase domain coverage, our results demonstrate the opposite. As such, we learn that careful augmentation, strict clustering, and selective filtering can contribute to comprehensive, yet concise taxonomies for large, data-rich domains such as AI skills in the workplace. Our work makes the following additional contributions:

1. We deepen the capabilities of LLM-based taxonomy building by asking and answering key questions surrounding the setup and execution of automatic taxonomy creation pipelines.
2. We showcase the ability of our proposed pipeline, TAXONOMYBUILDER, to efficiently and comprehensively map complex domains such as *AI Skills in the Workplace*, replicated on two large-scale job posting corpora.
3. We make the resulting AI skills taxonomies publicly available and open for further use.
4. We open-source TAXONOMYBUILDER as a generalizable tool to any domain of interest, available at <https://github.com/sjmeis/TaxonomyBuilder> and as a Python package (TAXONOMYBUILDER).

2 Related Work

Automated taxonomy construction. Automated taxonomy construction, often discussed alongside *ontology construction*, *ontology engineering*, or *taxonomy induction*, has been approached in diverse ways even before the era of LLMs. Early methods focus on lexico-syntactic patterns to define relationships in a hierarchy (Hearst, 1992), which are efficient yet suffer from low recall. Subsequent methods relied on the distributional hypothesis (Padó and Lapata, 2007), using the backbone of clustering algorithms and early text representations (e.g., bag of words). Hierarchical approaches (Gordon, 1996; Fountain and Lapata, 2012), allow for the vertical scaling of taxonomies, extending beyond similarity-based clustering of related terms. Other approaches supplement these methods with rich existing knowledge bases such as WordNet (Pasca, 2004; Kozareva and Hovy, 2010).

Automated taxonomy construction has been aided by probabilistic and model-based approaches (Snow et al., 2006; Poon and Domingos, 2010) or graph-based methods (Velardi et al., 2013). Building on the success of embeddings as text representations, ensuing works leveraged these embeddings to form hybrid taxonomy construction methods (Fu et al., 2014; Espinosa-Anke et al., 2016; Zhang et al., 2018). Other works utilized modern reinforcement learning (Mao et al., 2018) or transfer learning (Navarro-Almanza et al., 2020).

LLM-assisted taxonomy construction. The emergence of LLMs has sparked research interest in improving automated taxonomy construction (Vu et al., 2025). Many approaches have been proposed in recent years, including methods to augment existing taxonomies (Kargupta et al., 2025) as well as those that specialize in specific domains, such as scholarly research (Zhu et al., 2025; Lahiri et al., 2025). Typical LLM-based workflows include an extraction and clustering phase (Huang and He, 2025), followed by generative labeling via prompting or fine-tuning (Chen et al., 2023), and finally taxonomy construction (Li et al., 2025; Motamedi et al., 2026). While this workflow is bottom-up (starting from leaf nodes), others follow a top-down approach (Zeng et al., 2024; Marchenko and Dvoichenkov, 2024; Wan et al., 2024).

Our work adopts the bottom-up approach, as we investigate the case of modeling AI skills in job postings, for which no defined taxonomies exist. We build upon the work of approaches

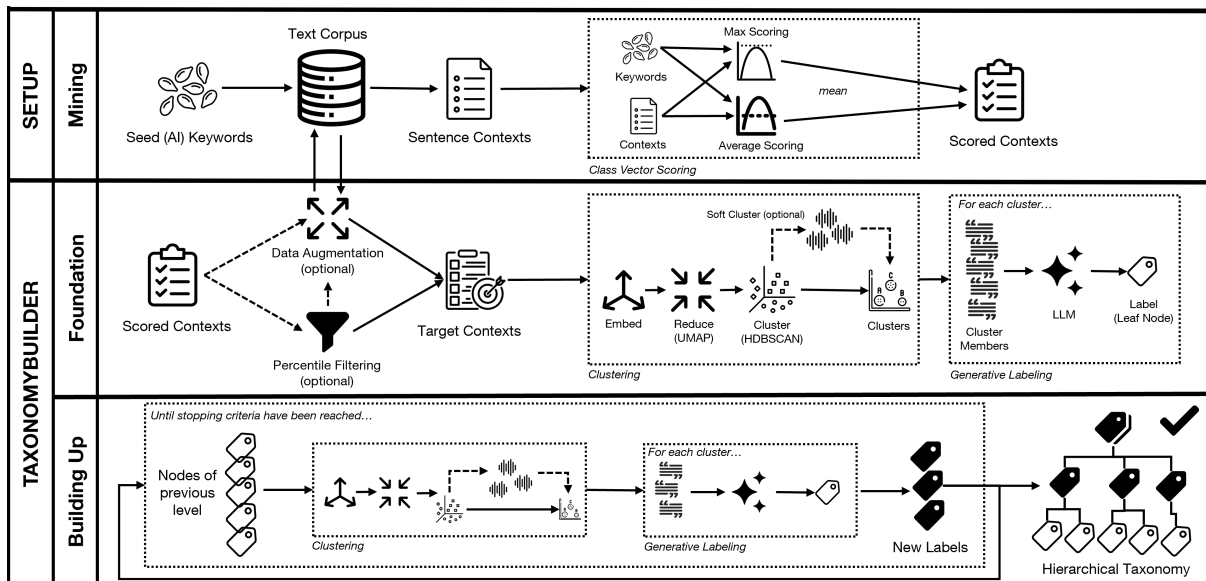


Figure 1: The TAXONOMYBUILDER method. In the top lane, we detail the setup method we follow as a precursor to taxonomy construction, which consists of keyword-based context mining and class-based scoring. The TAXONOMYBUILDER method, in turn, consists of two primary stages (depicted in the center and bottom lanes): (1) the construction of the foundation (leaf) level, followed by iterative vertical construction of further levels of the taxonomy hierarchy. We refer the readers to the main body for specific details on all stages of the process.

such as LLMTAXO (Zhang et al., 2025), who lay the groundwork for cluster labeling methods with LLMs (as well as LLM-based evaluation), and TAXOADAPT (Kargupta et al., 2025), which introduce important inter-mechanism redundancy checks. In terms of domain, the closest work to date is that of Li et al. (2025), who also operate in the domain of job posting data. However, we note that Li et al. (2025) explore LLM-assisted taxonomy construction on very limited data samples (with a maximum of just over 10k job postings), whereas we study a case with millions to tens of millions of unique data points in question. In this, we introduce and systematically evaluate important questions surrounding optimal data usage, which has been previously unexplored, and evaluation, which becomes highly relevant in big data scenarios to ensure proper domain coverage.

3 Datasets

We use two large corpora of job postings from the United States. The first is a collection of over 30 million job postings from the National Labor Exchange (NLx) between 2024 and 2025. To maintain reproducibility, we also evaluate our proposed method on an open corpus of 1.5 million federal job postings collected by Resh et al. (2025). Both datasets are described briefly below.

NLx. The National Labor Exchange (NLx) “provides workforce development professionals, academic researchers, employers, and other organizations that rely on labor market information (LMI) with high-quality, transparent, real-time and historical data that represents the diversity of jobs available in the labor market”¹ As of March 2026, the exchange hosts nearly 3.1 million open positions from 300,000 employers.

The entirety of the NLx corpus currently comprises over 173 million unique job postings from 2007 onwards. To maintain a manageable subset and to focus in on years which feature modern AI skill requirements, we narrow down our dataset to data from 2024 and 2025. Together, this results in a dataset of 32.12 million job postings within the two-year timespan. The average word count of the job postings is 875 (using NLTK on a sample of all December 2025 postings). We make no modifications to the original NLx job posting texts.

Use of NLx data precludes sharing any data in its original form. As such, we replicate our experiments on a second, publicly available dataset.

USAJOBS. We also utilize a corpus of 1.54 million job postings from the USAJOBS platform, the US government’s official employment website, made available by Resh et al. (2025). This corpus

¹<https://nlxresearchhub.org/about-us/>

contains job postings for positions in the federal government from 2017-2023. The USAJOBS postings are generally long, with an average word count of 5054 (as per NLTK on November 2023 postings).

Test set. It should be noted that we reserve the last month’s data in each corpus for evaluation purposes, i.e., December 2025 of NLx and November 2023 of USAJOBS. Thus, all procedures described in the next section, in which data is mined or augmented, leave out this reserved test set.

Data release. We make the USAJOBS dataset publicly available². Regarding outputs and analysis of NLx results, we cannot share the original data, a note we expand upon in our Ethics Statement.

4 The TAXONOMYBUILDER Method

In the following, we detail the design and implementation of TAXONOMYBUILDER, a simple, lightweight, and efficient library for scaling from large corpora of unstructured texts to comprehensive and hierarchical taxonomies of concepts. The entire method is graphically visualized in Figure 1.

4.1 The Setup: Mining contexts of interest

The first stage of taxonomy construction, which we formally placed outside of the confines of TAXONOMYBUILDER, is the curation of candidate *contexts* (i.e., sentences) which will serve as the basis for an AI skills taxonomy. Using our large corpus of job postings, we take a keyword-based approach for an initial efficient mining of candidates, followed by a scoring procedure via *class vectors*.

Candidate mining. Due to very large-scale nature of our text corpora, we opt for an efficient, coarse-grained keyword-based search for the initial extraction of candidate contexts for an AI skills taxonomy. To build a comprehensive list of AI-related keywords, we consult various previous works that curate dictionaries of AI keywords sources (Baruffaldi et al., 2020; Alekseeva et al., 2021; Lou and Wu, 2021; Goldfarb et al., 2023; Lightcast, 2023; Maslej et al., 2025; Tambe, 2026).

We compiled the keywords from the above sources, performed light editing (e.g., splitting entries with ‘/’ into two and removing parentheses), and then de-duplicated the list. This yielded a final set of 590 AI keywords, found in our repository.

²<https://huggingface.co/datasets/loyoladatamining/usajobs> Note: the corpus has been updated to include more postings than were used in this work.

We then performed an efficient keyword search using the PYAHOCORASICK library, keeping all sentences matching one or more of the keywords, with either a white space or sentence boundary (to avoid inner-string matches). We also enumerated the number of sentence matches per job posting. This keyword-based search yielded 5.58 million candidate sentences for NLx and 213k for USAJOBS. As a further filtering step, we kept only the candidate sentences from job postings with three or more matches (so as to remove noise and non-AI jobs). This resulted in a final set of 251k candidates for NLx and 19k candidates for USAJOBS.

Candidate scoring. Recognizing that a keyword-search introduces the potential for false positives (a prime example being *torch* as a welding instrument and not the machine learning library), we leveraged an efficient yet powerful method to score and rank all extracted candidates, allowing for further filtering. Motivated by the idea of *class label vector representations* (Schopf et al., 2020) and mean-max class scoring (Meisenbacher et al., 2024), we devised a hybrid scoring method that compares all candidate sentences to all keywords (here, our AI keywords) via cosine similarity of their sentence embeddings (Reimers and Gurevych, 2019).

For each given candidate sentence, its embedding is compared to all keyword embeddings, and then mean and max similarity scores are averaged to produce a final class-relatedness score. For numerical stability, we calculate this score on the basis of three embedding models: ALL-MINILM-L12-V2 (Reimers and Gurevych, 2019), GTE-LARGE (Li et al., 2023), and EMBEDDINGGEMMA-300M (Vera et al., 2025). The scores of these three models were averaged for the final score.

We note that while the three model approach was used for this work, the released TAXONOMYBUILDER code currently only supports class-based filtering with one embedding model of choice.

4.2 Forming the foundation

With the (large) set of candidate sentences mined, the next step becomes to distill these statements into a foundational layer of a taxonomy, one that is both representative of their diversity expressed among the candidates, but that also significantly compresses this information to form an accessible taxonomy. This process represents the first stage of TAXONOMYBUILDER: *forming the foundation*.



Figure 2: Abridged example of the taxonomy structure produced by TAXONOMYBUILDER for AI skills. The right-most nodes exemplify the initial foundation built from raw context candidates. Our methods then proceed to build upwards (“left”) upon this foundation. Note that *AI Skills Taxonomy* is inserted for completion.

Preliminaries: more and/or less data? How to effectively and intelligently leverage massive text corpora for an AI skills taxonomy using TAXONOMYBUILDER requires investigation and evaluation. Given the core set of candidate statements described above, we explore two divergent paths, asking the question of whether *more* (via augmentation) and/or *less* (via filtering) data is most sensible.

The first option, *data augmentation*, supplements the high-precision, low-recall, keyword-based candidate search with efficient embedding similarity augmentation. Given the set of (embedded) candidates, we perform a semantic search against all the documents in the respective corpus, *augmenting* the candidate set with any new contexts found that exceed a cosine similarity threshold (we choose 0.9). For this procedure, we solely use EMBEDDINGGEMMA-300M. Using this threshold augmented the NLx candidates with an additional 746k candidates and 31k candidates for USAJOBS.

We also explore the option of *less* data, in order to filter out noisy or irrelevant statements that have received a low class score. In this work, we experiment with keeping candidates (by score) above the top 75th, 50th, and 25th percentile. Note that such filtering also cascades to augmented candidates: if an original candidate is filtered out, all of its corresponding augmented sentences are as well.

Clustering. Given the target candidate contexts (following optional augmentation or filtering), the next step in TAXONOMYBUILDER is to cluster these contexts. Following the recommendations of Zhang et al. (2025), we choose HDBSCAN as the clustering algorithm (Campello et al., 2013), using the GPU-accelerated library provided by rapids.ai³. We perform clustering on embeddings that have been dimensionality reduced via UMAP to 10 dimensions (also using the CUMML acceleration). We set the minimum cluster size to 5.

³<https://rapids.ai/cuml-accel/>

Since HDBSCAN by design assigns data points to a “noise cluster” if no fitting cluster is found, we explore the option of re-introducing such points to the clustering results via *soft clustering*. This process involves calculating the distance of a noise point to all cluster centroids, and assigning it the cluster of closest proximity. We leave this option as a third point of investigation, along with data augmentation and percentile filtering.

The results of this process is a collection of clustered candidates, each of which represent a single entry in the bottom (leaf) level of the taxonomy.

Generative cluster labeling. To transform clusters of related statements into concise entries for a taxonomy foundation, we leverage the generative capabilities of LLMs to produce accurate representations (“labels”) of each cluster. This technique has been used in several previous works as an effective distillation method (Chen et al., 2023; Zeng et al., 2024; Zhang et al., 2025). For our main experiments, we use GPT-4O-MINI (OpenAI et al., 2024) with the prompt in Table 3 of the Appendix.

Label verification and pruning. To avoid error or redundancy propagation, we introduce a similarity-based check to the generated labels before proceeding to hierarchical taxonomy building. Specifically, the pairwise cosine similarity between each label embedding and all other labels is calculated, and all sets of labels exceeding a similarity threshold (here, 0.95) are aggregated into a consolidated label (using the zero-shot prompt of Table 4). The resulting collection of labels represent the leaf nodes of the taxonomy to be constructed.

4.3 Building up

The final stage of TAXONOMYBUILDER involves the iterative (vertical) creation of subsequent levels of the taxonomy. Given the foundational level labels, these are then used as the input for the next level, following the same embedding-clustering-

labeling pipeline as before. This process continues until one of two stopping criteria are met: (1) the number of labels generated at a given label are less than a user-defined value (default: 10), or (2) the number of levels generated has reached a user-defined threshold (default: 5). This mitigates the possibility of either scarce top levels, or taxonomies that are too tall and therefore hard to interpret.

An example AI skills taxonomy, excerpted from an output of our experiments, is depicted in Figure 2. The full taxonomies resulting from all experimental configurations are located in our repository.

5 Experimental Setup

The primary objective of our experiments is to evaluate the downstream effects of key design choices in the inclusion of data for taxonomy construction using TAXONOMYBUILDER. In particular, we study the impact of data augmentation, filtering measures, and soft clustering on the internal and external validity of the constructed AI skills taxonomies. Thus, we employ an intrinsic clustering-specific metric for clustering performance, as well as LLM-as-a-Judge criteria to evaluate the produced taxonomies as a whole. Finally, we empirically measure external validity by testing how well the constructed taxonomies generalize to real-world data, i.e., our held-out job posting test sets.

5.1 Taxonomy configurations

Using our two datasets, we construct an array of taxonomies based on different configurations, enabling a comparative study of key design decisions. In particular, we vary the following configurations:

- **Data Augmentation:** whether the original candidate sets are augmented with new examples from the corpus. We utilize EMBEDDINGGEMMA-300M with a similarity threshold of 0.9.
- **Percentile Filtering:** leveraging class vector scoring, at which score threshold are candidates filtered out. We choose three configurations: top 25th, 50th, and 75th percentile of scores, i.e., only keeping candidates within these percentiles.
- **Soft Clustering:** whether to include noise points back in clusters or not (via centroid proximity).

With the above configurations, we create and evaluate 12 taxonomies for each dataset (24 in total), leveraging the *TaxonomyBuilder* method.

5.2 Evaluation Metrics

We select three classes of evaluation metrics, spanning *clustering evaluation*, *LLM-based taxonomy assessment*, and *taxonomy domain coverage*.

Clustering evaluation. As the variation of our configurations, described above, can reasonably have a direct influence on the outcome of the HDBSCAN clustering, we employ a well-known cluster evaluation metric, namely the silhouette score. A higher score in this metric implies that clusters are more well-defined. We use the SKLEARN implementation, and we average the per-layer scores across all levels of a constructed taxonomy.

LLM-as-a-Judge. To evaluate a constructed taxonomy holistically, we adopt the LLM-as-a-Judge approach of Zhang et al. (2025), who design an evaluation based on taxonomy metrics from Kaplan et al. (2022), for a set of four evaluation categories:

- **Clarity:** consists of four metrics, namely *precision*, *ambiguity*, *consistency*, and *accessibility*.
- **Hierarchical Coherence:** consists of *gradational specificity* and *parent-child coherence*.
- **Orthogonality:** *distinctiveness* and *non-overlap*.
- **Completeness:** *coverage*, *depth*, and *balance*.

These metrics are defined in the LLM-as-a-Judge prompt we use, which is slightly adapted from Zhang et al. (2025). This is found in Table 5 of the Appendix. We use GPT-5.4-NANO (2026-03-17) (Singh et al., 2025) with a temperature of 0.

Domain coverage. To evaluate the generalizability and representativeness of a constructed taxonomy, we perform domain coverage tests, adapted from the evaluation of Li et al. (2025).

This metric measures how well the base level statements (“labels”) of a taxonomy can be used to map and classify real-world data. We first prepare a test set from the held-out data from each dataset (see Section 3), namely a random sample 10k job postings from NLx and 1k postings from USAJOBS. We sentence tokenize all of these postings (using NLTK), and prompt GPT-5.4-NANO and Google’s GEMINI-3.1-FLASH-LITE (with temperature of 0) to label each sentence as an AI-related job skill or not. The full prompt used for this is found in Table 6 of the Appendix.

With the labeled test sets, we evaluate coverage by running embedding-based semantic search procedures at various *confidence levels*, or thresholds.

Dataset	Configuration		Silh.	LLM-as-a-Judge (Category Avg.)				Lenient Coverage @ τ				Strict Coverage @ τ				Best Util.
	Aug. / Soft	Pct.		Clar.	H.Coh.	Orth.	Comp.	0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6	
NLx	Y / Y	25	0.047	2.62	3.08	1.62	3.02	0.472	0.595	0.585	0.346	0.498	0.545	0.447	0.227	0.580
	Y / Y	50	0.056	2.86	3.25	1.81	2.96	0.470	0.592	0.603	0.376	0.497	0.547	0.464	0.253	0.604
	Y / Y	75	0.092	3.07	3.57	2.21	3.29	0.462	0.579	0.630	0.427	0.494	0.557	0.494	0.297	0.610
	Y / N	25	-0.000	2.89	3.07	1.79	2.90	0.475	0.592	0.576	0.316	0.500	0.540	0.436	0.201	0.580
	Y / N	50	0.032	2.83	3.15	1.85	2.93	0.473	0.590	0.597	0.362	0.499	0.543	0.458	0.241	0.608
	Y / N	75	0.074	2.82	3.36	2.14	2.90	0.465	0.576	0.626	0.402	0.495	0.553	0.488	0.276	0.604
	N / Y	25	0.073	2.75	3.20	1.90	3.07	0.460	0.554	0.702	0.546	0.493	0.600	0.599	0.398	0.468
	N / Y	50	0.111	3.25	3.75	2.25	3.22	0.457	0.538	0.707	0.602	0.489	0.598	0.636	0.445	0.814
	N / Y	75	0.065	2.88	3.25	2.00	3.17	0.456	0.504	0.682	0.663	0.489	0.570	0.672	0.505	0.844
	N / N	25	0.026	2.92	3.33	1.83	2.89	0.461	0.558	0.700	0.533	0.497	0.596	0.589	0.385	0.447
	N / N	50	0.079	2.83	3.50	1.83	3.22	0.459	0.541	0.705	0.599	0.493	0.596	0.623	0.441	0.837
	N / N	75	0.096	2.92	3.50	2.00	3.11	0.456	0.510	0.685	0.648	0.489	0.576	0.662	0.488	0.851
USAJOBS	Y / Y	25	0.155	3.34	2.70	2.14	2.73	0.485	0.514	0.543	0.319	0.510	0.518	0.469	0.244	0.294
	Y / Y	50	0.159	3.40	2.81	2.13	2.84	0.483	0.496	0.573	0.436	0.506	0.526	0.518	0.354	0.300
	Y / Y	75	0.430	2.75	1.75	1.50	2.33	0.481	0.490	0.586	0.584	0.501	0.525	0.559	0.466	0.823
	Y / N	25	0.561	3.37	2.76	2.28	2.78	0.486	0.514	0.553	0.330	0.507	0.518	0.476	0.254	0.296
	Y / N	50	0.516	3.47	2.79	2.26	2.95	0.483	0.498	0.576	0.425	0.506	0.525	0.517	0.345	0.317
	Y / N	75	0.577	3.08	2.50	1.50	2.56	0.481	0.491	0.583	0.556	0.500	0.531	0.552	0.446	0.839
	N / Y	25	0.345	3.33	1.61	1.56	2.04	0.483	0.492	0.587	0.587	0.505	0.534	0.572	0.469	0.872
	N / Y	50	0.362	3.50	1.93	2.07	2.05	0.481	0.489	0.577	0.607	0.503	0.527	0.588	0.486	0.862
	N / Y	75	0.112	3.62	3.30	2.46	3.06	0.480	0.484	0.542	0.644	0.499	0.516	0.637	0.526	0.829
	N / N	25	0.673	3.33	1.71	1.67	1.97	0.482	0.491	0.586	0.589	0.506	0.533	0.571	0.470	0.835
	N / N	50	0.703	2.50	1.50	1.50	2.33	0.482	0.489	0.571	0.601	0.503	0.529	0.582	0.483	0.852
	N / N	75	0.481	3.50	3.05	2.38	3.08	0.481	0.484	0.541	0.636	0.501	0.515	0.625	0.515	0.821

Table 1: Consolidated evaluation results of taxonomies constructed with TAXONOMYBUILDER across the NLx and USAJOBS Datasets. The different configurations are represented by whether they include data **augmentation** (Y) or not (N), and similarly for **soft** clustering or not. LLM-as-a-Judge categories represent the average of their component metrics. The *Best Util* column reports the Label Utilization corresponding to the threshold (τ) that achieved the highest *strict* coverage. **Bold** values indicate the best score in each column per dataset.

We choose $\tau \in \{0.9, 0.8, 0.7, 0.6\}$ indicating that a sentence in question must match a taxonomy label with cosine similarity $> \tau$ to be classified as “AI”. To calculate the similarity score for thresholding, we use an ensemble of three embedding models, akin to ensemble used for class scoring in Section 4. At each τ , we report the macro-F1 score, which represents the *coverage rate*. This includes a *lenient* score, where the ground truth of “AI” (label = 1) occurs when at least one LLM label is 1 (see above), and *strict* where both LLMs agree. At the best (strict) coverage rate, we also report the *label utilization rate*, i.e., the percentage of taxonomy statements successfully matched to the test set in the optimal (highest coverage) scenario. Together, these metrics illustrate both *how well* a taxonomy represents a domain and *how efficiently* this is done with respect to the breadth of the taxonomy.

6 Results and Statistical Analysis

The consolidated results are presented in Table 1, where LLM-as-a-Judge scores are reported as averages across the four categories. The label utilization rate is only reported for the corresponding best strict coverage rate. For the complete results, we refer the readers to Table 7 of the Appendix.

An analysis of the results shows that the best results are achieved generally when *either* data augmentation *or* soft clustering is used, and a majority (22/28) of the best scores per metric are achieved at or above the 50th percentile filtering mark. To verify these observations, as well as to investigate the main effects of the different taxonomy construction configurations, we perform a macro-level statistical analysis using a factorial ANOVA test. The ANOVA test was conducted in the aggregated metrics (i.e., as presented in Table 7) to evaluate the effects of data augmentation, percentile filtering, and soft clustering on our evaluation metrics ($2 \times 3 \times 2$ factors). The statistically significant results of this analysis are presented in Table 2, and the full results can be found in Table 8 of the Appendix.

7 Discussion

In the following, we reflect on the main findings of our work, and we discuss paths for future research.

Less is more. Our work centers on the important question of systemically understanding complex, shifting, and evolving domains such as *AI skills*, specifically through the construction of representative taxonomies from massive text corpora. While

Dataset	Metric	Significant Factor	F-value	p-value	η^2	Direction of Effect / Explanation
NLx	Silhouette Score	Percentile	4.36	0.059 [†]	0.387	Stricter percentile filtering improves cluster fit
		Augmentation	3.68	0.097 [†]	0.163	Augmentation marginally decreases cluster fit
	LLM: Orthogonality	Percentile	3.45	0.091 [†]	0.467	Stricter percentile filtering improves orthogonality
	LLM: Completeness	Soft Clustering	3.92	0.088 [†]	0.223	Soft clustering marginally improves completeness
	Coverage (Strict, $\tau = 0.8$)	Augmentation	40.81	<0.001*	0.836	Augmentation severely reduces strict coverage rate
USAJOBS	Label Utilization	Percentile	4.60	0.053 [†]	0.475	Stricter filtering drastically improves label utilization
	Silhouette Score	Soft Clustering	18.76	0.003*	0.713	Soft clustering severely degrades cluster fit
	Label Utilization	Augmentation	13.48	0.008*	0.518	Augmentation severely reduces label utilization

* Significant at $p < 0.05$; [†] Marginally significant at $p < 0.10$.

Table 2: Factorial ANOVA results for effects of taxonomy construction configurations on the reported metrics. Only significant ($p < 0.05$) and marginally significant ($p < 0.10$) effects are reported. Effect sizes are denoted by η^2 .

the presence of such data may potentially serve as a fertile foundation for the challenging task of automatic taxonomy construction, we demonstrate that there are important design decisions revolving around *how to leverage this data*. Such questions have not been explored by previous works, who operate on much more compact datasets.

Several of our key findings point to a *less is more* philosophy when building taxonomies from large-scale data. We find that data augmentation does not help the quality of the resulting taxonomies, demonstrated by the fact that only one LLM-as-a-Judge category was won with data augmentation (out of the eight settings). In this light, using soft clustering after *excluding* data augmentation proved to be a winning recipe, achieving the best score in 20/28 cases. Thus, soft clustering likely only makes sense when not including augmented samples.

Studying the effect of percentile-based filtering measure reveals a similar trend. Taxonomies built with a weaker 25th percentile filter won zero LLM-as-a-Judge categories, and interestingly, only outperform all other taxonomies at very strict (high τ) coverage tests. As such, being too lenient with data inclusion (i.e., including more noisy points), can lead to sub-optimal domain-specific taxonomies.

Taxonomy building as trade-off balancing. We highlight another important finding, specifically in evaluating the *coverage* of taxonomies in a custom domain. One can observe an inverse relationship between data inclusion and coverage performance. Interestingly, at the strictest coverage threshold ($\tau = 0.9$), the coverage rates (both strict and lenient) are won exclusively by taxonomies built with a 25th percentile filter (more data). This is exactly the opposite for the lowest threshold ($\tau = 0.6$), at which exclusively 75th percentile taxonomies outperform all others. Intuitively, this might make

sense, as larger, potentially noisy taxonomies may provide wide coverage and thereby excel at higher similarity thresholds, whereas more concise taxonomies can still perform better at lower thresholds due to being more well-defined and less cluttered.

Such findings point to the need to study *trade-offs* in automatic taxonomy construction, particularly in the balance between domain coverage and intra-taxonomy variability (i.e., noisiness). Data augmentation and more permissive filtering could serve certain use cases well, in which breadth is prioritized over interpretability. On the other hand, taxonomies that are repetitive or not focused enough on the domain in question may only increase ambiguity and lack usability or complicate comprehension. Therefore, although our results support the recommendation of less (data) being more powerful, we caution that this is case-specific, and ultimately, trade-offs should be carefully weighed.

8 Conclusion

We propose TAXONOMYBUILDER as a generalizable tool to taxonomize custom domains from large-scale text corpora. Using TAXONOMYBUILDER, we conduct a systematic investigation into the practical considerations of leveraging large amounts of data, focusing on questions of optimal data use for grounding taxonomy construction. Our results highlight that less is more, and data augmentation should be exercised with caution. In addition, performing selective filtering of candidate inputs to the construction process is often advantageous. As future work, we recommend (1) the continued exploration of design decisions in automatic taxonomy construction, (2) the carrying out of ablation studies regarding LLM choice, clustering parametrization, and taxonomy pruning, and (3) the formalization of trade-off-specific taxonomy evaluation metrics, focusing on real-world generalizability.

Limitations

We acknowledge the main limitations of our work, primarily in that we did not study the impact our investigated factors (data augmentation, soft clustering, and percentile filtering) on other proposed LLM-based taxonomy construction methods. Rather, we focused on our proposed TAXONOMY-BUILDER method, as this framework was custom built for this systematic study, thereby not necessitating the modification of previous methods.

We also note the lack of an ablation study, as mentioned in the Conclusion, which we propose as direct follow-up work to ours. Important points of investigation could include but are not limited to: choice of LLM for label generation, the design of more intelligent methods to reduce noise in the context provided to the LLM (while maintaining representativeness, and conducting top-down checks to counteract bottom-up error propagation (e.g., induced via irrelevant candidate sentences).

Ethics Statement

We respect the confidentiality of data shared under the agreement with NLx and publicly release only aggregate information (the constructed taxonomies), and mitigate restrictions on sharing with the inclusion of a public corpus in the same domain.

We also note that the constructed taxonomies provided in our repository have been redacted to remove company names remaining in the leaf nodes.

References

- Daron Acemoglu, David Autor, and Simon Johnson. 2026. [Building Pro-Worker Artificial Intelligence](#).
- Liudmila Alekseeva, José Azar, Mireia Giné, Sampsá Samila, and Bledi Taska. 2021. [The demand for AI skills in the labor market](#). *Labour Economics*, 71:102002.
- Stefano Baruffaldi, Brigitte van Beuzekom, H el ene Derris, Dietmar Harhoff, Nandan Rao, David Rosenfeld, and Mariagrazia Squicciarini. 2020. [Identifying and measuring developments in artificial intelligence: Making the impossible possible](#). *OECD Science, Technology and Industry Working Papers*, 2020(05).
- Ricardo JGB Campello, Davoud Moulavi, and J org Sander. 2013. [Density-based clustering based on hierarchical density estimates](#). In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.
- Boqi Chen, Fandi Yi, and D aniel Varr o. 2023. [Prompting or fine-tuning? a comparative study of large language models for taxonomy construction](#). In *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pages 588–596.
- Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2004. [Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text](#). In *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI’04*, page 435–439, NLD. IOS Press.
- Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. [Supervised distributional hypernym discovery via domain adaptation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 424–435, Austin, Texas. Association for Computational Linguistics.
- Trevor Fountain and Mirella Lapata. 2012. [Taxonomy induction using hierarchical random graphs](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 466–476, Montr el, Canada. Association for Computational Linguistics.
- Morgan R. Frank, David Autor, James E. Bessen, Erik Brynjolfsson, Manuel Cebrian, David J. Deming, Maryann Feldman, Matthew Groh, Jos e Lobo, Esteban Moro, Dashun Wang, Hyejin Youn, and Iyad Rahwan. 2019. [Toward understanding the impact of artificial intelligence on labor](#). *Proceedings of the National Academy of Sciences*, 116(14):6531–6539.
- Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. [Learning semantic hierarchies via word embeddings](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, Maryland. Association for Computational Linguistics.
- Avi Goldfarb, Bledi Taska, and Florenta Teodoridis. 2023. [Could machine learning be a general purpose technology? A comparison of emerging technologies using data from online job postings](#). *Research Policy*, 52(1):104653.
- Allan D Gordon. 1996. [Hierarchical classification](#). *Clustering and classification*, page 65.
- Marti A. Hearst. 1992. [Automatic acquisition of hyponyms from large text corpora](#). In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.
- Chen Huang and Guoxiu He. 2025. [Text clustering as classification with llms](#). In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2025*, page 374–384, New York, NY, USA. Association for Computing Machinery.

- Dino Ienco and Rosa Meo. 2008. [Towards the automatic construction of conceptual taxonomies](#). In *International Conference on Data Warehousing and Knowledge Discovery*, pages 327–336. Springer.
- Angelika Kaplan, Thomas Kühn, Sebastian Hahner, Niko Benkler, Jan Keim, Dominik Fuchß, Sophie Corallo, and Robert Heinrich. 2022. [Introducing an evaluation method for taxonomies](#). In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering, EASE '22*, page 311–316, New York, NY, USA. Association for Computing Machinery.
- Priyanka Kargupta, Nan Zhang, Yunyi Zhang, Rui Zhang, Prasenjit Mitra, and Jiawei Han. 2025. [TaxoAdapt: Aligning LLM-based multidimensional taxonomy construction to evolving research corpora](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29834–29850, Vienna, Austria. Association for Computational Linguistics.
- Zornitsa Kozareva and Eduard Hovy. 2010. [A semi-supervised method to learn and construct taxonomies using the web](#). In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1110–1118, Cambridge, MA. Association for Computational Linguistics.
- Avishek Lahiri, Yufang Hou, and Debarshi Kumar Sanyal. 2025. [TaxoAlign: Scholarly taxonomy generation using language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30203–30223, Suzhou, China. Association for Computational Linguistics.
- Josh Lerner and Jean Tirole. 2002. [Some Simple Economics of Open Source](#). *The Journal of Industrial Economics*, 50(2):197–234.
- Nan Li, Bo Kang, and Tijl De Bie. 2025. [Building data-driven occupation taxonomies: A bottom-up multi-stage approach via semantic clustering and multi-agent collaboration](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1596–1614, Suzhou (China). Association for Computational Linguistics.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. [Towards general text embeddings with multi-stage contrastive learning](#). *Preprint*, arXiv:2308.03281.
- Lightcast. 2023. [The Lightcast open skills taxonomy](#). Technical report, Lightcast. White paper. Formerly Burning Glass Technologies/Emsi Burning Glass.
- Bowen Lou and Lynn Wu. 2021. [AI on Drugs: Can Artificial Intelligence Accelerate Drug Development? Evidence from a Large-Scale Examination of Bio-Pharma Firms](#). *Management Information Systems Quarterly*, 45(3):1451–1482.
- Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. [End-to-end reinforcement learning for automatic taxonomy induction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2462–2472, Melbourne, Australia. Association for Computational Linguistics.
- Oleksandr Marchenko and Danylo Dvoichenkov. 2024. [Taxorankconstruct: A novel rank-based iterative approach to taxonomy construction with large language models](#). In *ISS@ IT&I*, pages 11–27.
- Anoush Margaryan. 2023. [Artificial intelligence and skills in the workplace: An integrative research agenda](#). *Big Data & Society*, 10(2):20539517231206804.
- Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Toby Walsh, Armin Hamrah, Lapo Santarasci, and 4 others. 2025. [Artificial Intelligence Index Report 2025](#). *arXiv preprint*. ArXiv:2504.07139 [cs].
- Stephen Meisenbacher, Tim Schopf, Weixin Yan, Patrick Holl, and Florian Matthes. 2024. [An improved method for class-specific keyword extraction: A case study in the German business registry](#). In *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*, pages 159–165, Vienna, Austria. Association for Computational Linguistics.
- Ryszard S Michalski and Robert E Stepp. 1983. [Learning from observation: Conceptual clustering](#). In *Machine learning*, pages 331–363. Elsevier.
- Viktor Moskvoretskii, Irina Nikishina, Ekaterina Nemnova, Alina Lobanova, Alexander Panchenko, and Chris Biemann. 2026. [Large language models for creation, enrichment and evaluation of taxonomic graphs](#). *Semantic Web*, 17(1):22104968251404186.
- Elham Motamedi, Inna Novalija, and Luis Rei. 2026. [Semi-automatic hierarchical taxonomy creation from existing taxonomies with large language models](#). *Business & Information Systems Engineering*, pages 1–23.
- Abhishek Nagaraj. 2022. [The Private Impact of Public Data: Landsat Satellite Maps Increased Gold Discoveries and Encouraged Entry](#). *Management Science*, 68(1):564–582.
- Abhishek Nagaraj, Esther Shears, and Mathijs de Vaan. 2020. [Improving data access democratizes and diversifies science](#). *Proceedings of the National Academy of Sciences*, 117(38):23490–23498.
- Engineering National Academies of Sciences, and Medicine. 2025. [Artificial Intelligence and the Future of Work](#). The National Academies Press, Washington, DC.

- Raúl Navarro-Almanza, Reyes Juárez-Ramírez, Guillermo Licea, and Juan R Castro. 2020. [Automated ontology extraction from unstructured texts using deep learning](#). In *Intuitionistic and Type-2 fuzzy logic enhancements in neural and optimization algorithms: Theory and applications*, pages 727–755. Springer.
- Robert C Nickerson, Upkar Varshney, and Jan Muntermann. 2013. [A method for taxonomy development and its application in information systems](#). *European journal of information systems*, 22(3):336–359.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Sebastian Padó and Mirella Lapata. 2007. [Dependency-based construction of semantic space models](#). *Computational Linguistics*, 33(2):161–199.
- Marius Pasca. 2004. [Acquisition of categorized named entities for web search](#). In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM '04*, page 137–145, New York, NY, USA. Association for Computing Machinery.
- Hoifung Poon and Pedro Domingos. 2010. [Unsupervised ontology induction from text](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 296–305, Uppsala, Sweden. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- William Resh, Keunyoung Lee, and Yi Ming. 2025. [U.S. Federal Civil Position Job Postings \(2018-2023\)](#).
- Tim Schopf, Daniel Braun, and Florian Matthes. 2020. [Semantic label representations with lbl2vec: A similarity-based approach for unsupervised text classification](#). In *International conference on web information systems and technologies*, pages 59–73. Springer.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267.
- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. [Semantic taxonomy induction from heterogeneous evidence](#). In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia. Association for Computational Linguistics.
- Prasanna B. Tambe. 2026. [Reskilling the Workforce for AI: Domain Expertise and Algorithmic Literacy](#). *Management Science*, 72(1):515–537.
- Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. [OntoLearn reloaded: A graph-based algorithm for taxonomy induction](#). *Computational Linguistics*, 39(3):665–707.
- Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. [EmbeddingGemma: Powerful and lightweight text representations](#). *Preprint*, arXiv:2509.20354.
- Binh Vu, Rashmi Govindrajua Naik, Bao Khanh Nguyen, Sina Mehraeen, and Matthias Hemmje. 2025. [Automated taxonomy construction using large language models: A comparative study of fine-tuning and prompt engineering](#). *Eng*, 6(11):283.
- Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. 2024. [Tnt-llm: Text mining at scale with large language models](#). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24*, page 5836–5847, New York, NY, USA. Association for Computing Machinery.
- Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. 2024. [Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples](#). In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24*, page 3093–3102, New York, NY, USA. Association for Computing Machinery.
- Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. [Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering](#). In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18*, page 2701–2709, New York, NY, USA. Association for Computing Machinery.
- Haiqi Zhang, Zhengyuan Zhu, Zeyu Zhang, and Chengkai Li. 2025. [LLMTaxo: Leveraging large language models for constructing taxonomy of factual](#)

claims from social media. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19627–19641, Vienna, Austria. Association for Computational Linguistics.

Kun Zhu, Lizi Liao, Yuxuan Gu, Lei Huang, Xiaocheng Feng, and Bing Qin. 2025. [Context-aware hierarchical taxonomy generation for scientific papers via LLM-guided multi-aspect clustering](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15616–15634, Suzhou, China. Association for Computational Linguistics.

Ben Zweig. 2026. *Job Architecture: Building a Language for Workforce Intelligence*. Wiley, Hoboken.

A Supplemental

Prompts. Table 3 provides the LLM prompt used to create labels for collections of candidate sentences, in order to form a level of the taxonomy being constructed. Table 4 provides the prompt for the aggregation process (redundancy reduction).

On the evaluation side, Table 5 outlines the full LLM-as-a-Judge prompt, based on Zhang et al. (2025). Table 6 contains the prompt used to label the test sets for NLx and USAJOBS.

Results. Table 7 details the complete experimental results, providing a more full account than Table 1. Table 8 presents the full ANOVA test results, in supplement to the abridged version of Table 2.

You will be given a list of statements.
Your job is to produce a single sentence that summarizes these statements into a coherent task / skill description.
A task is a specific activity or function that a person would be required to do on the job.
A skill is the ability to perform a specific task and apply knowledge, particularly in the work context.
Avoid using generalizations like "various" and "across domains".
Answer simply with the generated description, nothing else is required.

Provide your feedback as follows:

Output::
Description: (GENERATED DESCRIPTION)

Here are some examples:
statements: ['develop or implement data analysis algorithms.', 'design and apply bioinformatics algorithms including unsupervised and supervised machine learning, dynamic programming, or graphic algorithms.', 'analyze or manipulate bioinformatics data using software packages, statistical applications, or data mining techniques.', 'develop or apply data mining and machine learning algorithms.', 'develop machine learning operations.', '+ develop and implement new computational and statistical methods.', 'he/she may also evaluate and develop novel algorithms and approaches for data analysis.', '* you will develop custom data models and algorithms to apply to data sets.']

Output::
Description: Develop or apply data mining and machine learning algorithms.

statements: ['• basic computer skills.', 'basic computer software skills.', 'basic computer software skills, i.e.', 'be able to perform basic computer skills.', 'basic skills in use of computers and software programs.', 'skills: basic computer skills.', '* basic programming experience.', 'computer skills and basic knowledge of software applications.', '+basic computer skills.', '- basic computer skills.', 'skills: basic computer knowledge.', 'ability to perform basic computer skills.', 'basic problem solving skills associated with software applications used is expected.', '- computer programming skills.']

Output::
Description: Has basic computer skills, including programming and use of computers and software programs.

Now here are the actual statements.

statements: {CANDIDATES}

Output::
Description:

Table 3: Prompt for creation of leaf-level taxonomy entries.

You will be given a list of statements.
They all express a similar meaning, with slight variations and differences.
Your job is to produce a single sentence that summarizes these statements into one coherent description, that captures the essence and intricacies of all statements.
Answer simply with the generated description, nothing else is required.

Provide your feedback as follows:

Output::
Description: (GENERATED DESCRIPTION)

Now here are the actual statements.

statements: {LABELS}

Output::
Description:

Table 4: Prompt for aggregation of similar labels (for label pruning).

You will be given a taxonomy related to AI skills in the workplace. Please evaluate the taxonomy in the json format using the evaluation metrics provided. Give each evaluation criteria a score from 1-5. No additional explanation is necessary. Output only the json results.

Below is the taxonomy:
{TAXONOMY JSON STRING}

Here are the metrics:

Taxonomy Evaluation Metrics

Give the score from [1-5] 5: best/strongly agree, 1: worst/strongly disagree

Clarity: Assess whether the topic labels are clear, precise, and unambiguous.
Purpose: Ensure that each topic label communicates its content effectively to avoid confusion.

Evaluation Criteria:

- Precision: Each topic label uses specific and well-defined terms.

o Score:

- Unambiguity: Topic labels should have only one interpretation, preventing misunderstanding.

o Score:

- Consistency: Use of terminology is consistent across all levels of the taxonomy.

o Score:

- Accessibility: Language is straightforward, avoiding jargon where possible unless it is standard within the covered domain.

o Score:

Hierarchical Coherence: Assess whether the taxonomy follows a clear and meaningful hierarchical structure.

Purpose: Ensure that the taxonomy's structure facilitates easy navigation and understanding by clearly organizing information from the most general to the most specific.

Evaluation Criteria:

- Gradational Specificity: There is a logical progression from broader to more specific categories.

o Score:

- Parent-Child Coherence: Parent-child relationships are well-formed, ensuring that child nodes logically belong to their parent nodes.

o Score:

- Consistency: The hierarchy maintains consistent levels of detail throughout the taxonomy, ensuring that no topics are too broad or too narrow relative to others at the same level.

o Score:

Orthogonality: Assess whether the topics are well-differentiated without duplication.

Purpose: Maintain distinct boundaries between topics to ensure that each topic captures unique aspects of the domain.

Evaluation Criteria:

- Distinctiveness: Topics at each level progressively add meaningful distinctions rather than just rephrasing broader topics.

o Score:

- Non-overlap: For each topic, there is minimal to no overlap in the scope or content with other topics.

o Score:

Completeness: Assess whether the taxonomy captures a broad and representative set of topics across different aspects of the domain.

Purpose: Cover as many areas of the topic to ensure the taxonomy is comprehensive.

Evaluation Criteria:

- Domain Coverage: The taxonomy covers a variety of significant aspects of the domain it represents.

o Score:

- Depth: The taxonomy provides sufficient depth in each branch to capture nuanced distinctions within topics.

o Score:

- Balance: The topics are evenly distributed across the taxonomy. This involves assessing whether some branches are disproportionately detailed while others are underdeveloped, which could lead to an imbalance that might skew the taxonomy's effectiveness and navigability.

o Score:

Important: ensure that the output is in json, and following the above-given criteria names exactly.

Now please provide the scores.

Table 5: LLM-as-a-Judge prompt for comprehensive taxonomy evaluation.

You will be given a dictionary of statements from a job posting, each mapped to a unique ID (integer).

Your job is to indicate which of the statements (marked by their IDs), contain information related to any Artificial Intelligence (AI) task, skill, activity, expertise, or requirement.

"AI-related" can also include soft skills required for AI jobs in addition to technical abilities or tasks.

Make sure, though, that only AI-related statements are chosen!

Answer simply with a comma-separated Python list of ONLY the statement IDs marked as AI-related. Make sure to stick to this output format!

Provide your feedback as follows:

Output::

Classification: (LIST OF IDENTIFIED AI-RELATED STATEMENT IDs)

Now here are the actual statements.

texts: {DICTIONARY OF MAPPED JOB POSTING SENTENCES}

Output::

Classification:

Table 6: Prompt for LLM labeling of test set for domain coverage tests.

Data	Config		Silh.	Clarity				H. Coh.		Orth.		Comp.			Cov (0.9)			Cov (0.8)			Cov (0.7)			Cov (0.6)		
	Aug/Soft	Pct		Pr	Un	Co	Ac	Gr	PC	Di	NO	Co	De	Ba	L	S	U	L	S	U	L	S	U	L	S	U
NLx	Y/Y	25	0.047	3.14	2.33	2.14	2.86	2.86	3.29	2.10	1.14	3.62	3.43	2.00	0.472	0.498	0.101	0.595	0.545	0.580	0.585	0.447	0.813	0.346	0.227	0.847
	Y/Y	50	0.056	3.50	2.69	2.13	3.13	3.06	3.44	2.19	1.44	3.75	3.13	2.00	0.470	0.497	0.091	0.592	0.547	0.604	0.603	0.464	0.838	0.376	0.253	0.871
	Y/Y	75	0.092	3.71	2.86	2.43	3.29	3.43	3.71	2.57	1.86	4.00	3.71	2.14	0.462	0.494	0.058	0.579	0.557	0.610	0.630	0.494	0.867	0.427	0.297	0.899
	Y/N	25	-0.000	3.43	2.43	2.14	3.57	2.86	3.29	2.29	1.29	3.71	3.00	2.00	0.475	0.500	0.113	0.592	0.540	0.580	0.576	0.436	0.815	0.316	0.201	0.852
	Y/N	50	0.032	3.50	2.50	2.20	3.10	2.90	3.40	2.30	1.40	3.60	3.20	2.00	0.473	0.499	0.104	0.590	0.543	0.608	0.597	0.458	0.850	0.362	0.241	0.881
	Y/N	75	0.074	3.57	2.71	2.00	3.00	3.00	3.71	2.57	1.71	3.57	3.14	2.00	0.465	0.495	0.067	0.576	0.553	0.604	0.626	0.488	0.870	0.402	0.276	0.907
	N/Y	25	0.073	3.40	2.40	2.00	3.20	2.80	3.60	2.40	1.40	3.80	3.40	2.00	0.460	0.493	0.042	0.554	0.600	0.468	0.702	0.599	0.824	0.546	0.398	0.897
	N/Y	50	0.111	4.00	3.00	2.83	3.17	3.50	4.00	2.67	1.83	4.00	3.67	2.00	0.457	0.489	0.031	0.538	0.598	0.460	0.707	0.636	0.814	0.602	0.445	0.908
	N/Y	75	0.065	3.50	2.50	2.50	3.00	3.00	3.50	2.50	1.50	4.00	3.50	2.00	0.456	0.489	0.025	0.504	0.570	0.447	0.682	0.672	0.844	0.663	0.505	0.936
	N/N	25	0.026	3.67	2.67	2.00	3.33	3.00	3.67	2.33	1.33	3.67	3.00	2.00	0.461	0.497	0.056	0.558	0.596	0.447	0.700	0.589	0.829	0.533	0.385	0.916
	N/N	50	0.079	3.67	2.67	2.00	3.00	3.33	3.67	2.33	1.33	4.00	3.67	2.00	0.459	0.493	0.043	0.541	0.596	0.447	0.705	0.623	0.837	0.599	0.441	0.924
	N/N	75	0.096	3.67	2.67	2.33	3.00	3.33	3.67	2.33	1.67	4.00	3.33	2.00	0.456	0.489	0.031	0.510	0.576	0.451	0.685	0.662	0.851	0.648	0.488	0.943
USAJOBS	Y/Y	25	0.155	3.92	3.45	2.61	3.37	2.16	3.24	2.45	1.83	3.02	2.55	2.63	0.485	0.510	0.060	0.514	0.518	0.294	0.543	0.469	0.846	0.319	0.244	0.931
	Y/Y	50	0.159	3.97	3.54	2.75	3.33	2.18	3.43	2.38	1.89	3.11	2.78	2.64	0.483	0.506	0.070	0.496	0.526	0.300	0.573	0.518	0.848	0.436	0.354	0.933
	Y/Y	75	0.430	3.50	2.50	2.00	3.00	2.00	1.50	2.00	1.00	3.00	2.00	2.00	0.481	0.501	0.035	0.490	0.525	0.253	0.586	0.559	0.823	0.584	0.466	0.924
	Y/N	25	0.561	3.90	3.51	2.73	3.34	2.19	3.32	2.57	2.00	3.01	2.59	2.73	0.486	0.507	0.066	0.514	0.518	0.296	0.553	0.476	0.835	0.330	0.254	0.922
	Y/N	50	0.516	3.96	3.62	2.87	3.44	2.16	3.41	2.50	2.03	3.09	2.84	2.91	0.483	0.506	0.070	0.498	0.525	0.317	0.576	0.517	0.855	0.425	0.345	0.948
	Y/N	75	0.577	3.33	2.67	2.67	3.67	2.33	2.67	2.00	1.00	3.00	2.00	2.67	0.481	0.500	0.028	0.491	0.531	0.271	0.583	0.552	0.839	0.556	0.446	0.947
	N/Y	25	0.345	3.67	3.44	1.89	4.33	1.56	1.67	1.44	1.67	1.67	1.78	2.67	0.483	0.505	0.053	0.492	0.534	0.275	0.587	0.572	0.872	0.587	0.469	0.947
	N/Y	50	0.362	3.57	3.71	2.29	4.43	1.57	2.29	1.71	2.43	1.86	1.71	2.57	0.481	0.503	0.028	0.489	0.527	0.245	0.577	0.588	0.862	0.607	0.486	0.946
	N/Y	75	0.112	4.26	3.85	3.07	3.30	2.67	3.93	2.85	2.07	3.33	3.00	2.85	0.480	0.499	0.014	0.484	0.516	0.157	0.542	0.637	0.829	0.644	0.526	0.968
	N/N	25	0.673	3.42	3.42	2.33	4.17	1.58	1.83	1.50	1.83	1.67	1.67	2.58	0.482	0.506	0.050	0.491	0.533	0.255	0.586	0.571	0.835	0.589	0.470	0.950
	N/N	50	0.703	2.50	2.00	2.00	3.50	1.50	1.50	2.00	1.00	3.00	2.50	1.50	0.482	0.503	0.056	0.489	0.529	0.237	0.571	0.582	0.852	0.601	0.483	0.959
	N/N	75	0.481	4.20	3.70	2.90	3.20	2.40	3.70	2.75	2.00	3.50	3.05	2.70	0.481	0.501	0.037	0.484	0.515	0.174	0.541	0.625	0.821	0.636	0.515	0.972

Table 7: Comprehensive evaluation results of taxonomies constructed with TAXONOMYBUILDER across the NLx and USAJOBS Datasets. The different configurations are represented by whether they include data augmentation (Y) or not (N), and similarly for soft clustering or not. *Pct* denotes the percentile filtering threshold. All LLM-as-a-Judge metrics are reported in the order they were introduced in the paper, and are abbreviated with two representative letters. Domain Coverage (*Cov*) metrics include Lenient (L), Strict (S), and Utilization (U) across four thresholds.

Dataset	Evaluation Metric	Factor	F-value	p-value	η^2
NLx	Silhouette Score	Augmentation	3.68	0.097 [†]	0.163
		Soft Clustering	3.11	0.121	0.138
		Percentile	4.36	0.059 [†]	0.387
	LLM: Clarity	Augmentation	0.63	0.454	0.065
		Soft Clustering	0.14	0.716	0.015
		Percentile	0.91	0.444	0.190
	LLM: Hierarchical Coherence	Augmentation	2.82	0.137	0.190
		Soft Clustering	0.09	0.770	0.006
		Percentile	2.48	0.153	0.334
	LLM: Orthogonality	Augmentation	0.48	0.512	0.032
		Soft Clustering	0.38	0.555	0.026
		Percentile	3.45	0.091 [†]	0.467
LLM: Completeness	Augmentation	2.98	0.128	0.170	
	Soft Clustering	3.92	0.088 [†]	0.223	
	Percentile	1.84	0.228	0.209	
Strict Coverage ($\tau = 0.8$)	Augmentation	40.81	<0.001*	0.836	
	Soft Clustering	0.11	0.750	0.002	
	Percentile	0.46	0.650	0.019	
Best Label Utilization	Augmentation	3.18	0.118	0.164	
	Soft Clustering	<0.01	0.986	<0.001	
	Percentile	4.60	0.053 [†]	0.475	
USAJOBS	Silhouette Score	Augmentation	0.38	0.556	0.015
		Soft Clustering	18.76	0.003*	0.713
		Percentile	0.09	0.912	0.007
	LLM: Clarity	Augmentation	0.07	0.797	0.009
		Soft Clustering	0.25	0.634	0.033
		Percentile	0.11	0.895	0.030
	LLM: Hierarchical Coherence	Augmentation	0.83	0.392	0.094
		Soft Clustering	0.01	0.933	0.001
		Percentile	0.50	0.628	0.113
	LLM: Orthogonality	Augmentation	0.01	0.920	0.002
		Soft Clustering	0.03	0.873	0.004
		Percentile	0.03	0.973	0.008
LLM: Completeness	Augmentation	1.27	0.297	0.127	
	Soft Clustering	0.18	0.686	0.018	
	Percentile	0.79	0.489	0.158	
Strict Coverage ($\tau = 0.8$)	Augmentation	0.17	0.691	0.021	
	Soft Clustering	0.04	0.856	0.004	
	Percentile	0.48	0.640	0.117	
Best Label Utilization	Augmentation	13.48	0.008*	0.518	
	Soft Clustering	<0.01	0.974	<0.001	
	Percentile	2.77	0.130	0.213	

* Significant at $p < 0.05$; [†] Marginally significant at $p < 0.10$.

Table 8: Complete factorial ANOVA results. Significant ($p < 0.05$) and marginally significant ($p < 0.10$) effects are denoted. Effect size is represented by η^2 .

Using Topological Data Analysis to Characterize the Layers of Language Models Before and After Word Substitution Attacks

Adam Tang*¹, Catherine Liu¹, Kimberly Lopez¹, Shreya Subramanian¹,
Leif Zinn-Brooks¹, Alexia Schultz², Adaku Uchendu*²

¹Harvey Mudd College Claremont, CA, USA

²MIT Lincoln Laboratory, Lexington, MA, USA

*Corresponding authors: adtang@hmc.edu, adaku.uchendu@ll.mit.edu

Abstract

Large language models are known to be vulnerable to adversarial perturbations such as synonym-based word substitutions. However, previous analyses of adversarial influence focus only on output behavior and provide limited insight into the propagation of substitution-based input perturbations through internal representations. In this work, we introduce a topological data analysis (TDA) framework to study the structural effects of adversarial attacks on attention maps across model layers. We evaluate small encoder-based architectures (BERT, RoBERTa, DistilBERT) fine-tuned to solve binary classification on the IMDb review dataset, which were attacked using TextFooler. We convert attention maps into distance matrices and apply TDA to extract topological features, which we then compare using Wasserstein distances between original and perturbed features. In parallel, we compute a non-TDA baseline on attention maps using per-head L_1 distances between original and perturbed attentions. In addition, we analyze these models on a layer-by-layer basis. We find that adversarial perturbations induce systematic and statistically significant topological changes across layers, with the largest deviations occurring in late layers and smaller but notable effects in early layers. These patterns are consistent across models and are validated using both non-parametric (Kruskal–Wallis, Dunn) and parametric (one-way ANOVA, Tukey) tests on log-transformed Wasserstein distances. Compared to our non-TDA baseline, our results show more distinct layer-wise separation and provides a robust and interpretable framework for evaluating how adversarial perturbations alter internal model structure. Our code is publicly available at: https://github.com/angelinatsai04/mitll_clinic/tree/adam_spring¹.

¹DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D-0001 or FA8702-25-D-B002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s)

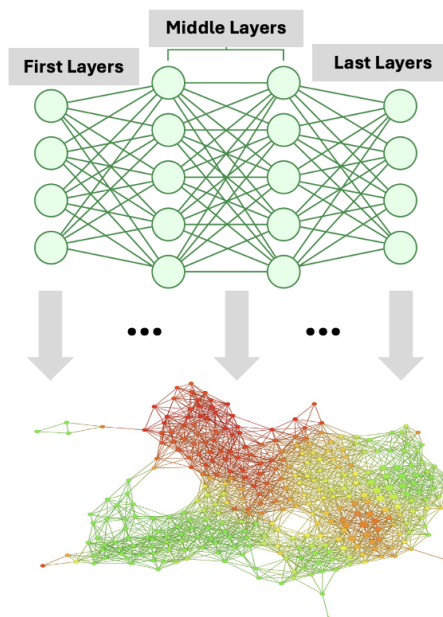


Figure 1: Illustration of using TDA to characterize the topology of the latent space, tracking adversarial activity across model layers.

1 Introduction

Large language models (LLMs) have proven to be strong tools in solving many tasks in NLP, from part-of-speech tagging to sentiment analysis to spam detection. Furthermore, LLMs are easily customizable to fit several needs (Chen et al., 2024), making them suitable for a variety of applications. However, while they provide great utility, they suffer from vulnerabilities due to the introduction of adversarial perturbations. In some cases, LLMs may produce very different outputs when semantic-preserving changes are made to the input (Arakelyan et al., 2024). One such case is point-based word substitution (Qi et al., 2021), in which

and do not necessarily reflect the views of the Department of the Air Force. © 2026 Massachusetts Institute of Technology. Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

individual words in a text input are replaced with synonyms, which may have undesirable effects on LLM-based text classification models. Therefore, before deploying LLM solutions, we need to understand what scenarios (i.e., adversarial perturbations encourage or discourage vulnerabilities in language models).

We study TextFooler (Jin et al., 2020), a word substitution attack that has a high success rate in disrupting classification predictions for fake news prediction and sentiment classification tasks (Jin et al., 2020). Like other model-agnostic attacks, TextFooler evaluates robustness at the output level, but provides few insights into how the input perturbation propagates through a model. Despite this limitation, TextFooler generates a rich set of “successful” perturbations to the output (Jin et al., 2020). Consequently, building defense strategies for this type of attack is difficult. Thus, we use TextFooler, a black-box attack, to adversarially perturb texts and probe how these perturbed texts (in comparison with the original) are represented in the embedding space of encoder-only models. Furthermore, we track how perturbations change the embedding space across layers.

To investigate the robustness of a model’s internal states, we evaluate the structure of *attention maps* (Kushnareva et al., 2021), which are intermediate representations of an input that show how much each token contributes to the meaning of each other token. We introduce the use of topological data analysis (TDA) in order to better quantify the effect of TextFooler on each layer’s attention maps and to visualize changes in the structure of a model’s internal representations across layers. See Figure 1 for an illustration on how TDA can be used to characterize the topology of each layer in a language model to investigate adversarial influence. Using these tools, we investigate the following three research questions (RQs):

- RQ1:** Can we observe distinct topologies in the attention maps of encoder models, before and after word substitution?
- RQ2:** Are these distinct topologies observable across the layers of a model?
- RQ3:** Are these distinct topologies observable across different encoder models?

Finally, our findings reveal that adversarial perturbations such as word substitution attacks do affect the topology of the embedding space of language models. In addition, we find that adversarial

influence is most noticeable in the early and late layers for the encoder models (i.e., BERT, RoBERTa, DistilBERT) investigated.

2 Related Work

Adversarial attacks have been used to evaluate the robustness of AI systems. Specifically, for LLMs, these range from prompt-based attacks (Zhang et al., 2025), using LLMs to paraphrase texts to evade author attribution (Alperin et al., 2025), to attacks that probe LLM embeddings like poisoning attacks (Fendley et al., 2025). We have also observed several defense techniques and more recently the application of TDA to detect adversarial activity (Uchendu and Le, 2024). This includes using TDA to track topological changes in hidden space as a result of adversarial attacks (Fay et al., 2026; Vu et al., 2025; Chauhan and Kaul, 2022; Perez and Reinauer, 2022). These applications suggest that TDA can characterize the topological structure well, such that changes caused by adversarial perturbations are detectable. In this study, we study the granular topological effects of adversarial perturbations by tracking its evolution across all layers.

3 Topological Data Analysis (TDA)

In order to quantify the properties of a point cloud in high-dimensional space, we move beyond using summary statistics and employ TDA. TDA methods analyze the topology induced by the connectivity of points in a space across a variety of scales, allowing for more comprehensive and informative results (Wasserman, 2018).

We use *persistent homology* (Edelsbrunner et al., 2008), one of two key methods in TDA, to analyze both the local and global features of language model internal representations. In particular, given a set of points in a metric space, we perform a *Vietoris–Rips filtration* (Sheehy, 2012) across a parameter r ; for each r , we construct a graph containing points as vertices and connecting vertices when they are within a distance of r from one another. Persistent homology analyzes the creation (“birth”) and filling-in (“death”) of topological features within this graph, such as connected components (H_0 features), loops (H_1 features), and voids (H_2 features) (Uchendu et al., 2024). These features are compiled into a *persistence diagram* (Cohen-Steiner et al., 2005), a plot which shows birth and death values (b_i, d_i) for each feature in

the filtration. *Persistence* describes how long each topological feature lasts in the filtration and is given by $\text{Persistence}((b_i, d_i)) = d_i - b_i$. This is the vertical distance from the feature to the $y = x$ diagonal on the diagram.

Persistent homology is a suitable tool for our task since attention maps are a natural way to quantify the distance between token representations. Unlike hidden states, the space induced by attention values is not Euclidean (Kobayashi et al., 2020; Vaswani et al., 2017); however, persistent homology does not require a Euclidean space (Edelsbrunner et al., 2008; Wasserman, 2018). In addition, persistence diagrams generated from persistent homology are robust to noise (Cohen-Steiner et al., 2005). This means metrics computed from persistence diagrams are also generally robust to noise and are a suitable tool for sensitivity analysis via input perturbation. Because persistence diagrams are stable under small perturbations of the underlying metric space, large Wasserstein deviations indicate structurally meaningful changes rather than numerical noise alone.

4 Problem Definition

4.1 RQ1: Topology of Adversarial Perturbations

In making small point perturbations to the input, we expect to see the appearance or disappearance of high-persistence 0-dimensional and 1-dimensional features rather than shifts in the overall distribution of persistence values. In other words, we expect text substitution to primarily influence the *tail behavior* of persistence values. We examine the tail behavior of persistence using a suite of graphical visualizations. To better evaluate the tail behavior of persistence values, we propose **persistence survival curves**. By plotting and overlaying the persistence percentiles on a logarithmic scale, we emphasize the difference in the weights of high persistence tails as a result of perturbation, rather than show the relatively diffuse and noisy behavior of the main mass of points.

4.2 RQ2: Layer Analysis with TDA

Different layers in an encoder model typically characterize different levels of the meaning of the input. Early layers typically encode local syntax features, while every subsequent layer encodes more global meaning. In this paradigm, the added classification head represents the highest-level meaning, i.e.,

the overall sentiment, of the input. With this in mind, we visualize the distribution of persistence values across layers 0, 6, and 11, representing early, middle, and late layers. Further, we investigate the sensitivity of the heads in each layer to adversarial perturbation.

4.3 RQ3: Generalizability of the Topology of the Embedding Space Across Models

We compare patterns across several encoder models to identify whether the observed topological behaviors and layer-wise sensitivities are consistent. This serves to validate results developed from individual model layer analysis.

5 Methodology

5.1 Datasets

The IMDb dataset consists of 50,000 long-form movie reviews (25,000 positive and 25,000 negative) collected from the IMDb website. Reviews contain an average of 231 words with standard deviation 171 words.

5.2 Selected Language Models

We evaluate the impact of TextFooler on three encoder-only models: BERT-base-uncased, RoBERTa-base, and DistilBERT. Each model was modified via a classification head consisting of a dropout and a linear layer during training, then fine-tuned to perform binary classification on the IMDb dataset. For consistency, we capped the number of input tokens to the first 128 tokens of each example. BERT and RoBERTa were trained to a level of greater than 90% test accuracy, and DistilBERT was trained to a level of greater than 80% test accuracy.

5.3 Adversarial Attack

Using TextFooler, Jin et al. (2020) propose a model-agnostic, score-based point substitution attack for text classification tasks. Suppose we have a binary classification task and have fine-tuned a language model f_θ to output the probability of class 1 on dataset $\mathcal{D} = \{(q_i, y_i)\}$, $y_i \in \{0, 1\}$. Given an input q_i with label y_i , a substitution attack perturbs the input by replacing a small number of words in q_i , resulting in a semantically similar input q'_i . The attack aims to achieve

$$|y_i - f_\theta(q'_i)| > 0.5,$$

an incorrect prediction, in as few substitutions as possible.

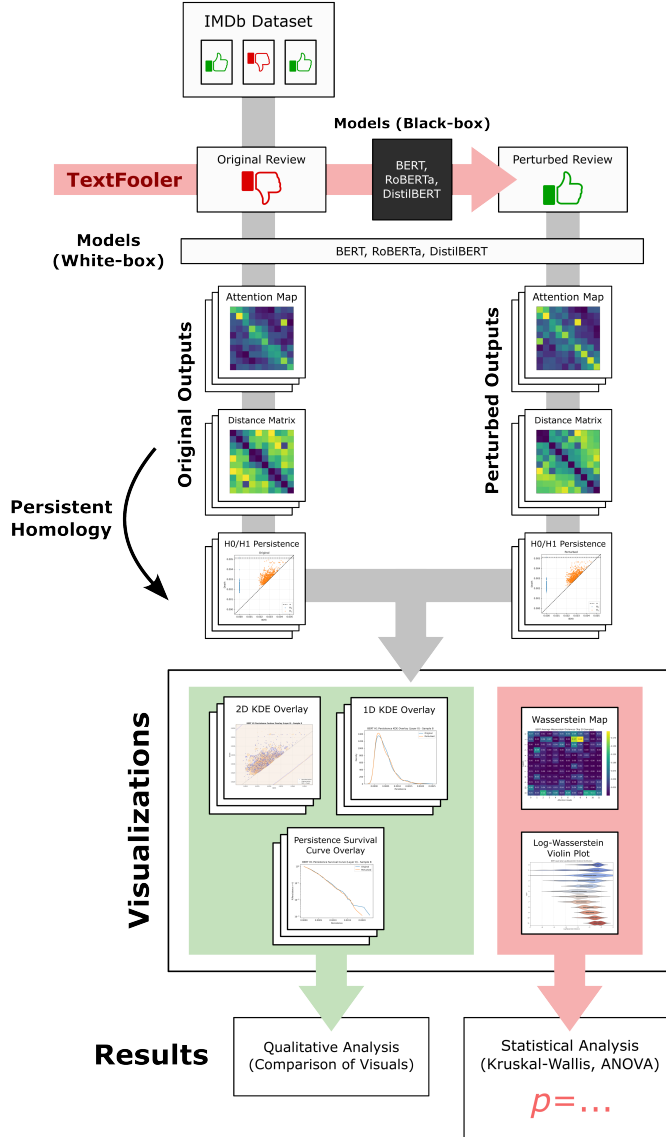


Figure 2: Overall pipeline for adversarial attack, visualization, and layer-wise analysis of attention maps. Recall that TextFooler is a black-box attack, but the subsequent step of extracting attention maps is white-box. Note that some outputs are “stacked,” meaning there are multiple diagrams, one per layer or head.

6 Experimental Setup

6.1 Attack Pipeline

Given a model where only output logits are visible, the TextFooler attack first determines the “importance” of each word in the input q_i by masking word individually and observing the change in the logits of the output. Then, the attack will perform a search by iteratively replacing important tokens with similar words with some minimum cosine similarity threshold — effectively, replacing words with synonyms. For each model, In our input generation step, we store the original input q_i and the altered input q'_i , and log the number of words that

have been changed for each input. For each model, we use the same set of 1024 examples from the IMDb test split to ensure direct comparability. Finally, the attack results were sorted by highest to lowest “effectiveness” of the attack, which is measured by the absolute change in the prediction logits after an attack. Next, for each model, we stored all input sentence pairs and attack statistics in tabular form for analysis:

- **Original and Perturbed Texts:** In both texts, the affected words were wrapped in the special punctuation $[[...]]$, which was also cleaned via regex prior to feeding into our models.
- **Original and Perturbed Scores:** *Score* mea-

sures the deviation of the logit corresponding to the correct label from a fully confident logit of 1.

- **Absolute Score Difference:** The loss in score after an attack. This is positive for all attack instances where the model initially was greater than 50% confident in the correct label but then switched its prediction.

An example original-perturbed input pair is provided below. The replaced words (in this case, just one) are in bold.

Original Input: This version is very **painful** to watch. All of the acting is very stilted. . . (*Negative, 99.97% confidence*)

Perturbed Input: This version is very **scathing** to watch. All of the acting is very stilted. . . (*Positive, 99.95% confidence*)

6.2 Attention Map Extraction

For each original-perturbed text pair, we performed inference by individually feeding each example into their corresponding models, and extracted attentions per head per layer. For BERT and RoBERTa, there were 12 layers and 12 attention heads per layer; for DistilBERT, there are 6 layers and 12 attention heads per layer. Thus, for each head, we have a $seq_len \times seq_len$ attention map (Vaswani et al., 2017).

We now formalize the sense of “distance” between a pair of tokens in the input sentence. For a sequence of tokens (t_1, t_2, \dots, t_k) , each attention map is a $k \times k$ matrix A that encodes pairwise attention weights a_{ij} , with each a_{ij} being nonnegative and each row summing to 1. As discussed previously, this matrix representation naturally encodes a sense of distance between tokens, where higher attention weights a_{ij}, a_{ji} between a pair of tokens corresponds to a lower distance between the tokens in space. To achieve this, we create a distance matrix D containing the pairwise distances between tokens, which we define to be 1 minus the average attention between the tokens (Kushnareva et al., 2021). In other words, for each pair of tokens (t_i, t_j) , we set

$$d_{ij} = 1 - \frac{1}{2} (a_{ij} + a_{ji}).$$

We also set $d_{ii} = 0$ for $i \in 1, 2, \dots, k$. Our resulting distance matrix is then used to generate

a Vietoris–Rips filtration and perform persistent homology.

6.3 Visualizations

In order to highlight different changes in persistence upon perturbation, we introduce several visualization modes of persistence:

1. **Persistence Diagram:** a plot of birth-death values (b_i, d_i) for H_0 and H_1 topological features (Cohen-Steiner et al., 2005). This is the standard mode of visualization for persistent homology.
2. **2D KDE Plot:** a contour plot directly on the H_1 persistence diagram. This smooths the discrete distribution, reduces noise, and is more visually intuitive in showing systematic trends in the overall distribution than comparing point clouds (Rosenblatt, 1956).
3. **1D KDE Plot:** a plot of H_1 persistence values as KDE plots, showing the distribution of mass in persistence values (Rosenblatt, 1956).
4. **Persistence Survival Curve:** a plot of the survival function versus the persistence, where the survival function is plotted on a logarithmic scale.
5. **Wasserstein Heatmap:** a heatmap of average Wasserstein metrics between original and perturbed H_1 persistence diagrams, showing variation across heads and layers (Scholkemper et al., 2024).
6. **Wasserstein Violin Plot:** a violin plot of the distribution of log-Wasserstein metrics compiled across examples per layer.

6.4 Statistical Tests

To assess which layers in an encoder model are more sensitive to adversarial perturbations, we compute the Wasserstein distance (Rüschendorf, 1985) between H_1 persistence diagrams for each attention head across the 20 sentence pairs corresponding to attacks that maximized output-logit changes. These distances were aggregated per head and per layer. As a non-topological baseline, we additionally computed raw L_1 distances between corresponding original and perturbed attention maps for each head and layer. These L_1 distances were aggregated identically to the Wasserstein metrics, allowing direct comparison between TDA-based and conventional attention-space sensitivity measurements. Unlike raw L_1 distances, which measure elementwise deviations in attention weights, persistent homology summarizes higher-order con-

nectivity structure across multiple scales. Thus, TDA-based Wasserstein distances are more sensitive to global structural changes induced by perturbations even when local attention differences remain small.

Since we did not assume any particular distribution of Wasserstein or L_1 distances across heads and examples, we first applied non-parametric tests. Specifically, we used a rank-based Kruskal–Wallis test (McKight and Najab, 2010) on Wasserstein distances and L_1 distances to evaluate whether layers differed in their distance metrics, followed by a post-hoc Dunn test with Holm correction for pairwise layer comparisons. These tests are robust to skewed distributions and do not require normality assumptions (McKight and Najab, 2010). In addition, the rank-based non-parametric statistics are unaffected by the logarithmic transformation because logarithms preserve ordering.

In parallel, we also performed parametric tests. Based on the log-Wasserstein violin plots (Hintze and Nelson, 1998), the log transformation greatly reduced right skew and produced distributions within each layer that approximated normality. This allowed us to perform a parametric one-way ANOVA (Quirk, 2021) to assess differences in mean log-Wasserstein distances between layers. By using both parametric and non-parametric tests, we maximize the explanatory power of our observations: the Kruskal–Wallis test provides a robust baseline that makes no assumptions about normality, while ANOVA offers greater power by leveraging the improved symmetry of the log-transformed data.

7 Results

7.1 Topology of Adversarially Perturbed Embedding Space

In our analysis of adversarial topologies, we observed that the vast majority of points in the H_1 persistence diagrams were concentrated in large clusters of points, making the original and perturbed diagrams difficult to differentiate via basic inspection or overall population statistics. However, in many of these diagrams, there were a few outliers with high birth and death values and persistence values (i.e., points up the diagonal and points farther away from the diagonal, respectively). For this type of deviation, we observe a marked improvement in qualitative visual analysis by using logarithm-scale survival curves over histograms,

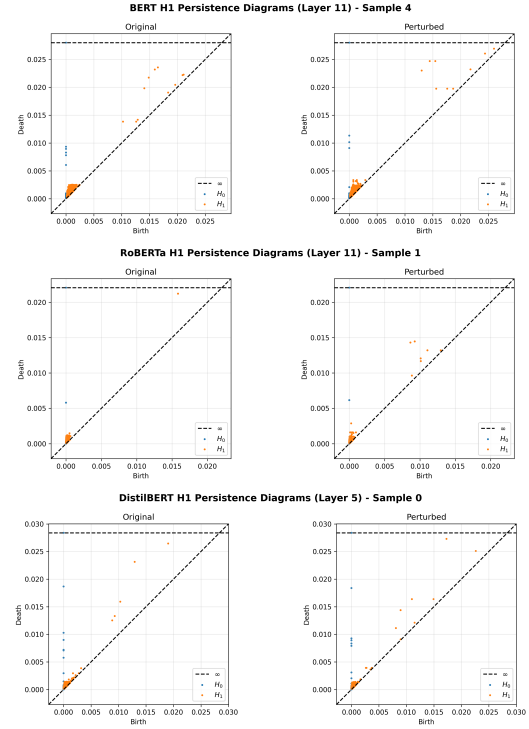


Figure 3: Persistence diagrams for a text pair across the three encoder models where “worst” was replaced with “gravest,” averaging attentions across heads of the final layer prior to computing persistence. Low-death, low-persistence points tend to cluster in the bottom left and are less important than the scattered high-persistence points toward the top of each plot.

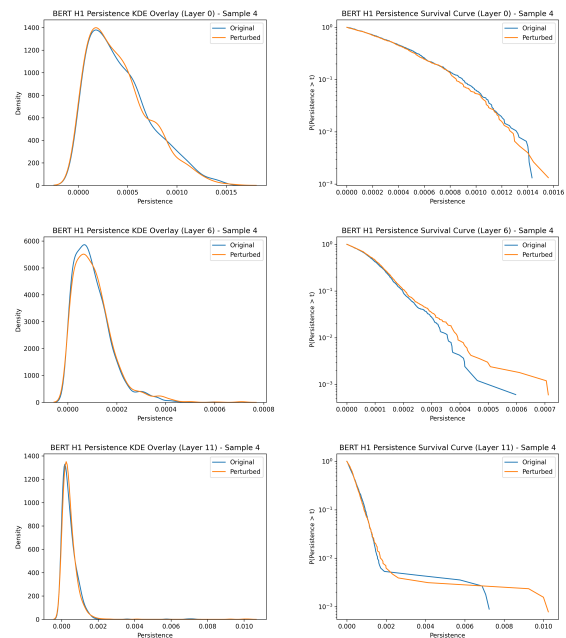


Figure 4: BERT KDE overlays (left) and survival curves (right) for layers 0, 6, and 11 for the substitution attack in Figure 3. The difference in mass in high-persistence regions before and after perturbation is more clearly visible in the survival curves.

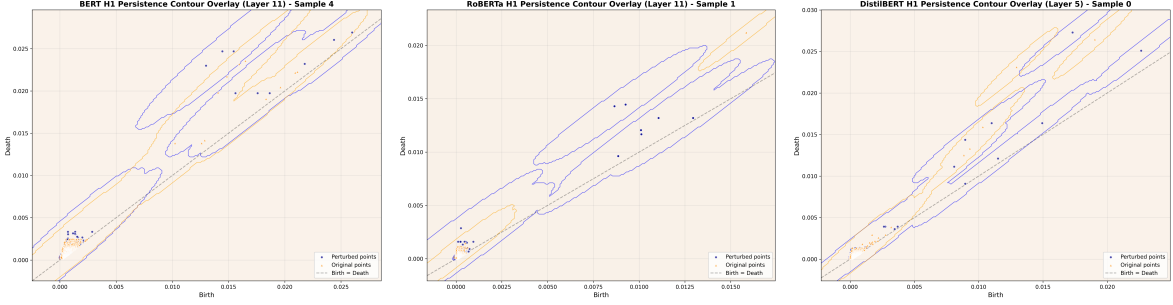


Figure 5: 2D KDE plots on the H_1 persistence diagrams in Figure 3. We see that the majority of the mass of persistence features in the H_1 diagram remain stationary, but there are shifts in high-persistence outlier points.

KDE overlays, and simple persistence diagrams. Survival curves more clearly show deviations in tail behavior compared to the raw persistence diagrams and KDEs (Figures 3, 4, and 5). Additional comparisons can be found in Appendix C.

Model	Method	H	p_{KW}	Dunn
BERT	TDA	902	2.3×10^{-186}	11/11
	Baseline	422	1.1×10^{-83}	10/11
RoBERTa	TDA	753	1.9×10^{-154}	10/11
	Baseline	430	2.3×10^{-85}	5/11
DistilBERT	TDA	494	1.7×10^{-104}	5/5
	Baseline	58.8	2.2×10^{-11}	5/5

Table 1: Non-parametric analysis of layer-wise differences for both TDA-based and baseline non-TDA distances. Entries report Kruskal–Wallis statistics and corresponding p -values. “Dunn” reports the number of significant pairwise differences between the final layer and all preceding layers under Dunn tests with Holm correction. TDA-based distances consistently produce stronger statistical separation across layers than baseline L_1 distances.

Model	Method	F	p_{ANOVA}	Tukey
BERT	TDA	51.9	7.6×10^{-105}	11/11
	Baseline	37.3	1.3×10^{-75}	10/11
RoBERTa	TDA	32.9	1.5×10^{-66}	9/11
	Baseline	44.7	1.3×10^{-90}	5/11
DistilBERT	TDA	50.2	5.2×10^{-48}	5/5
	Baseline	13.6	5.1×10^{-13}	5/5

Table 2: Parametric analysis of layer-wise differences using one-way ANOVA and Tukey’s HSD tests for both TDA-based and baseline non-TDA distances. Entries report ANOVA statistics and corresponding p -values. “Tukey” reports the number of significant pairwise differences between the final layer and all preceding layers under Tukey’s HSD tests. While both approaches detect significant layer-wise effects, TDA-based distances generally yield stronger and more discriminative layer separation, with the exception of RoBERTa.

7.2 Layer Sensitivity Analysis with TDA

We observe from the persistence survival curves (Figures 4; and in Appendix, 8, 9) that persistence diagrams (Figures 3) show more drastic differences in tail behavior as the layer number increases from $0 \rightarrow 6 \rightarrow 11$. The Wasserstein distances per head point to a similar phenomenon. The distance matrices in Figure 6 are average Wasserstein distances per head across the 20 examples with the highest deviation in output logits. These matrices show a substantial increase in Wasserstein distances in the final two layers for each model. In addition, some heads in layers 0, 1, and 2 also show large average Wasserstein distances as a result of perturbation. Unlike in late layers, this behavior is typically only shown in a small number of heads rather than across all heads (Figure 6).

We also plotted the distribution of log-Wasserstein distances per layer, aggregating the $12 \times 20 = 240$ observations per head per example for each layer. Even in log-space, there is a clear trend in log-Wasserstein distances, with median distances per layer roughly following a sideways “U” (or “W,” in the case of RoBERTa) shape for each model (Figure 7). To test this hypothesis, we use both non-parametric and parametric tests, as outlined in Section 5. For BERT, the Kruskal–Wallis test suggests very strong differences between layers ($H = 902, p = 2.3 \times 10^{-186}$; see Table 1), and post-hoc Dunn tests show that the final layer differs significantly from all other layers. The baseline L_1 distances for the same set of examples show a much weaker trend statistically and visually, with distances weakly increasing over layers (Figure 12). While the majority of layer comparisons still yield significant results, the separation of effect sizes across layers is observed to be much stronger in the TDA analysis. Similar results are observed for RoBERTa and DistilBERT, although on a weaker

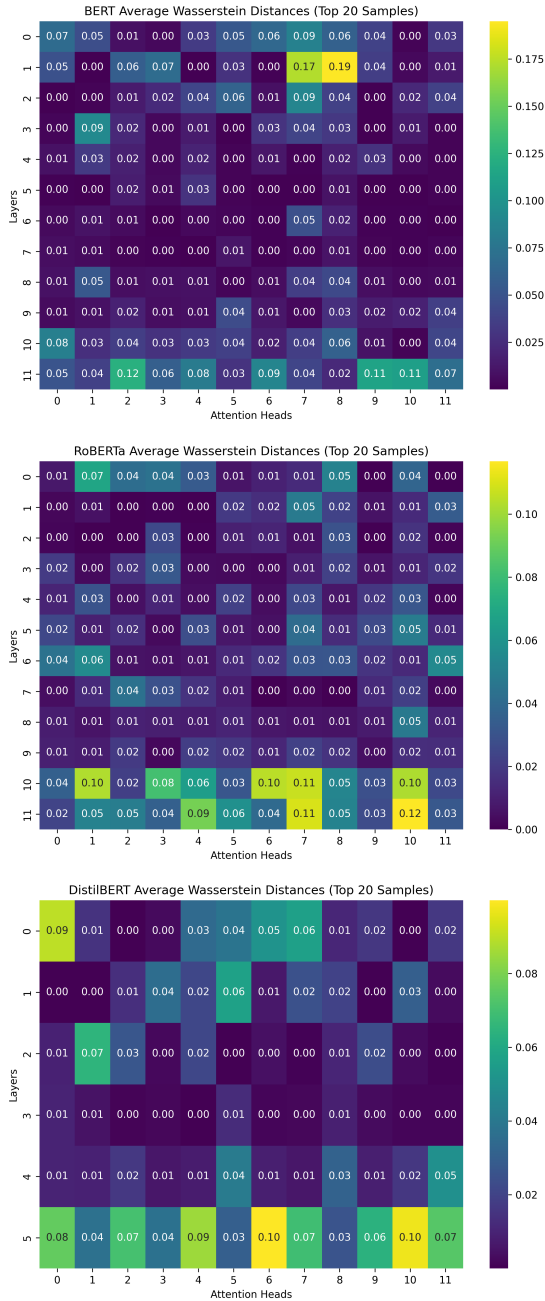


Figure 6: Wasserstein distance matrices showing distance values per head across the 20 examples with the highest deviation in logits. Notice the consistent high activations in the latest and earliest layers.

scale (Table 1).

Parametric analysis on the log-transformed Wasserstein distances using one-way ANOVA corroborates these findings, with the final layer differing significantly in mean log-Wasserstein distance across layers for all models (Table 2). Tukey’s HSD post-hoc tests confirm the non-parametric results, though the discrepancy between RoBERTa’s middle layers and late layers is less pronounced. A comprehensive table of pairwise p-values from

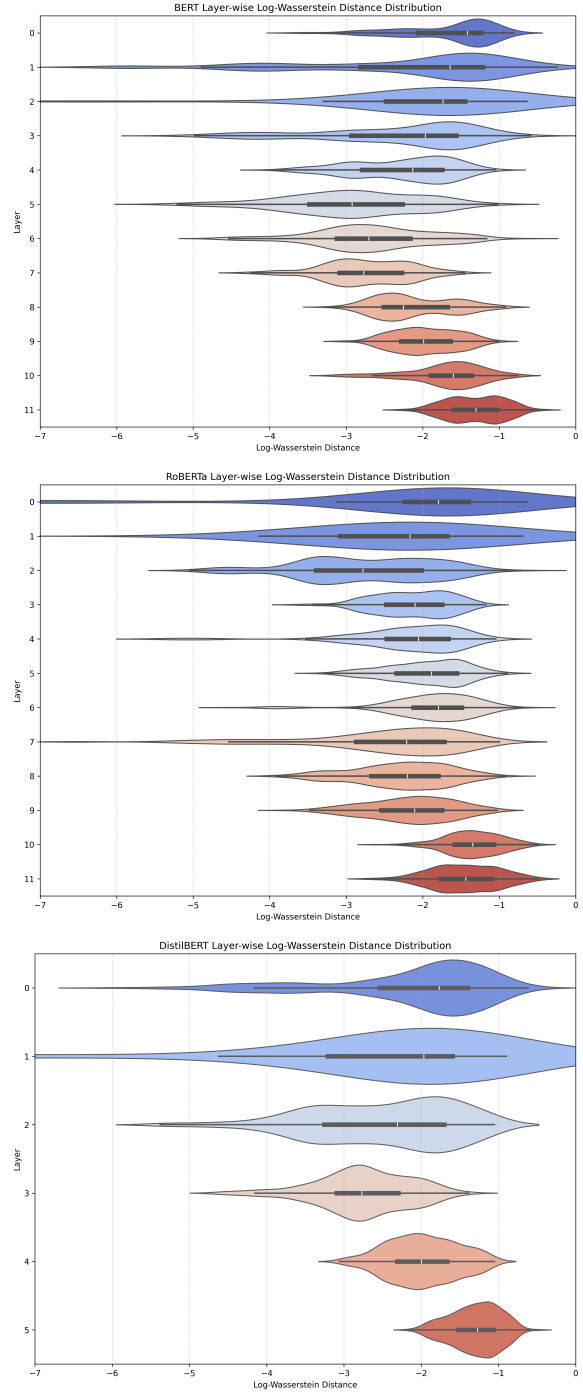


Figure 7: Per-layer log-Wasserstein distance violin plots with box plot overlay.

each test is provided in Appendix D.

7.3 Generalization Across Models

Our violin plots show that, across all models, the very early and very late layers show the highest distances, and the middle layers show the lowest distances. Our statistical tests confirm this claim. For all models, the final layer exhibits significantly higher log-Wasserstein distances than nearly all

preceding layers, reflecting increased sensitivity to perturbations at the top of the network. However, RoBERTa differs from the other models in a notable way: its middle-layer attention maps show a pronounced increase in distances, suggesting a resurgence in sensitivity in middle layers that is largely absent in BERT and DistilBERT. In general, RoBERTa’s average log distances are more evenly distributed across layers, although the final layer still maintains significantly higher distances than earlier layers. This pattern highlights the common trend of greater sensitivity in later layers, and a model-specific feature of RoBERTa, where perturbation effects are more distributed across the network.

8 Discussion

Adversarial Influence is Most Detectable in Early and Late Layers. The qualitative and quantitative findings in Section 7 can be explained in the context of model evolution over layers. At each layer, our encoder models consistently incorporate additional context that accumulate across each layer, resulting in the highest deviation in attention shape just before the decision layer. The large Wasserstein distances in a subset of heads in early layers is also intuitive: since low-level representations are close to individual token embeddings, single-word perturbations have a large impact on early layers’ attention topology.

RoBERTa Exhibits Relatively Uniform Sensitivities Across Layers. This pattern is apparent in both baseline and TDA methods, showing fewer significant pairwise layer comparisons than other models in its family. This can be attributed to several properties of RoBERTa’s design and pre-training. First, RoBERTa has more rigorous pre-training than BERT (Liu et al., 2019), using more data and more dynamic masking. These features prevent the model from overfitting on data and concentrating specific features in a few layers or heads. Second, RoBERTa was pretrained specifically for masked language modeling (MLM) (Liu et al., 2019), whereas BERT was pretrained for both MLM and next sentence prediction (NSP) (Devlin et al., 2019). MLM is much more relevant to our binary text classification task than NSP since sentence coherence does not produce contextual features. Therefore, since all heads are dedicated to learning contextual features, there is less specialization of attention heads, meaning heads are less

sensitive to word-level perturbations.

TDA Captures Structural Rather Than Purely Local Changes. Although both the TDA-based Wasserstein distances and the non-TDA L_1 baseline detect statistically significant effects of adversarial perturbations, the TDA metrics consistently produce stronger and more discriminative layer-wise separation. In particular, the TDA-based analyses reveal clear U-shaped or W-shaped sensitivity trends across layers, whereas the baseline distances show weaker and less consistent patterns across different models (Figures 7 and 12; Tables 1 and 2). Because persistent homology summarizes connectivity structure across multiple scales, the resulting Wasserstein distances are more sensitive to these large-scale structural perturbations than raw matrix norms alone.

Practical Applications. Beyond characterizing layer-wise behavior, Wasserstein sensitivity analysis has practical use cases. For example, it can be used to better inform future studies of the weakest parts of an LLM. When implementing safeguards to prevent model overconfidence, developers of an LLM may reinforce residual connections from less sensitive layers to more sensitive layers to mitigate the overall impact of single-word substitutions. In addition, for text classification tasks using LLMs, Wasserstein sensitivity analysis can inform model confidence by aggregating raw logits and Wasserstein metrics as a result of adversarial perturbation. Given an input, if the average Wasserstein distance as a result of input word substitution is higher than usual, then the model may be over-relying on individual words in the input. Thus, it should decrease its confidence in its prediction for that example.

9 Conclusion

Our qualitative analysis reveals that the topology of the attention maps of each encoder model in this study changes on the global level as a result of adversarial influence, while local features stay roughly intact. Compared to conventional attention-space baseline distances, the TDA-based Wasserstein metrics produce clearer layer-wise separation and more discriminative sensitivity patterns across models. This is bolstered by our quantitative sensitivity analysis, which indicates that very early and late layers experience the greatest change in high-persistence features and show significantly higher Wasserstein distances than middle layers.

10 Limitations

One key limitation of our sensitivity analysis is that it is not directional: the Wasserstein metric only captures the magnitude of change between persistence diagrams, but does not indicate the direction or nature of the change. Consequently, while we can quantify how much the topological structure changes under perturbation, we cannot directly infer whether the change increases or decreases specific topological features. When examining directional trends in persistence, we did not find consistent patterns across examples, highlighting that magnitude alone may not capture the full dynamics of model sensitivity.

In addition, our per-layer Wasserstein analysis relies on a small subset of the data. Although we found statistically significant results, we only tested the 20 examples with the highest change in output logits. Although upon manual inspection, the vast majority of these examples preserved the semantics, the top 20 examples may not be representative of the average successful attack. Since Wasserstein distances are linear in the scale of the distance map, it is possible that some examples with “large” initial perturbations would have disproportionately large Wasserstein distances, which would disrupt parametric statistical tests, which rely on normality assumptions. To mitigate this effect, we suggest three potential solutions:

- **Larger Sample Size:** We can increase the number of samples to include in our analysis. This would allow for more robust and powerful statistical tests.
- **Random Sampling with Bootstrapping:** To get more representative distributions and 95% confidence intervals for mean log-Wasserstein distances, we could use bootstrapping in conjunction with random sampling of text examples.
- **Distance Matrix Normalization:** Prior to computing persistence, we could normalize the distances to decrease the chance of single examples that dominate the Wasserstein distance analysis. This could introduce bias in our Wasserstein distance calculations, especially if we do not scale distance matrices belonging to the same sentence pair identically.

We also acknowledge that Wasserstein distance calculations are generally slow; the fastest known algorithms for computing Wasserstein distance, such as the Hungarian algorithm, have time complexity scaling with f^3 , where f is the number of H_1 features, which itself can be up to quadratic in the number of tokens `seq_len`. This can become infeasible for larger inputs. In future studies, we propose using the bottleneck distance as a metric instead, as it is slightly faster computationally and isolates the effect of high-persistence values changing due to adversarial perturbation.

11 Ethical Statement

There are potential dangers to extending any conclusions found on these particular models and datasets to more general use cases. As the LLM ecosystem evolves, there are no guarantees that newer, state-of-the-art models will have the same vulnerabilities. In addition, any findings presented in this paper may not generalize to text classification tasks on other datasets.

All datasets and models, including pretrained model weights, are open-source and can be found on Hugging Face.

Acknowledgments

We acknowledge the Harvey Mudd College Computer Science and Math Clinic program and the MIT Lincoln Laboratory for facilitating this research project and providing computing power for our experiments. We also acknowledge Dr. Adaku Uchendu for her invaluable expertise and guidance throughout the project.

References

- Kenneth Alperin, Rohan Leekha, Adaku Uchendu, Trang Nguyen, Srilakshmi Medarametla, Carlos Levya Capote, Seth Aycock, and Charlie Dagli. 2025. [Masks and mimicry: Strategic obfuscation and impersonation attacks on authorship verification](#). In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 102–116, Albuquerque, USA. Association for Computational Linguistics.
- Erik Arakelyan, Zhaoqi Liu, and Isabelle Augenstein. 2024. Semantic sensitivities and inconsistent predictions: Measuring the fragility of nli models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 432–444.

- Jatin Chauhan and Manohar Kaul. 2022. Bertops: Studying bert representations under a topological lens. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, and 1 others. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World wide web*, 27(4):42.
- David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. 2005. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Herbert Edelsbrunner, John Harer, and 1 others. 2008. Persistent homology—a survey. *Contemporary mathematics*, 453(26):257–282.
- Aideen Fay, Inés García-Redondo, Qiquan Wang, Haim Dubossarsky, and Anthea Monod. 2026. [The shape of adversarial influence: Characterizing LLM latent spaces with persistent homology](#). In *The Fourteenth International Conference on Learning Representations*.
- Neil Fendley, Edward W. Staley, Joshua Carney, William Redman, Marie Chau, and Nathan Drenkow. 2025. [A systematic review of poisoning attacks against large language models](#). *Preprint*, arXiv:2506.06518.
- Jerry L. Hintze and Ray D. Nelson. 1998. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.
- Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075.
- Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial text detection via examining the topology of attention maps. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 635–649.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *Preprint*, arXiv:1907.11692.
- Patrick E McKight and Julius Najab. 2010. [Kruskal-wallis test](#). In *The Corsini Encyclopedia of Psychology*, pages 1–10. Wiley Online Library.
- Ilan Perez and Raphael Reinauer. 2022. The topological bert: Transforming attention into topology for natural language processing. *arXiv preprint arXiv:2206.15195*.
- Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4873–4883.
- Thomas J. Quirk. 2021. One-way analysis of variance (anova). In *Excel 2019 for Social Science Statistics: A Guide to Solving Practical Problems*, pages 167–184, Cham. Springer International Publishing.
- Murray Rosenblatt. 1956. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837.
- Ludger Rüschendorf. 1985. [The wasserstein distance and approximation theorems](#). *Probability Theory and Related Fields*, 70(1):117–129.
- Michael Scholkemper, Damin Kühn, Gerion Nabbefeld, Simon Musall, Björn Kampa, and Michael T. Schaub. 2024. [A wasserstein graph distance based on distributions of probabilistic node embeddings](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9751–9755.
- Donald R Sheehy. 2012. Linear-size approximations to the vietoris-rips filtration. In *Proceedings of the twenty-eighth annual symposium on Computational geometry*, pages 239–248.
- Adaku Uchendu and Thai Le. 2024. Unveiling topological structures from language: A survey of topological data analysis applications in nlp. *arXiv preprint arXiv:2411.10298*.
- Adaku Uchendu, Thai Le, and Dongwon Lee. 2024. Topformer: Topology-aware authorship attribution of deepfake texts with diverse writing styles. In *27th European Conference on Artificial Intelligence, ECAI 2024*, pages 1446–1454. IOS Press BV.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Minh Vu, Geigh Zollicoffer, Huy Mai, Ben Nebgen, Boian Alexandrov, and Manish Bhattarai. 2025. Topological signatures of adversaries in multimodal alignments. In *Forty-second International Conference on Machine Learning*.

Larry Wasserman. 2018. Topological data analysis. *Annual review of statistics and its application*, 5(2018):501–532.

Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Ming Hu, Jie Zhang, Yang Liu, Shiqing Ma, and Chao Shen. 2025. Jailguard: A universal detection framework for prompt-based attacks on llm systems. *ACM Trans. Softw. Eng. Methodol.*, 35(1).

.1 Code Availability

Our implementation for adversarial attack generation, attention map extraction, topological data analysis, and statistical evaluation is publicly available at: https://github.com/angelinatsai04/mitll_clinic/tree/adam_spring.

A Reasoning for Selection of Visualization Methods

Generally, the KDE plots are suited for showing trends in overall movement in the mean persistence of the distribution, that is, any consistent shifts in large masses of points. This would correspond to topological effects such as the destruction of topologies on small or large scales, or overall topological compression. The survival curve is generally more suitable for isolating global topological features that persist across scales.

The KDE plots and survival curves were plotted as overlays to facilitate direct visual comparison. For the persistence diagram, we ensured that plots of original and perturbed persistence were the same scale per example to also facilitate direct comparison.

Computing Wasserstein distances on H_1 persistence diagrams is preferred to computing norms on raw attention maps because persistent homology has guarantees about numerical stability and robustness to noise. In practice, we observed that the resulting Wasserstein distances were strongly right-skewed across heads and layers. To improve interpretability in both visualization and statistical analysis, we applied a logarithmic transformation to all Wasserstein distance observations.

B Top TextFooler Perturbations

Some of the successful attacks for each model that resulted in the highest decrease in correct prediction probability are reproduced in Table 3. Based on these samples, all of which are negative-turned-positive, negative reviews appear to be much more vulnerable to single-word attacks than positive reviews. The models seem to misclassify phrases containing words like “egregious” and “gravest,” which typically have negative connotations but can also be found in a positive review; these words are easily influenced by contextual cues. In addition, the review that appeared in every model’s top three examples was especially tricky; every model suffered from the same single substitution from “worst” to “gravest.” Given the remainder of the review, which is unchanged in the attack, it is clear that the review is negative, but this small change signals to the model that the series is mostly *hilarious* and not bad.

C Additional Visualizations for RoBERTa and DistilBERT

Figures 8 and 9 show the comparison between 1D KDE plots and their corresponding survival curves. DistilBERT has a less clear discrepancy between original and perturbed survival curves for this example. It is possible that, as a simpler model, DistilBERT only recognizes the word substitution on a local semantic level, explaining the early-layer differences, but this effect does not propagate as the context widens. We also computed persistences after averaging all heads in the layer, meaning any effect localized in few heads would be diluted in this visualization.

D Per-layer p-value Matrices

The heatmaps in Figures 10 and 11 show p-values for both parametric and non-parametric tests performed on log-Wasserstein distances, and are more detailed summaries of the results from Tables 1 and 2.

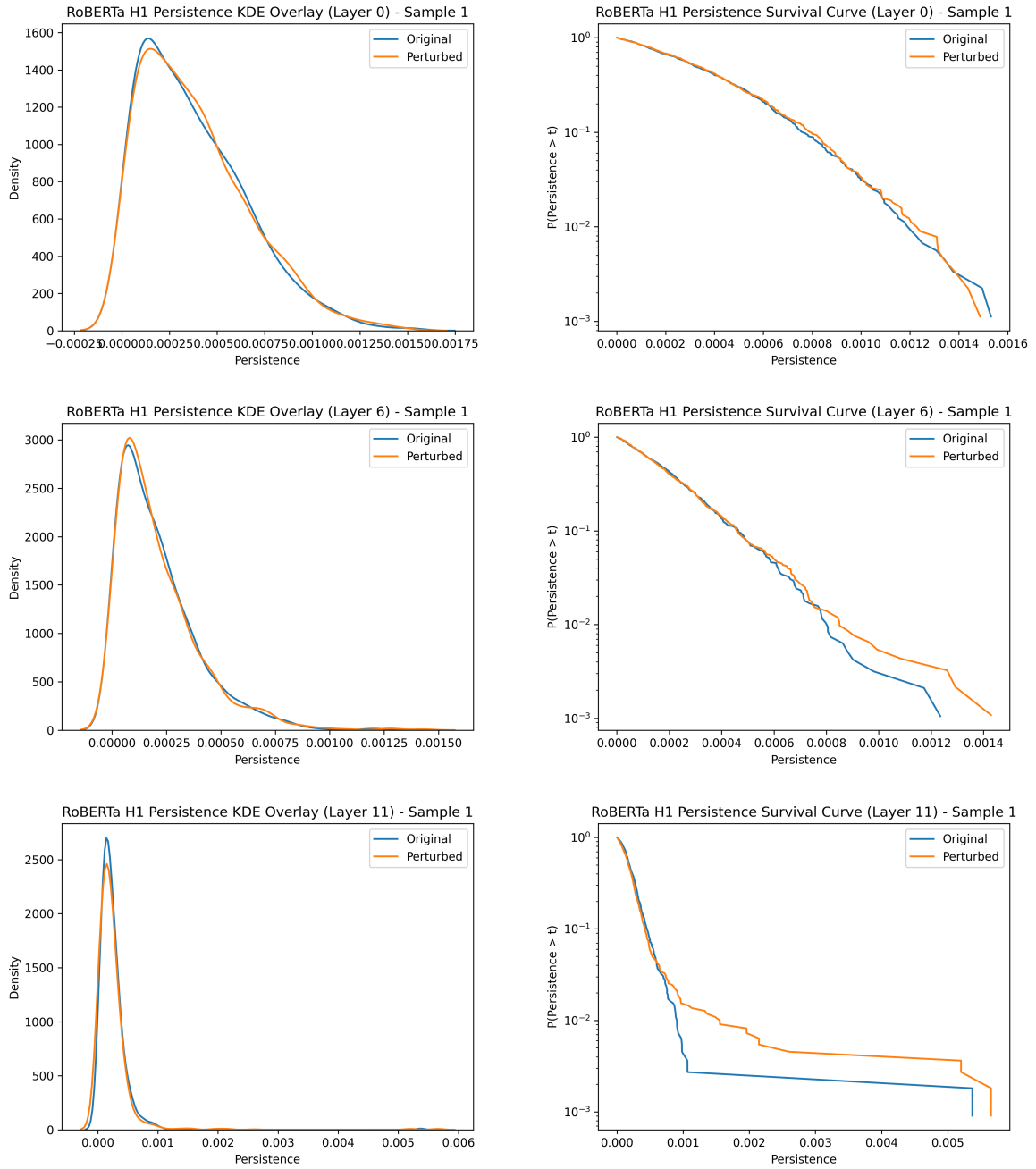


Figure 8: RoBERTa KDE overlays (left) and survival curves (right) for layers 0, 6, and 11 for the substitution attack in Figure 4.

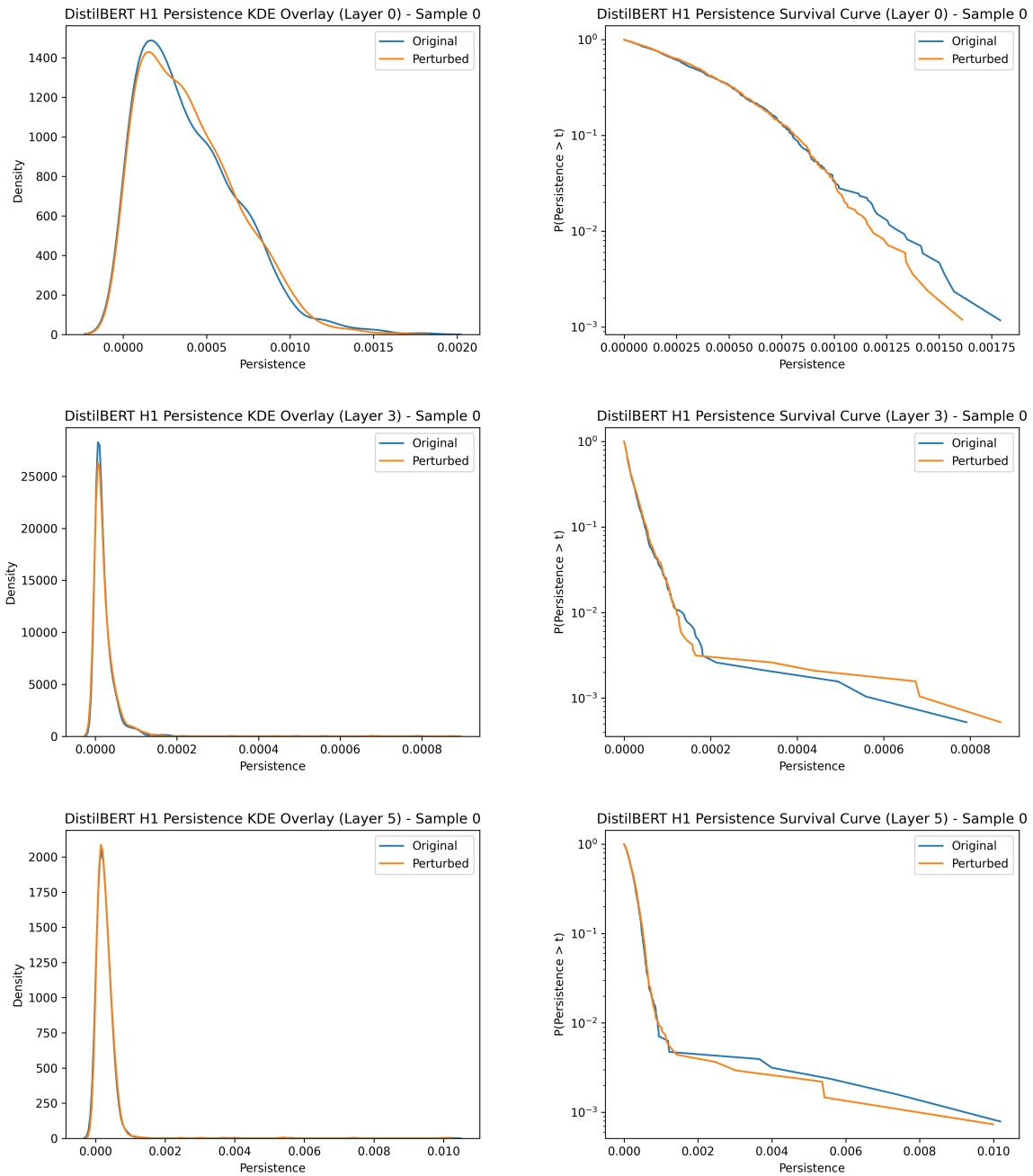


Figure 9: DistilBERT KDE overlays (left) and survival curves (right) for layers 0, 3, and 5 for the substitution attack in Figure 4.

BERT	
while Hillary Swank is great for the role, the plot to the movie is just dreadful . <i>Negative (99.94%)</i>	while Hillary Swank is great for the role, the plot to the movie is just egregious . <i>Positive (99.94%)</i>
The film is annoying . <i>Negative (99.90%)</i>	The film is troubling . <i>Positive (99.93%)</i>
[Y]ou'll agree with me that this is one of the worst and yet hilarious series ever made. <i>Negative (99.84%)</i>	[Y]ou'll agree with me that this is one of the gravest and yet hilarious series ever made. <i>Positive (99.94%)</i>
RoBERTa	
[Y]ou'll agree with me that this is one of the worst and yet hilarious series ever made. <i>Negative (99.27%)</i>	[Y]ou'll agree with me that this is one of the gravest and yet hilarious series ever made. <i>Positive (97.40%)</i>
This is a truly awful film. <i>Negative (99.83%)</i>	This is a truly spooky film. <i>Positive (96.51%)</i>
Don't see this movie! It's... repulsive! <i>Negative (99.74%)</i>	Don't presume this movie! It's... unsavory! <i>Positive (96.01%)</i>
DistilBERT	
[Y]ou'll agree with me that this is one of the worst and yet hilarious series ever made. <i>Negative (98.79%)</i>	[Y]ou'll agree with me that this is one of the gravest and yet hilarious series ever made. <i>Positive (98.61%)</i>
This is fairly typical for the Sci-Fi Channel: one-dimensional characters, a ridiculous plot, and terrible special effects . <i>Negative (99.40%)</i>	This is fairly typical for the Sci-Fi Channel: one-dimensional characters, a ridiculous plot, and tragic special impact . <i>Positive (96.90%)</i>
I really did not want to write a harsh review of this movie... However this movie is truly awful . <i>Negative (99.45%)</i>	I really did not want to write a harsh review of this movie... However this movie is truly egregious . <i>Positive (95.55%)</i>

Table 3: Adversarial examples showing original (left) and perturbed (right) text substitution pairs, producing prediction flips from negative to positive with high confidence across models. Replaced text is in bold.

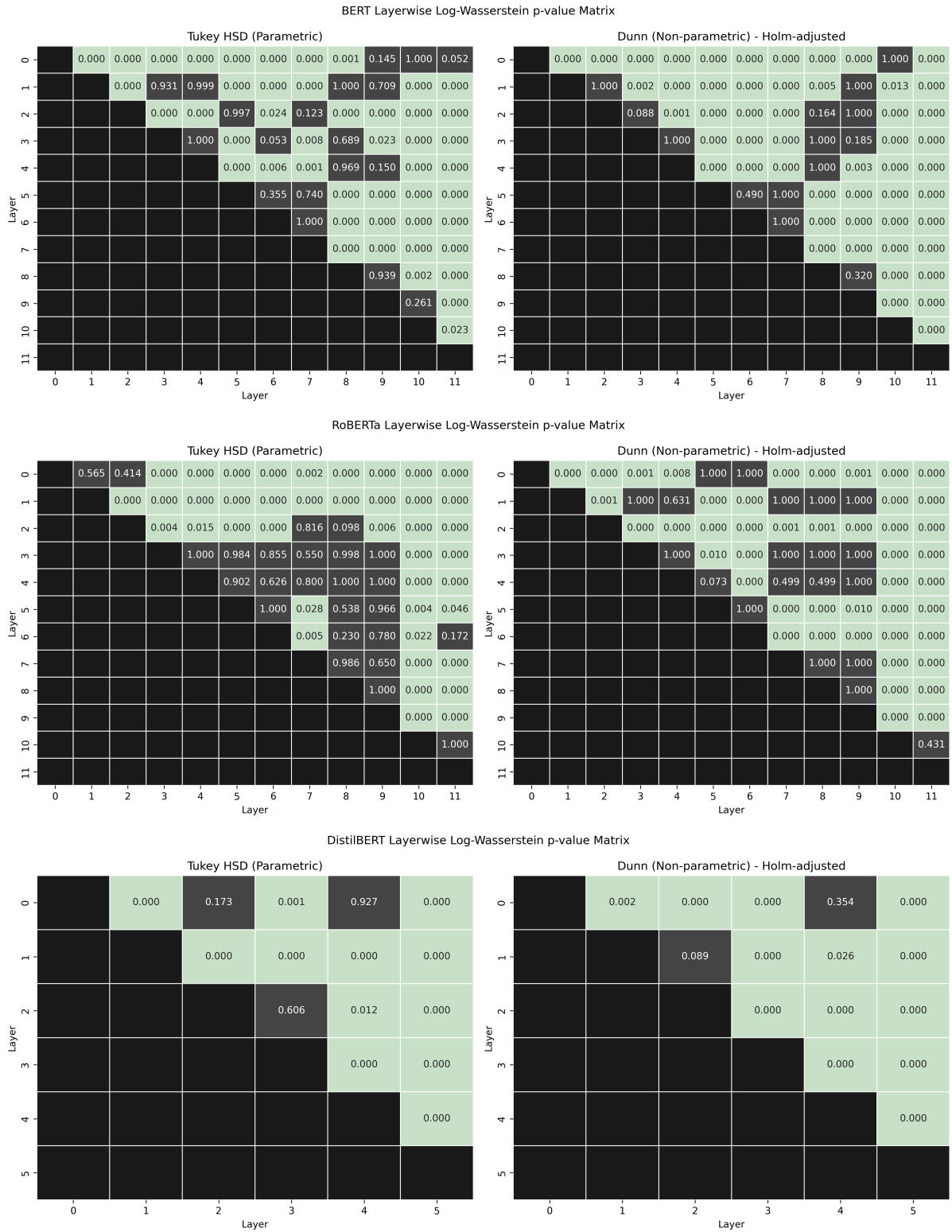


Figure 10: Layer-wise Dunn’s (left) and Tukey’s HSD (right) test p-value heatmaps of log-Wasserstein distances, with significant layer pairings in green. Note that the final layer in all models has significantly different log-Wasserstein distances from almost all other layers.

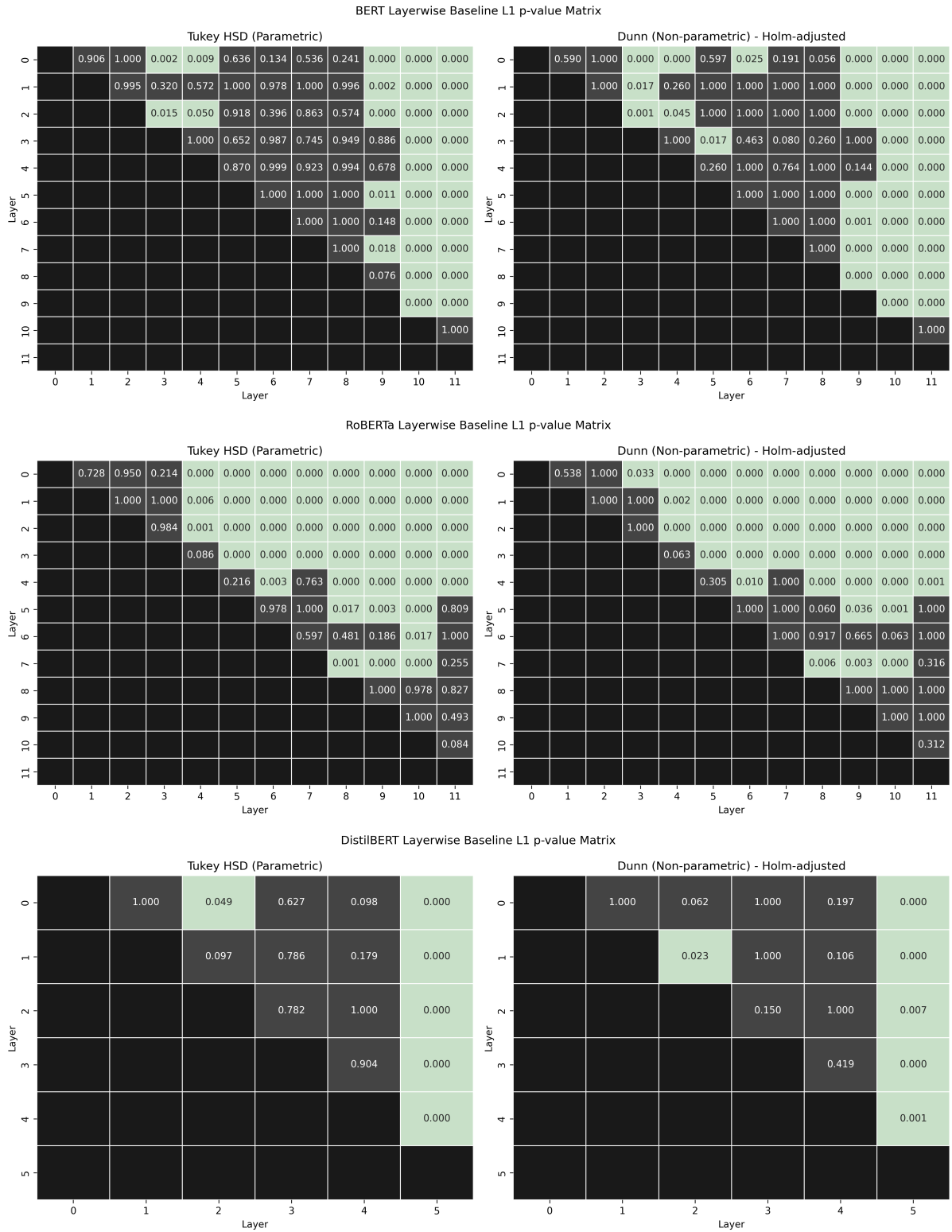


Figure 11: Layer-wise Dunn’s (left) and Tukey’s HSD (right) test p-value heatmaps of non-TDA baseline L_1 distances, with significant layer pairings in green. The U-shaped trend observed in our TDA results is less clear and varies much more between models.

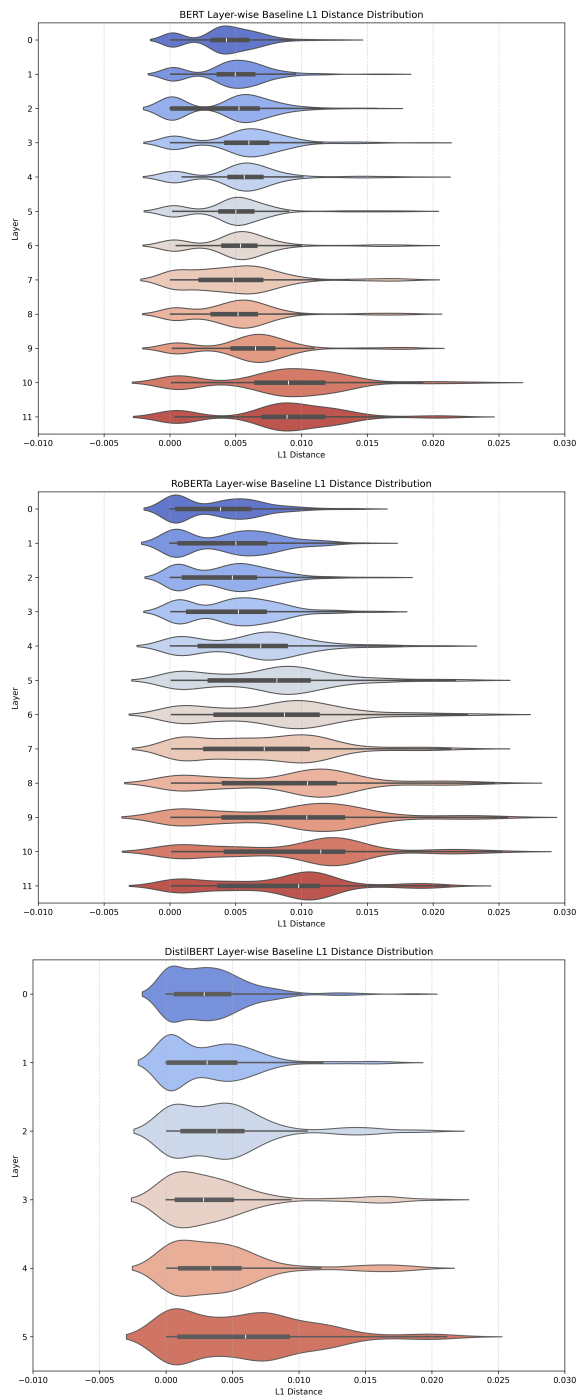


Figure 12: Per-layer Non-TDA L_1 baseline distance violin plots with box plot overlay.

Customizing ASR for Language Documentation and Resource Prioritization

Alexandra Fort and Shobhana Chelliah
Indiana University, Bloomington, IN USA

Abstract

Research in language documentation has the potential to benefit from integration of ASR models, especially through the assisted transcription of recordings with audio. Recent advancements in ASR for low-resource languages demonstrate the ability to adapt general, multilingual models for unseen languages with limited fine-tuning data, supporting the creation of custom ASR models. However, resources are still required to collect and prepare the fine-tuning data, necessitating exploration of optimization of resource allocation within the process of data collection and preparation. This paper outlines important considerations for the collection and preparation of data for customizing an ASR model for use in language documentation projects. With the development of a Lamkang ASR model as an example, prioritization of tasks within a language documentation project is outlined by analyzing the relative impact of time spent on transcription correction versus time spent on manual alignment on ASR model performance. Results from this research suggest prioritization of transcription correction over manual-alignment of data and suggest fine-tuning multilingual ASR systems produces superior results to zero-shot ASR models, despite recent advancements in the technology.

1 Introduction

In language documentation projects, the most valuable resource is commonly the time that the language experts, both linguists and speakers, spend refining the language data in a project. In the documentation process, a corpus of the language is developed to provide a foundation for research on the language and to guide the development of linguistic resources (Himmelmann et al., 2006). Audio recordings of speech acts, such as a conversation, interview, or narrative, are included in the corpus to provide acoustic examples of the language. These recordings are frequently accompanied by transcriptions that depict their contents.

The act of transcribing is considered the slowest part of the process of documenting a speech act, producing a phenomenon known as transcription bottleneck, which refers to the idea that the speed of transcription slows the rate at which a documentation project can progress, both due to the time it takes to transcribe and the amount of skilled transcribers available (Bird, 2020).

The idea of a transcription bottleneck is further complicated in language documentation projects where the orthography of the language is being established simultaneously. The process of establishing an orthography is time-intensive, but very important for the language community’s ability to interact with the project (Seifart et al., 2006). When orthography development and language documentation are happening concurrently, the initial transcription of a speech act is subject to change, requiring additional editing as decisions about the orthography progress. In such scenarios, the project may decide to either transcribe recordings phonetically, such as through the use of IPA, or through the use of the developing orthography.

There are various pros and cons to each approach. If using a phonetic representation, the transcriptions can be converted to whichever orthography is established after an orthography is established. However, this requires the transcriber to be trained to use a phonetic alphabet. Additionally, the transcribed recordings are likely to be inaccessible to most of the community until being converted to a representation that is being used in the community (Dobrin and Schwartz, 2021). Interpretability by the community is important both for avoiding extractive research practices and creating space for community language experts to contribute their invaluable knowledge to the project. This input during the transcription process is the most significant benefit of using the developing orthography instead of IPA. Further, use of the developing orthography supports familiarization and continued discus-

sion within the community about the orthography, which is required to further formalize the orthography. The primary limitation of this approach lies in the necessitation of updates, as transcripts using a developing orthography require updates. Moreover, if the developing orthography moves from a phonetically-condensed representation (such as using one character to represent multiple phonemes) to a more phonetically-specific representation, the ability to perform automatic conversion of the orthography is limited.

This specific study concerns a project that has chosen the method of using a developing orthography for transcription, detailing observations from the process of trying to develop the best-possible ASR model with the existing data. While low-resource is used to cover an expanse of situations, this particular project includes use of less than 3 hours of language data. The research questions addressed in this paper are:

1. Does time invested in alignment of transcripts or updating the orthography result have more impact on ASR model performance?
2. How does the relationship of data quantity and data quality impact training results in low-resource settings?
3. How do various models designed for use in low-resource settings compare to one another, both in performance and usability?

2 ASR and Language Documentation

ASR-aided transcription is often proposed as a method with the potential to address the transcription bottleneck. Examples of near-perfect performance of transcription tasks for high-resource languages motivate researchers to investigate how these tools could be adapted and applied in lower-resource settings. Previously, traditional ASR systems for low-resource languages made use of phoneme and lexicon dictionaries to produce language-specific systems, but recent research has proven the utility of end-to-end (E2E) systems for the task. Particularly, multilingually-trained E2E systems have been successfully fine-tuned on a small subset of labeled data in many low-resource settings (Olev and Alumae, 2022; Shi et al., 2021; Sadeque et al., 2022; Coto-Solano et al., 2022), both in scenarios where the initial E2E system includes the language in their training data and in scenarios where fine-tuning is the system’s first

exposure to the language. This approach requires less language-specific preparation of phoneme and lexicon dictionaries, lowering the barrier of preparation required to utilize ASR systems for unseen languages.

Limited research exists on the utility of the generated transcripts or the threshold of performance required by the ASR model to produce a useful transcript. Specifically, research is needed to determine at which point a transcript has too many errors, making error-correction a more difficult task than generating the transcript manually. This is a complicated topic of research as many factors are likely to impact the perceived difficulty of error correction, including transcriber preference, setting of the initial recording, and standardization of the operational orthography. For example, Prud’hommeaux et al. (2021) worked with the Seneca community and found that ASR-assisted transcription is able to produce transcripts in a shorter time period and with fewer errors, but that, regardless, many speakers prefer to do unassisted transcription. There is not a particular character error rate (CER) or word error rate (WER) that has been established as sufficient for aiding transcription, just a general assumption that the lower the rate is, the easier and faster the transcript will be to correct. However, it is also important to note that error-correction tasks come with their own challenges, as the distribution and types of error impact their detection (Søby et al., 2023; Point and Baruch, 2023).

Rendering an audio recording into a transcript necessitates decisions about what should be transcribed from the input. These decisions are of on-going interest for both high-resource and low-resource settings. For example, in high-resource settings, ASR systems have been shown to create accessibility issues for users with speech disfluencies, such as stuttering, by not being able to properly identify what users are saying (Mujtaba et al., 2024). Liao et al. (2023) suggest post-processing of ASR transcripts to reduce the influence of the transcription of disfluencies and speech errors on the readability of the final transcript. However, in language documentation scenarios where transcribers are simultaneously learning the components of speech that are relevant to the linguistic inventory of the language, reduction of such phenomena is complicated. Further, Ko and Burch (2025) emphasize the relevance and importance of maintaining speaker variation in transcription, as variation in transcription can change for both differ-

ent speakers and the same speaker on different days. They note various transcriptions can be produced that are valid representations of the audio, with different considerations for what is being annotated and stylistic decisions.

Results from languages without an established orthography demonstrate that ASR systems are able to produce a phonetically-consistent transcript for an unseen language (Chizzoni and Vietti, 2024). However, depending on the phonetic consistency of a language’s orthography, this default ability to produce phonetically-consistent transcripts may not contribute to improved results for a particular language, as not all phonetically apparent components are relevant to speakers. Bird (2020) builds on this issue, asserting that careful transcription of all components of a transcript may be misaligned with a community’s priorities and that sparse transcription could be more beneficial for some language documentation projects.

While fine-tuning E2E systems is the popular approach in most of the existing research, being able to use an ASR system without additional fine-tuning would increase access to the technology for many language documentation projects. While using off-the-shelf ASR systems for transcription when the target language is not in the training data has historically produced suboptimal results (Lin et al., 2025; Zahrer et al., 2020; Zheng et al., 2022), the recent Omnilingual model releases have shown improvement for zero-shot ASR (Omnilingual ASR Team et al., 2025), wherein the target language is missing from the training data and no fine-tuning is performed. Specifically, the zero-shot model, omniASR_LLM_7B_ZS, is able to use a multilingually trained ASR model plus an LLM to generate a reasonable ASR model for the language with just 1-10 recordings (of up to 30 seconds) provided for context.

However, Omnilingual ASR Team et al. (2025) do show that fine-tuning Wav2Vec2 models with even just 10 hours of language-specific data outperforms usage of the zero-shot model. Other projects have also seen promising results with less than 10 hours of data for fine-tuning an E2E, such as the use of 4 hours for Cook Island Maori (Coto-Solano et al., 2022) and the use of less than 99 minutes of single-speaker data for Limbu, Dotyal, Duoxo, Nahsta, Mwotlap, and Vatlong (Boulianne, 2022). This suggests that given even a very limited amount of transcribed data, fine-tuning an existing multilingual E2E model is likely to provide improved per-



Figure 1: Highlight of the Chandel district within the Manipuri state of India, the primary area where Lamkang is spoken (Commons, 2021)

formance over the zero-shot model. Nonetheless, integrating results from both approaches in this study allows for further discussion of strengths and weaknesses of zero-shot ASR versus fine-tuning multilingual E2E models.

3 Lamkang

3.1 Language Overview

Lamkang (iso: lmk) is a Tibeto-Burman language in the South Central (formerly known as Kuki-Chin) branch. It is spoken in Manipur, India by about 10,000 people, based on the results from India’s 1999 census (Eberhard et al., 2024). The language is primarily spoken in villages in the Chandel District, though recent conflict in the Manipur region has resulted in the relocation of speakers to nearby cities. The language is mentioned in the 1927 Linguistic Survey of India (Grierson, 1927), two grammatical sketches have been written for Lamkang (Thounaojam and Chelliah, 2007), as well as a handful of linguistic articles about reduplication, conjugation, and spatial terminology (Chelliah et al., 2020, 2019; Chelliah and Utt, 2017).

Two NSF grants (#1160640 and #0755471) from 2008-2016 were used to develop a corpus for the language. The corpus includes transcriptions of monologues, discussions, interviews, and elicitation narratives from 6 speakers (Sumshot Khullar, Rex Rengpu Khullar, Swamy Tholung Ksen, Daniel Tholung, Shekarnong Sankhil and Kumar Sankhil). The recording environment is partially controlled, but background noise and crosstalk are apparent. Transcriptions are accompanied by the

Updated	mthungbi ava thung thang va, mtii thang ngi, talu daat to boorkaang ne
Original	mbih ava' thung thang vah, mtii thang ngi, talu daat a boor kaang ne
Translation	And then there, beneath there, what was that, a basket, right?

Table 1: A side-by-side of a sentence in the updated orthography and the original orthography. Punctuation added for clarity in translation.

original recording audio, with a limited number being time-aligned in ELAN.

Lamkang uses a Latin-based orthography, but is still in the process of formalizing the orthography. [Chelliah et al. \(2023\)](#) details orthographical variation in Lamkang based on speaker writings collected during 12 years of language documentation work. Comparing this variation to orthographical decisions made in related languages and considering language-specific circumstances, suggestions are made for a series of orthographical decisions, such as segmentation and vowel length. Based on this analysis, Lamkang orthography uses 39 graphemes to represent vowels and consonants, as shown in the table found in the appendix 3. While Lamkang is a tonal language, currently the orthography does not mark tone. Analysis of the tonal system of Lamkang is ongoing in the documentation progress, so discussion of specifics related to Lamkang tone is left to future work.

3.2 Lamkang ASR Data

The current Lamkang project contains 38 transcribed files. The audio includes monologues, discussions, interviews, and elicitation narratives (such as Pear Stories). The files are primarily in Lamkang, though some of the interviews include questions posed in English. Additionally, instances of Lamkang-English code-switching, commonly singular words borrowed from English into Lamkang grammar structures, are found in some of the recordings. The files are between 12 seconds long and 20 minutes 26 seconds long with an average recording length of 5 minutes and 3 seconds. Of the 38 files, 19 have been aligned manually while another 19 are unaligned. The manually-aligned files have been aligned at phrasal levels in chunks varying in length from 1-12 seconds.

The initial transcriptions of the files were done

by linguists in conjunction with community members. However, the orthography has evolved since the initial transcriptions and the initial transcriptions require updates based on recently established orthographic conventions, primarily related to word boundaries, representations of vowel length, and inclusion of phonemes that have been reduplicated or inserted at word boundaries due to phonological rules. 8 of the 38 files have been updated to reflect the latest orthography. Compared to the original, if considering the latest orthography the gold-standard, there is a character error rate (CER) of .106 between the 8 updated transcripts and the 8 original transcripts. Variations in the initial transcriptions make automatic methods of updating the transcriptions to the existing standard unreliable, meaning that original transcriptions require review by a language expert or linguist. This leaves two primary tasks for data preparation of the files: transcription review and alignment.

4 Methodology

Considering the two data preparation tasks for the Lamkang data, transcription review (updating the orthography) and alignment, significant time is required to prepare the data for ASR. However, as time is a costly resource, this research methodology is meant to explore the most efficient allocation of time in this context, specifically whether updating transcriptions or manually aligning data provides a larger boost to the performance of a Lamkang ASR model, as well as how this relates to the amount of training data. Specifically, should researchers focus on correcting the original transcripts to the current standard, depending on tools for automatic alignment, or would it be better to correct and manually align a subset of the original transcripts? This question is intrinsically tied to the performance of auto-alignment methods, as improvements in automatic alignment methods should reduce the dependence of projects on manual alignment. Additionally, it contributes to broader discussions about data quantity versus quality.

With the goal of investigating the most productive path forward, 6 Lamkang ASR models are fine-tuned or tested, as shown in table 2. The first 5 models are fine-tuned versions of Wav2Vec2 ([Baeovski et al., 2020](#)) while the last model in the table represents the results of in-context learning (ICL) using the latest released Omnilingual ASR model, `omniASR_LLM_7B_ZS` ([Omnilingual ASR](#)

Team et al., 2025). Hyperparameters for fine-tuning can be found in Appendix B. The fine-tuned versions of Wav2Vec2 are the primary focus for this paper due to the consensus about improved performance reported in section 2, but the omniASR_LLM_7B_ZS Omnilingual model was released during this research and results are presented for comparison of performance and relative ease of use.

For naming shorthand, Man and Auto are used to refer to manual versus automatic alignment while C and O refer to use of corrected versus original transcripts. Com is shortened to indicate combined and Omni to indicate the Omnilingual model. Of the Wav2Vec2 fine-tuned models, the ManC model is trained on 7 recordings that have been manually reviewed by both a linguist and a native speaker and manually-aligned, representing prioritization of thoroughly cleaned and reviewed data at the expense of quantity. The ManO model includes the same 7 manually-aligned recordings, but retains the original transcriptions, showing the impact of updating the orthography. The AutoC model uses the 7 corrected transcripts, but replaces the manual alignment with automatic alignment, allowing for assessment of manual alignment versus automatic. Lastly, the AutoO model uses 37 of the original transcripts and automatic alignment with the goal of evaluating whether data quantity is more important than quality for ASR performance.

The ComC model uses the 7 corrected recordings, but uses both the manually-aligned and automatically-aligned transcripts to test the impact of a data augmentation method. This differs slightly from a simplistic duplication of the fine-tuning data by introducing the model to different prosodic chunks of the audio. For the OmniC model, 10 examples can be provided to the base model. For this set of 10, the 7 longest clips from each of the 7 corrected, manually-aligned recordings were selected, plus the 3 next longest clips when looking at all of the clips in aggregate. Documentation suggests use of clips of less than 30 seconds for the context examples, but when a set of clips that were about 20 seconds long on average were used for training, the transcript produced repetition loops. Providing a smaller subset of 5 recordings with the same average duration of 20 seconds long produced the same error, suggesting that using longer durations of audio for ICL of the Omnilingual model does not improve the results and may trigger a common LLM failure mode, such

as repetition loops. The average duration of the clips used in the reported model is 6.55 seconds, as this set of clips did not produce the repetition loop error.

The total length of the set of audio clips used for fine-tuning or ICL is reported in the third column of 2. Note that small discrepancies in the total time of the audio clips of the ManC, ManO, and AutoC models is a result of differing segmentations and pauses between words in those different segmentations. Code for data preparation, fine-tuning, and testing models is found at https://github.com/aconeil/lamkang_asr.

5 Data Preparation

For the AutoC and AutoO models, the first step of data preparation was to automatically-align the files using the accompanying text transcripts. An initial review of the text files, both corrected and original, included removing any speaker diarization marks, general punctuation cleanup, and references to indistinct noise or uncertainty¹. While many of these aspects are handled later on in the ASR fine-tuning pipeline, cleanup was required to compare the updated and original transcriptions and support correct segmentation for automatic alignment. Manual correction was used in place of automatic cleanup of the text files to protect against accidental deletion of transcript-relevant phenomena. Automatic alignment was done by resampling the audio to 16000Hz and using the MMS forced alignment model (MMS_FA) (Pratap et al., 2024) with PyTorch. This produced word-level alignment for the transcripts, with white-space in the transcripts delimiting words. The alignments were exported to ELAN files using the `speech` Python library.

For all Wav2Vec2 fine-tuned models, the ELAN files were split by annotation segments to produce wav files for training the model. An accompanying csv file was produced for the split annotations that maps the annotations to each wav file. The automatic word-level alignments are too short for fine-tuning ASR models, so the words are grouped into 7-gram long segments with remainder words in a transcript being appended to the final recording clip of the file. This methodology means that the manually-aligned segmentations are more likely to follow human-intuition and account for suprasegmental features, such as prosody and tone, while

¹Sequences were marked variably in transcripts and referred to different phenomena (i.e. mumbling, fast speech, trailing speech)

the automatically-aligned segmentations are split without regard to these features. Selection of the clips for the OmniC model follows the description provided in 4 section, with clips coming from the processing and segmentation of the corrected, manually-aligned files described above.

6 Results

The evaluation of these models uses character error rate (CER) and word error rate (WER). Since CER represents the normalized edit distance between two strings by calculating the insertions, deletions, or substitutions needed to turn one string into another and the next step of using an automatic transcription tool is to have a proficient speaker or linguist review and edit the output, CER can be used to approximate the edits required by the reviewer. WER is also provided, as updates to tokenization in Lamkang are part of the ongoing process of standardizing the orthography and the cognitive effort of identifying word breaks is likely to impact the speed of transcript correction.

In order to compare the performance of the models to one another, the test set is the same for all of the models. Additionally, since the goal in a documentation project would be to conform to the most updated orthography, the test set comes from one of the corrected transcripts. The test set consists of one recording with updated orthography, coming from a recording that is 1 minute and 47 seconds long. Accordingly, all versions of the transcription of this file and audio are removed from fine-tuning sets, including the transcription of the audio in the older version of the orthography.

The relative performance of these models suggests that updating the orthography is more beneficial to model performance than manual alignment, at least when the updates to the orthography are around a 10% CER, as is the case with this dataset. Using original orthography and manual alignment resulted in a model with a WER and CER that were respectively 11.8% and 3.7% worse, while using updated orthography and automatic alignment produced a model with a WER and CER that were 1.7% and 3.5% worse. Though the updated orthography has about a 10% CER when compared to the original orthography, the CER between the AutoC and ManO models is similar. However, there is a more than 10% improvement in WER in the AutoC model, demonstrating the utility of using the updated orthography to facilitate correct tokenization.

The best-performing model, the ComC model, achieves the lowest CER and WER, showing that automatic alignment can be leveraged for model improvement when used as an augmentation method. Compared to just using the manually-aligned and corrected recordings, the addition of automatically-aligned and corrected recordings gave a WER improvement of 8.7% and a CER improvement of 1.9%. Though the significant decrease between performance in the ManC and AutoC models shows that using automatic alignment in place of manual alignment decreases performance, it is beneficial when used in conjunction with the manually-aligned data.

In this research, the automatic alignment model consisting of all of the original transcripts (AutoO) is unable to properly fine-tune, regardless of adjustments to hyperparameters. In the best configuration of hyperparameters, very early stopping (at step 500) could result in a AutoO model with a CER of around .80, suggesting an underlying issue with the data itself in that the addition of data is actually hurting the model's performance almost immediately. Through error analysis and manual review of the automatically-aligned ELAN files, a significant issue that arose with automatic alignment was attribution of non-speech sounds to words in the transcript. This was especially problematic for recordings that began with or included a longer period of background noise. Trimming the beginning of such audio files and performing noise reduction would likely result in improved alignment, but removing these elements from the recordings would also diminish the ability of the model to appropriately handle such occurrences in recordings. Generally, more review of the transcripts would be required to make them useful for automatic alignment, though it is not clear that these steps would fully address this issue. Further, it is not clear manual cleaning of the recordings would be significantly faster than manual alignment.

Lastly, the OmniC model has a reasonably low CER at .268, but a high WER at .815. As the model required minimal resources and efforts for ICL, the CER is impressive at .268. The high WER but low CER indicates that tokenization is a bottleneck for the omniASR_LLM_7B_ZS model, though phoneme identification performs relatively well. Though this project is in a low-resource setting, providing 13 minutes of fine-tuning data to Wav2Vec2 is still able to produce superior results to the omniASR_LLM_7B_ZS model, results that mirror

Name	Description	Duration	WER	CER
ManC	Manual-align, corrected	00:13:18	.575	.205
ManO	Manual-align, original	00:13:16	.693	.242
AutoC	Auto-aligned, corrected	00:14:36	.592	.240
AutoO	Auto-aligned, original	02:58:13	1	1
ComC	Both, corrected	00:27:54	.488	.186
OmniC	Manual-align, corrected	00:01:05	.815	.268

Table 2: Description of models trained and total duration of wav files used for training (HH:MM:SS), followed by their character error rate (CER) and word error rate (WER)

the findings of the model release paper (Omnilingual ASR Team et al., 2025).

7 Discussion

Regarding the discussion of data quality versus data quantity, the model results showcase how determining which to prioritize is situation dependent. The ManO model outperforming the ManC model provides an example of quantity eclipsing quality. However, as alignment issues arise with the AutoO model, we see that the quality of the data can be detrimental to other parts of the ASR pipeline. Automatic alignment presents some utility for data augmentation, especially when used with the corrected transcripts, as seen by the ComC model, but is not able to produce adequate results with the current state of the original data.

If the original transcription data is very clean, such as list elicitation in a well-controlled environment, it is possible automatic alignment would be more beneficial to the project. For continuing research on tone identification, establishing a pipeline that utilizes list elicitation in a highly controlled environment and using automatic alignment could allow for efficient processing and integration of tonal data and mitigate the issues seen with non-speech sounds in automatic alignment of the existing transcripts. As for the more naturalistic settings (interviews, story-telling, and discussion) found in these transcripts, automatic alignment will be unlikely to produce data that is ready for fine-tuning an ASR model due to idiosyncrasies of transcribers and small deviations between transcript and audio that naturally occur when transcribing longer audio files with more background noise. It is possible that automatic alignment at the word level could serve as a starting point for someone manually aligning the transcripts, with the caveat that adjusting the annotation boundaries and merging words may or may not be quicker for a transcriber than manually

selecting prosodic chunks.

Considering the process of fine-tuning the models, the Omnilingual model was much easier to use. It does still require programming, but the bar to entry is much lower in terms of data preparation and data availability. This approach would likely only be recommended in language documentation situations in which there is no transcribed data in the language, but a linguist and/or speaker are able to transcribe 10 short recordings to get the process started. At this point, there is the possibility that the transcripts produced by the Omnilingual could be used as a starting point for further transcription via error correction, though, as mentioned in section 2, additional research is needed to determine at which point an ASR-produced transcript has a CER that is sufficiently low enough for the process of error correction to be faster than starting from scratch. Further investigation of the utility and application of the Omnilingual model at the nascent phase of a language documentation project would provide further insight as to the model’s utility, but the model does not apply well for the needs of this project.

8 Conclusion

Based on the initial results of this research, this project would see a greater benefit from updating the orthography than manually aligning the transcripts. While both are important to improving the results of an ASR system, greater gains in WER and a comparable CER is seen when including updates to the orthography compared to using the original manually-aligned data. However, each project must consider the resources available for review. For example, manual-alignment of data requires less knowledge of the language and specialized attention than updating the orthography, so this approach may be a better option if those available for review have less time and language-

specific expertise.

When considering data quality and data quantity, we see the importance of the two fluctuate depending on the degree of automation involved in the pipeline. If depending on additional tools, such as automatic alignment, data quality has a heavier influence, but when considering quality in terms of orthographic updates, quantity was more influential. Noting the limitations of automatic alignment based on data quality, we see the potential to improve model performance when the method is used with cleaned, high-quality data, such as the corrected transcripts.

Lastly, in considering the ease of use and performance of the various models, the results support preceding research in finding that significant gains in performance are found when fine-tuning E2E ASR models with limited labeled data. Though the zero-shot model requires less effort for data preparation and a dramatically reduced quantity of data, the current results of the model are easily surpassed with about 13 minutes of annotated data. The omnilingual model is more likely to benefit the very beginning phases of a documentation project and aid in the process of building enough labeled language data to fine-tune a model. While it presents a impressive step forward in zero-shot ASR, more improvements, especially in tokenization, are required for it to surpass the performance of a fine-tuned, multilingually-trained E2E ASR model.

Limitations

This research presents findings that are specific to the development of an ASR model for the Lamkang language documentation project, so research results are influenced by the specific language and project context. Though these results inform research on low-resource languages, specifics of other languages and projects are necessary to gain a fuller understanding of challenges and strengths of current ASR models for low-resource languages. The data used in the project is very limited, as it comes from an actual language documentation project that is in progress. The research guides the language documentation project on the most efficient way to create more annotated data, which can then be used to increase the sample size and significance of results. However, the current amount of audio available does limit the statistical significance of the results.

Acknowledgments

This research would not be possible without the dedication, work, and insight into the Lamkang language provided by Setpu-One Silsi.

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

References

- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.
- Steven Bird. 2020. Sparse transcription. *Computational Linguistics*, 46(4):713–744.
- Gilles Boulianne. 2022. Phoneme transcription of endangered languages: an evaluation of recent asr architectures in the single speaker scenario. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2301–2308.
- Shobhana Chelliah, Evaline Blair, Melissa Robinson, Rex Khullar, and Sumshot Khular. 2020. Reduplication in lamkang: Form, function, feeling. *Expressive morphology in the languages of South Asia*, pages 167–186.
- Shobhana Chelliah, Rachel Garton, Sumshot Khular, and Rex Khullar. 2023. Orthography development for languages of the south central branch of tibeto-burman: Lessons from lamkang. *Himalayan Linguistics*, 22(1).
- Shobhana Chelliah, David Peterson, Tyler Utt, Evaline Blair, and Sumshot Khular. 2019. Lamkang verb conjugation. *Himalayan Linguistics*, 18(1).
- Shobhana Lakshmi Chelliah and Tyler P Utt. 2017. The syntax and semantics of spatial reference in lamkang verbs. *Himalayan Linguistics*, 16(1).
- Ilaria Chizzoni and Alessandro Vietti. 2024. Towards an asr system for documenting endangered languages: A preliminary study on sardinian. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 214–220.
- Wikimedia Commons. 2021. [Chandel in manipur \(india\)](#).
- Rolando Coto-Solano, Sally Akevai Nicholas, Samiha Datta, Victoria Quint, Piripi Wills, Emma Ngakuravaru Powell, Liam Koka’ua, Syed Tanveer, and Isaac Feldman. 2022. Development of automatic speech recognition for the documentation of cook islands māori. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3872–3882.

- Lise Dobrin and Saul Schwartz. 2021. The social lives of linguistic legacy materials. *Language Documentation and Description*, 21.
- David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2024. *Ethnologue: Languages of the World: Lamkang*, 27th edition. SIL International, Dallas, Texas.
- George Abraham Grierson. 1927. *Linguistic survey of India*. Office of the superintendent of government printing, India.
- Nikolaus P Himmelmann et al. 2006. Language documentation: What is it and what is it good for. *Essentials of language documentation*, 178(1).
- Edwin Ko and Jem Burch. 2025. Transcription as an iterative and interpretive practice: Documenting connected speech in apsáalooke (crow). *Language Documentation and Description*, 25(1).
- Junwei Liao, Sefik Eskimez, Liyang Lu, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng. 2023. Improving readability for automatic speech recognition transcription. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(5):1–23.
- Kaiying Kevin Lin, Hsiyu Chen, and Haopeng Zhang. 2025. Formosanbench: Benchmarking low-resource austronesian languages in the era of large language models. *arXiv preprint arXiv:2506.21563*.
- Dena Mujtaba, Nihar Mahapatra, Megan Arney, J Yaruss, Hope Gerlach-Houck, Caryn Herring, and Jia Bin. 2024. Lost in transcription: Identifying and quantifying the accuracy biases of automatic speech recognition systems against disfluent speech. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4795–4809.
- Aivo Olev and Tanel Alumae. 2022. Estonian speech recognition and transcription editing service. *Baltic Journal of Modern Computing*, 10(3):409–421.
- Omnilingual ASR Team, Gil Keren, Artyom Kozhevnikov, Yen Meng, Christophe Ropers, Matthew Setzler, Skyler Wang, Ife Adebara, Michael Auli, Can Balioglu, Kevin Chan, Chierh Cheng, Joe Chuang, Caley Droof, Mark Duppenthaler, Paul-Ambroise Duquenne, Alexander Erben, Cynthia Gao, Gabriel Mejia Gonzalez, Kehan Lyu, Sagar Miglani, Vineel Pratap, Kaushik Ram Sadagopan, Safiyyah Saleem, Arina Turkatenco, Albert Ventayol-Boada, Zheng-Xin Yong, Yu-An Chung, Jean Maillard, Rashel Moritz, Alexandre Mourachko, Mary Williamson, and Shireen Yates. 2025. *Omnilingual ASR: Open-source multilingual speech recognition for 1600+ languages*.
- Sébastien Point and Yehuda Baruch. 2023. (re) thinking transcription strategies: current challenges and future research directions. *Scandinavian Journal of Management*, 39(2):101272.
- Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.
- Emily Prud’hommeaux, Robbie Jimerson, Richard Hatcher, and Karin Michelson. 2021. Automatic speech recognition for supporting endangered language documentation. *Language documentation and conservation*, 15.
- Zarif Al Sadeque et al. 2022. *Automatic speech recognition for documenting endangered first nations languages*. Ph.D. thesis, University of Saskatchewan.
- Frank Seifart et al. 2006. Orthography development. *Essentials of language documentation*, pages 275–299.
- Jiatong Shi, Jonathan D. Amith, Rey Castillo García, Esteban Guadalupe Sierra, Kevin Duh, and Shinji Watanabe. 2021. Leveraging end-to-end ASR for endangered language documentation: An empirical study on yolóxochitl Mixtec. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1134–1145, Online. Association for Computational Linguistics.
- Katrine Falcon Søyby, Byurakn Ishkhanyan, and Line Burholt Kristensen. 2023. Not all grammar errors are equally noticed: error detection of naturally occurring errors and implications for eye-tracking models of everyday texts. *Frontiers in Psychology*, 14:1124227.
- Harimohon Thounaojam and Shobhana L Chelliah. 2007. The lamkang language: Grammatical sketch, texts and lexicon. *Linguistics of the Tibeto-Burman Area*, 30(1):1–212.
- Alexander Zahrer, Andrej Žgank, and Barbara Schuppler. 2020. Towards building an automatic transcription system for language documentation: Experiences from muyu. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2893–2900.
- Weiyi Zheng, Alex Xiao, Gil Keren, Duc Le, Frank Zhang, Christian Fuegen, Ozlem Kalinli, Yatharth Saraf, and Abdelrahman Mohamed. 2022. *Scaling ASR Improves Zero and Few Shot Learning*. In *Interspeech 2022*, pages 5135–5139.

A Lamkang Orthography

The following table displays graphemes in Lamkang.

B Hyperparameters

The following hyperparameters were used during training:

Grapheme	Phoneme	Example	Gloss
a	[a], occasionally [ə]	arhang kal ma	<i>Don't climb up</i>
aa	[a:]	prkhaa	<i>almond</i>
ai	[aj]	phaivang	<i>ant</i>
aai	[a:j]	psaai	<i>elephant</i>
ao or au	[a:w]	phkao; auva	<i>reptiles; that one</i>
b	[b]	baak rek	<i>bats</i>
ch	[tʃ], occasionally [ts]	chen	<i>to run</i>
d	[d]	dii	<i>water</i>
e	[e]	chet lam da	<i>they went</i>
ee	[e:]	mkheel thung bi ngu	<i>when they asked</i>
ei	[ej]	nei	<i>I</i>
h	[h]	heem	<i>to hit</i>
i	[i]	in	<i>house</i>
ii	[i:]	kmiing	<i>my name</i>
iiu	[i:w]	tkhiiu	<i>seven</i>
k	[k]	keel	<i>goat</i>
kh	[k ^h]	khuung	<i>drum</i>
l	[l]	loon	<i>hill</i>
m	[m]	mei	<i>fire</i>
n	[n]	nii	<i>day</i>
ng	[ŋ]	ngaa	<i>fish</i>
o	[ɔ]	non	<i>snout</i>
oo	[o:]	oon	<i>to call</i>
p	[p]	puu	<i>grandfather/uncle</i>
ph	[p ^h]	phul	<i>water pot</i>
r	[r]	raal	<i>war</i>
s	[s], occasionally [ç]	som	<i>ten</i>
t	[t]	talu	<i>what</i>
th	[t ^h]	thung	<i>inside</i>
thl or ṭl*	[ṭl]	thlaa	<i>moon/month</i>
tl	[tl]	tloo	<i>do</i>
tx or ṭ*	[ṭs]	txim	<i>half</i>
txh or ṭh*	[ṭs ^h]	txhi	<i>to lead</i>
u	[u]	thuk	<i>come out</i>
uu	[u:]	nuu	<i>mother</i>
uui	[u:j]	uui	<i>dog</i>
v	[v]	vak	<i>pig</i>
y	[j]	yaan	<i>night</i>
'	[ʔ] or [ʔ̣]	t'loo	<i>to take</i>

Table 3: Vowels and Consonants in Lamkang, adapted from Chelliah et al. (2023). *Grapheme versions without the combining dot diacritic are used in the current form of orthography

- learning_rate: 0.0001
- total_train_batch_size: 16
- train_batch_size: 8
- optimizer: Use Optimizer-Names.ADAMW_TORCH_FUSED with betas=(0.9,0.999) and epsilon=1e-08 and optimizer_args=No additional optimizer arguments
- eval_batch_size: 8
- seed: 42
- gradient_accumulation_steps: 2

- lr_scheduler_type: linear
- lr_scheduler_warmup_steps: 300
- num_epochs: 100
- mixed_precision_training: Native AMP

Improving Medical Hallucination Detection with System Combination and Rule-based Customization

Jonathan Lasko, Damianos Karakos, Francis Keith

RTX BBN Technologies

Cambridge, MA

{jonathan.lasko,damianos.karakos,francis.keith}@rtx.com

Abstract

The presence of factuality errors (hallucinations) in the outputs of patient-facing medical chatbots is a serious problem: they can lead to patient harm and erode people’s trust in the medical profession. For this reason, it is crucial to detect hallucinations in chatbot outputs and forward them to clinicians for review. In this paper, we present the system we built for detecting such errors: it consists of multiple LLM-powered detectors which are combined together with a novel alignment procedure. We ran our system on the MedExpert-Benchmark dataset (Yarmohammadi et al., 2025) and our results on two use cases, Mental Health and Prenatal Care, show that the combined system gives nice gains over the individual systems. Additionally, we show that further customization of the system to each one of the use cases leads to further gains, but at the cost of reduced generalizability. Our code and dataset are available here: <https://github.com/BBN-E/medic-customnlp4u>.

1 Introduction

It is generally accepted that patient-facing medical chatbots have the potential to transform the practice of medicine at many levels: more widespread and faster access to healthcare, supporting patients in understanding their condition and managing their symptoms, display of empathy and stronger guarantees on privacy. However, the presence of factuality errors (hallucinations) in the outputs of medical chatbots is still a significant problem, as they can lead the patient the wrong path in terms of diagnosis and treatment. Clearly, such errors can result in patient harm, ranging (in terms of severity) from minor to life-threatening.

In this paper, we present a hallucination detection system, consisting of multiple components which are combined together. The components consist of LLM-as-judge, RAG, and Agentic workflows, and they all “vote” on the presence of a

hallucination in the medical answer. The confidence of the detector, along with the estimated harm level and the automatically-generated explanation, are used as input into a novel combination scheme, which *aligns* the individual detections and combines their strengths, leading to improved performance. It is important to note that, in our setup, the error detectors can only *observe* the output of the chatbot; they cannot probe the chatbot by prompting it with variations of the question, or perturb its generations in some other way, or access its internal components. Although such perturbation approaches have been found to be successful for estimating the confidence of the chatbot (and, hence, its tendency to produce errors) (Manakul et al., 2023) they are not relevant in our case; the chatbot’s response is all we have available.

Additionally, we perform an analysis of the system’s failure modes that are due to non-medical reasons. We then use this analysis to come up with rules that customize the evaluation technology on each of the use cases under consideration. Application of the rules leads to improved performance, but at the cost of reduced generalizability.

2 Prior Work

Hallucination detection in the outputs of LLMs (and medical chatbots) has been an active area of research. For general hallucination detection, a popular approach (e.g., of (Manakul et al., 2023)) is that of generating multiple LLM outputs and then comparing them with each other to measure stability. However, as mentioned earlier, our setup does not allow us to generate multiple chatbot outputs per answer; one output is all we have. This also rules out white-box methods, which assume that likelihoods or other internal model components are observable. Multiple other approaches (e.g., (Agarwal et al., 2024; Pandit et al., 2025)) use LLM-as-judge for automatic hallucination detec-

tion, but these detections are not combined together. Although there are approaches for combining together hallucination detections such as (Chen and Mueller, 2024; Bouchard and Chauhan, 2025), they are not applied to the medical domain. A notable exception is the very recent paper by (Hussain et al., 2026), which uses machine learning on top of LLM-derived features. This approach is very different from the one we describe in this paper; we plan to compare the two in a future publication.

3 The Detection System

Our hallucination detection system takes as input a medical chatbot response and routes it through multiple detection components, which can potentially generate multiple detections per response. The outputs of these components are combined together using an alignment-based system combination approach, similar to how it is done in other fields such as Speech Recognition and Machine Translation (Fiscus, 1997; Rosti et al., 2008). The various system components are briefly described below.

3.1 Detection Components

LLM-as-Judge: We used a similar approach as in (Liu et al., 2025). Briefly, the LLM-as-judge detector utilizes the internal knowledge of the LLM to find hallucinations, using a prompt similar to the one shown in Section 10.9.1 of (Liu et al., 2025). Both open-source LLMs (e.g., Llama3-ChatQA-1.5-8b) and closed, API-based LLMs (e.g., GPT-5) are supported. To find hallucinations, the detector prompts an LLM to find sentences that contain hallucinations and outputs (i) each sentence containing a detected hallucination; (ii) an explanation for why this is a hallucination; and (iii) an estimated level of harm to the patient (chosen from “none”, “very low”, “low”, “medium”, “high”). The LLM-as-judge Detector loads open-source LLMs via the Hugging Face transformers API (Hugging Face) and utilizes langchain (LangChain) to solicit and parse structured responses.

LLM-as-Judge with RAG: Our RAG-based LLM-as-Judge hallucination detector (or, more simply, RAG Detector) uses retrieval-augmented generation (RAG) to augment the information an LLM can utilize while evaluating an answer for the presence of hallucinations. Specifically, the RAG Detector takes the original question and uses it with a retriever module optimized for the medical domain to query a database of high-quality medical

reference data (e.g., DSM-V and other medical textbooks, and the StatPearls medical test practice material). The corpora are taken from MedRAG (Xiong et al., 2024). Like the LLM-as-judge Detector, the RAG Detector uses the Hugging Face transformers and langchain APIs.

Agentic: Our agentic hallucination detector follows an approach similar to (Liu et al., 2025). It places hallucination detection agents in a semi-structured round-robin conversation with critic agents who point out various weaknesses in the detections. Each agent is assigned its task via user-provided system prompts, including instructions to use JSON formats for the detections. A reviewer agent supervises the conversation, instructing the detector agents to revise their detections based on feedback from critic agents, and determining when the revisions have converged. The conversation is terminated after a maximum number of rounds (30) are reached. The Harm Critic agent gives feedback on the patient harm levels assigned to each hallucination. The Explanation critic gives feedback on the explanations given for each detection by leveraging retrieval-augmented generation (RAG) from medical-specific text corpora for its critiques. For the system prompts used in the agentic detector, see Section 10.9.2 of (Liu et al., 2025).

MedScore: We used the two-step factuality detection system from (Huang et al., 2025). It first decomposes a sentence of a response into multiple claims (i.e., facts, statements) and then verifies each claim by a given gold reference (e.g. related medical snippets from PubMed, StatPearls, Medical Textbook). The decomposition step utilizes OpenAI’s gpt-4o-mini (Hurst et al., 2024) to find context information and decomposes the targeted sentence into independent, condition-aware claims. In the second step, MedScore evaluates the correctness of each claim by prompting a verifier LLM (OpenAI’s GPT-4o (Hurst et al., 2024)) to judge the claim’s correctness using its internal knowledge. Optionally, the verifier can be configured to use the claim as a query to search for the top ten relevant medical snippets from the MedRAG corpus (Xiong et al., 2024); the verifier LLM then uses these snippets as a gold reference/context.

3.2 System Combination

As mentioned earlier, each hallucination detector outputs one or more detections per answer. In our approach to system combination, we aim at increasing the confidence of the detections that agree

with each other and (conversely) reducing the confidence of detections that do not agree with each other. To measure agreement, we proceed incrementally, by aligning the outputs of two detectors at a time: we first compute the cosine similarity between the embeddings of the explanations given by two detectors, and then use the Hungarian algorithm (Wikipedia contributors, n.d.) to align the two sets of detections together. If two detections are aligned, they are “merged” into one, and its score/confidence is set to the linear combination of the individual confidences. If a detection is not aligned with any other detection then its confidence is essentially interpolated with zero, thus, reducing its strength (discounted).

4 Experiments and Results

We ran our detection experiments on the publicly available dataset MedExpert-Benchmark (Yarmohammadi et al., 2025). It consists of two medical use cases, Mental Health (MH) and Prenatal Care (PC), consisting of 280 and 260 chatbot answers, respectively. The hallucination rates are: 27.5% for MH and 26.8% for PC. We measure performance using the F1 metric.

Results with the aforementioned detectors and with the system combinations appear in Table 1. In the case of MedScore (which uses GPT-4o) and LLM-as-a-judge with GPT-5, each run produces different results due to the random sampling done by these LLMs. For this reason, Table 1 shows the *average* precision, recall, and F1 across multiple runs of these two systems – four for MedScore, and five for LLM-as-a-judge with GPT-5. All individual samples are included in the system combinations.

As can be seen from these results, performance in terms of precision and recall can vary widely as a function of the approach and the underlying model. For example, for Mental Health, although their F1 scores are almost identical, GPT-5 as LLM-as-judge results in the best precision (38%) and low recall (63.4%) while the system combination without GPT-5 has a lower precision (31.8%) and a very high recall (81.8%). The system combination we propose does seem to combine the strengths of the different systems: combining the two aforementioned systems results in the best F1 score (51.9% for Mental Health, 51.8% for Prenatal Care) without using any complex machine learning system, resulting in both improved precision and recall. In-

terestingly, the combinations without GPT-5 result in F1 that is on par with GPT-5 alone.

5 Error Analysis

We looked at some of the errors made by our system on our MedExpert-Benchmark (Yarmohammadi et al., 2025) dataset. We focused on the *non-medical errors* made by LLM-based detector(s). These are errors caused by non-adherence to the provided guidelines, or due to a misunderstanding of the chatbot answer, or structural issues. The errors with a medical basis were forwarded to a clinician on our team; his analysis will appear in future work.



Figure 1: Breakdown of false positive detections of non-medical nature, based on our team’s manual review. They are sorted from least frequent to most frequent).

We manually reviewed 192 hallucination detections made by our system over our two use cases. We focused mainly on false positives, as these are the ones affecting our precision score. We found out that 61 of these false positives (about 31.8%) did not seem to have a relevant medical reason. Figure 1 shows a breakdown of the identified error patterns over both use cases. Explanations of the most frequent of these patterns appear in Table 4 in the Appendix.

6 Customization and Compromises

In this section, we try to answer the question: how can we quickly improve performance on

Method	Model(s)	Mental Health			Prenatal Care		
		F1	Precision	Recall	F1	Precision	Recall
System Combination	Llama3-ChatQA-1.5-8B, Mistral-Nemo-Instruct-2407, GPT-4o-mini, GPT-4o, GPT-5	0.519	0.368	0.883	0.518	0.402	0.729
System Combination (no GPT-5)	Llama3-ChatQA-1.5-8B, Mistral-Nemo-Instruct-2407, GPT-4o-mini, GPT-4o	0.458	0.318	0.818	0.488	0.343	0.843
LLM-as-Judge	Llama3-ChatQA-1.5-8B	0.435	0.282	0.948	0.405	0.276	0.757
LLM-as-Judge*	GPT-5	0.466	0.380	0.634	0.486	0.361	0.786
RAG	Llama3-ChatQA-1.5-8B	0.445	0.286	1.000	0.432	0.279	0.957
Agentic	Mistral-Nemo-Instruct-2407	0.407	0.309	0.597	0.435	0.297	0.814
MedScore*	GPT-4o-mini, GPT-4o	0.372	0.279	0.562	0.465	0.324	0.825

Table 1: Hallucination detection results on MedExpert-Benchmark. The best result per use case is in **bold**. Rows for methods utilizing GPT-4 or GPT-5 are denoted with an asterisk (*) indicating that the Precision, Recall and F1 score are averaged across multiple runs to account for non-deterministic variations in the responses. LLM-as-Judge GPT-5 scores are averaged across 5 sample runs whose individual F1 scores ranged between 0.434 – 0.506 for MH and 0.477 - 0.508 for PC. MedScore scores are averaged across 4 sample runs whose individual F1 scores ranged between 0.361 - 0.395 for MH and between 0.444 - 0.488 for PC. Because of this, combining those rows’ averaged precision and recall with the F1 formula does not result in the averaged F1 score we report in that row.

Use case	Rules	Original F1	New F1 after applying rules	New F1 after applying rules of other use case
Mental Health	Snippet says talk to a medical professional (MH) + Snippet Resembles Explanation (Jaccard) + Snippet Resembles Explanation (Levenshtein)	0.519	0.540	0.475
Prenatal Care	Snippet says talk to a medical professional (PC) + Snippet Resembles Explanation (Jaccard) + Snippet Resembles Explanation (Levenshtein) + Explanation is a critique of risk framing + Explanation references guidelines	0.518	0.536	0.515

Table 2: The impact of applying error-correcting rules to filter out false positives is approximately 2% absolute increase in F1 score for each use case.

MedExpert-Benchmark by customizing the detection system on the use cases of interest, and what compromises are required? The customization was done by creating sets of rules that address some of the types of errors we found in our analysis: the rules remove some of the erroneous detections made by our system.

Using the results from the error analysis described in Section 5 (where we manually reviewed false positives from our detectors against the MedExpert dataset), we crafted rules for filtering out false positives. Some of these initial rules for eliminating false positives (snippet/explanation similarity, omission phrases) showed promise. We then prompted GPT-5-thinking with two files, the annotated false positives and the full set of detections, for each use case, and instructed the LLM to identify patterns differentiating false positives from true

positives. This produced an additional eight rules. Table 5 in the Appendix has the full set of rules.

Using these rules to filter out positives achieves approximately 2% absolute improvement in F1 score, for each use case, as shown in Table 2. Since each rule was derived from the outputs of our system using a single use case of the MedExpert dataset, they are a form of customization to the use case. Some rules may generalize across use cases, but others utilize domain-specific terminology (e.g., therapist, DSM, depression, serotonin). The best results for each use case were achieved using *different* rule sets, and using the rule set from one use case degraded performance on the other use case, as shown in the last column of Table 2. This means that these rules result in some kind of overfitting to the use cases and to specific patterns in our system’s outputs.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	—	0.107	0.0146	0.0998	0.0988	0.6308	0.188	0.109	0.0842	0.0006	0.0044	0.4874	0.0272	0.0	0.0
2	0.0996	—	0.2336	0.0088	0.0128	0.129	0.0224	0.4118	0.5442	0.0284	0.0324	0.9862	0.227	0.0	0.0
3	0.5706	0.16	—	0.0016	0.0032	0.0556	0.0066	0.6256	0.8732	0.0754	0.0636	0.736	0.4502	0.0	0.0
4	0.83	0.155	0.5924	—	0.8776	0.0792	0.6602	0.005	0.0046	0.0002	0.0002	0.0668	0.003	0.0	0.0
5	0.0838	0.0078	0.0352	0.0468	—	0.1666	0.8172	0.0086	0.0056	0.0004	0.0002	0.0644	0.0064	0.0	0.0
6	0.296	0.0362	0.156	0.297	0.3846	—	0.2568	0.0652	0.0806	0.0042	0.0014	0.3202	0.054	0.0002	0.0
7	0.0108	0.0006	0.0024	0.0044	0.2366	0.0446	—	0.0098	0.0108	0.0004	0.0002	0.1212	0.007	0.0	0.0
8	0.0	0.0	0.0	0.0028	0.068	0.0224	0.3378	—	0.6076	0.613	0.07	0.3416	0.9268	0.0924	0.0264
9	0.0286	0.0026	0.0104	0.1568	0.6536	0.3594	0.8294	0.1956	—	0.0378	0.0152	0.6186	0.7424	0.0014	0.0002
10	0.0466	0.0018	0.0146	0.1928	0.7772	0.4178	0.6566	0.1544	0.8508	—	0.3094	0.1458	0.4574	0.0276	0.0008
11	0.0414	0.0032	0.0134	0.2054	0.7564	0.4382	0.6808	0.1388	0.8562	1.0	—	0.0148	0.2158	0.66	0.2658
12	0.2414	0.0362	0.1786	0.289	0.7222	0.4964	0.886	0.4738	0.9688	0.8608	0.8654	—	0.5212	0.0084	0.0026
13	0.0108	0.0006	0.0024	0.0044	0.2366	0.0446	1.0	0.3378	0.8294	0.6566	0.6808	0.886	—	0.0146	0.0012
14	0.0032	0.0002	0.0012	0.0004	0.04	0.0094	0.1006	0.7266	0.1872	0.1312	0.1144	0.335	0.1006	—	0.0008
15	0.0004	0.0002	0.0008	0.0002	0.014	0.001	0.0282	0.3478	0.066	0.043	0.0338	0.1644	0.0282	0.169	—

Table 3: Combined pairwise p-value table. The upper-triangular entries correspond to Mental Health, the lower-triangular entries correspond to Prenatal Care. Boldface indicates statistical significance ($p \leq 0.05$).

7 Comparing Hallucination Detection Outputs

We used a paired bootstrap test on F1 (Berg-Kirkpatrick et al., 2012) to compare all pairs of hallucination detection outputs, including using the rules of Table 2. The p-values from all these comparisons are shown in Table 3; the upper-triangular entries correspond to MH and the lower-triangular entries correspond to PC (we combined the two use cases into a single table to save space). The rows (and columns) correspond to the following outputs: row 1: Agentic
row 2: LLM-as-Judge (Llama3)
row 3: RAG
rows 4-7: four MedScore runs
rows 8-12: five LLM-as-Judge runs with GPT-5
row 13: System Combination without GPT-5
row 14: System Combination with GPT-5
row 15: after applying rules

Some observations are in order: (i) The differences between the “weakest” systems (low-index rows and columns) and the “strongest” systems (high-index rows and columns) are clearly statistically significant; these correspond to the table entries far from the main diagonal. (ii) The MH case has 56 pairs (53.3%) of outputs that have a statistically significant difference, while the PC case has 45 such pairs (42.9%); this could be a consequence of the fact that PC has fewer data points. (iii) The system combinations give statistically significant gains when compared to most of the single-system outputs that were included in the combination. However, the difference from the *best* single system included in the combination is not always statistically significant. (iv) Outputs obtained with the same “family” of systems via sampling (such as MedScore runs or LLM-as-Judge runs with GPT-5) do not always have a non-statistically significant

difference; for example, half of the differences between MedScore runs are statistically significant in the PC case, while no MedScore differences are statistically significant in the MH case (rows/columns 4-7). We see the opposite trend in the case of LLM-as-Judge with GPT-5 (rows/columns 8-12).

8 Concluding Remarks

In this paper, we tried to answer a number of research questions related to the evaluation of the factuality of medical chatbots. We found that our detection system’s main weakness is its low precision; it falsely detects too many hallucinations, despite the fact that system combination does improve precision and/or recall over the best individual system. Our error analysis of these false positives revealed that there are many cases in which the LLM-as-judge does not properly follow directions, misunderstands the task, or simply hallucinates about the presence of errors. Based on what we found, we came up with various rules (and we prompted GPT-5 for additional rules) that helped us reduce the prevalence of false positives. These rules are a form of customization of the system to the medical use case; using these rules on a different use case degraded results. We hope that this finding will help guide future research in this area.

9 Acknowledgments

This research was, in part, funded by the Advanced Research Projects Agency for Health (ARPA-H). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Government.

References

- Vibhor Agarwal, Yiqiao Jin, Mohit Chandra, Munmun De Choudhury, Srijan Kumar, and Nishanth Sastri. 2024. Medhalu: Hallucinations in responses to healthcare queries by large language models. *arXiv preprint arXiv:2409.19492*.
- Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 995–1005.
- Dylan Bouchard and Mohit Singh Chauhan. 2025. Uncertainty quantification for language models: A suite of black-box, white-box, llm judge, and ensemble scorers. *arXiv preprint arXiv:2504.19254*.
- Jiuhai Chen and Jonas Mueller. 2024. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5186–5200.
- Jonathan G Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *1997 IEEE workshop on automatic speech recognition and understanding proceedings*, pages 347–354. IEEE.
- Heyuan Huang, Alexandra DeLucia, Vijay Murari Tiyyala, and Mark Dredze. 2025. Medscore: Factuality evaluation of free-form medical answers. *arXiv e-prints*, pages arXiv–2505.
- Hugging Face. Transformers. <https://huggingface.co/docs/transformers/en/index>. Online; accessed 2026-04-02.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Khizar Hussain, Bradley A Malin, Zhijun Yin, Sannah Leigh Rose, and Murat Kantarcioglu. 2026. Blending human and llm expertise to detect hallucinations and omissions in mental health chatbot responses. *arXiv preprint arXiv:2604.06216*.
- LangChain. Langchain docs. <https://docs.langchain.com/oss/python/langchain/overview>. Online; accessed 2026-04-02.
- Jonathan Liu, Haoling Qiu, Jonathan Lasko, Damianos Karakos, Mahsa Yarmohammadi, and Mark Dredze. 2025. Statistically significant results on biases and errors of llms do not guarantee generalizable results. *The Second Workshop on GenAI for Health Potential, Trust, and Policy Compliance (GenAI4Health@NeurIPS 2025)*.
- Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.
- Shrey Pandit, Jiawei Xu, Junyuan Hong, Zhangyang Wang, Tianlong Chen, Kaidi Xu, and Ying Ding. 2025. Medhallu: A comprehensive benchmark for detecting medical hallucinations in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2858–2873.
- Antti-Veikko Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. 2008. Incremental hypothesis alignment for building confusion networks with application to machine translation system combination. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 183–186.
- Wikipedia contributors. n.d. Hungarian algorithm — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Hungarian_algorithm. [Online; accessed 2026-04-02].
- Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251.
- Mahsa Yarmohammadi, Alexandra DeLucia, Lillian C Chen, Leslie Miller, Heyuan Huang, Sonal Joshi, Jonathan Lasko, Sarah Collica, Ryan Moore, Haoling Qiu, and 1 others. 2025. Medexpert: An expert-annotated dataset for medical chatbot evaluation. In *Machine Learning for Health 2025*.

A Appendix

Error Category	Description
Explanation agrees with snippet	Detector copies or near-copies the snippet as its explanation
Detector focusing on trivialities	Snippet and explanation point to an insignificant issue
Omission not hallucination	Explanation flags a missing piece rather than a hallucinated claim
Issue addressed elsewhere in the answer	Detector ignores other parts of the answer that address the concern
Detector misconstrues answer statement	Detector misinterprets the original answer
Snippet is consistent with literature	Detector declares that snippet agrees with literature
Explanation claims snippet not found in answer	Detector wrongfully claims that snippet is not in the answer
Just quoting a snippet from the answer	Detector just copies a snippet from the answer
Explanation seems irrelevant	Explanation is not relevant to the snippet
Snippet is hallucinated	The snippet does not appear in the answer, even in approximate form
Detector unsure about use case	Detector gives explanation that is irrelevant to use case

Table 4: Description of the non-medical false-positive error categories

Use Case	Rule	Implementation Details (Key Phrases or Thresholds)	F1
MH	Snippet resembles Explanation (J)	Jaccard similarity between 3-grams from snippet and 3-grams from explanation	0.524
MH	Snippet resembles Explanation (L)	Normalized Levenshtein similarity score between snippet and explanation	0.525
MH	Snippet says talk to a medical professional	consult (alyour) (doctor physician professional therapist)	0.533
MH	Snippet has non-medical hedging	may might can could often sometimes generally typically BUT NOT mg dosel dosage treatment therapy diagnosis diagnostics schizophrenia depression bipolar anxiety melatonin serotonin ssri snri antidepressant antipsychotic benzodiazepine symptom disorder disease	0.504
MH	Snippet has psycho-education	can affect can influence can impact often involves may experience may develop may have symptoms can lead to (symptoms changes) can cause (symptoms changes) is often associated with may include symptoms	0.509
MH	Explanation supports Snippet	is consistent with	0.521
MH	Explanation contains omission phrases	((important crucial essential fails) to does not doesn't omit lacks) (mention state note emphasize highlight)	0.490
MH	Explanation references guidelines	guideline guidelines current evidence literature DSM ICD criteria	0.500
MH	Explanation overly long	Threshold at 55 words	0.500
MH	Explanation critiques framing	may mislead patients create barriers framing overemphasizing	0.520
PC	Snippet resembles Explanation (J)	Jaccard similarity between 3-grams from snippet and 3-grams from explanation	0.523
PC	Snippet resembles Explanation (L)	Normalized Levenshtein similarity score between snippet and explanation	0.520
PC	Snippet says talk to a medical professional	contact your healthcare provider, talk to your doctor, seek medical attention, consult your doctor	0.523
PC	Snippet has reassurance	mild common temporary normal usually resolves should resolve	0.518
PC	Snippet has causal hedging	(can may) (cause lead to result in)	0.512
PC	Explanation supports Snippet	is consistent with	0.518
PC	Explanation contains omission phrases	((important crucial essential fails) to does not doesn't omit lacks) (mention state note emphasize highlight)	0.489
PC	Explanation critiques risk framing	downplay, understate risk, minimize risk, risk framing	0.523
PC	Explanation references guidelines	guidelines, current evidence, literature, DSM, criteria	0.526
PC	Explanation overly long	Threshold at 55 words	0.479
PC	Explanation critiques framing	may mislead patients create barriers framing overemphasizing	0.516

Table 5: Full list of rules, with corresponding effects on F1 score. Improved F1 scores are shown in **bold**.

Asking the Right Questions: Can expert-prompted LLMs reformulate legal queries from non-experts?

Katherine Atwell^{*1}, Morgan A. Gray^{*2}, Jaromir Savelka³, Len Rial⁴,
Sera Linardi⁵, Malihe Alikhani¹,

¹Northeastern University, ²University of Saint Thomas, ³Carnegie Mellon University,
⁴ Allegheny County Department of Human Services, ⁵University of Pittsburgh

{atwell.ka,m.alikhani}@northeastern.edu, gray7776@stthomas.edu,
jsavelka@cs.cmu.edu, len.rial@alleghenycounty.us, linardi@pitt.edu

^{*}Denotes equal contribution

Abstract

Large language models are widely used by everyday users, and can be asked to perform tasks that require specialized expertise, such as interpreting contractual terms and conditions, filing personal taxes, or diagnosing medical symptoms. Although these tools should not be used in place of professional advice, they can be useful starting points for users seeking professional help, improving users’ access and interactions with professionals.

In this vein, this paper introduces a legal question reformulation task to assist non-experts in their interactions with lawyers. This has the potential to streamline discussions between lawyers and clients, who may not know the correct legal language to communicate their needs. Using a novel evaluation framework informed by legal expertise, we investigate the quality of model-generated legal question reformulations on in-the-wild data from non-experts seeking legal advice. Our findings indicate that LLMs have significant potential in legal reasoning, but some unexpected safety concerns may emerge. Further, adding linguistically-aligned in-domain text samples can improve performance for smaller models, even when the samples are not aligned factually with the given question.

1 Introduction

Individuals take legal actions in many everyday situations, such as signing leases, employment agreements, or terms of use. They must also navigate interactions with institutions and government systems that are governed by complex legal rules and procedures (Ahn et al., 2022). However, they typically do not have a lawyer present when they engage in these tasks; legal advice is expensive to obtain, as individually lawyers spend years in school learning their craft. This gap in legal understanding can limit individuals’ ability to advocate for themselves, understand their rights, or effectively

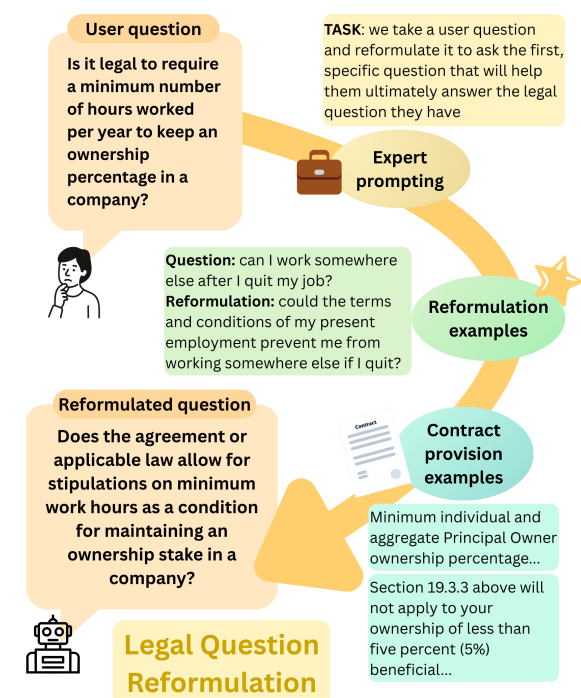


Figure 1: A depiction of the legal question reformulation task and the approach we describe in this work, with an example of a user question from Justia reformulated by GPT 4 Turbo.

engage with systems that shape their daily lives. Although lawyers are trained to “issue spot”, or identify from factual descriptions what legal rights or obligations an individual may have, clear communication between lawyers and clients is crucial for effective representation (Muchman, 2023). It is thus imperative that a non-lawyer seeking help be able to clearly communicate their question or circumstances. Misunderstandings between a lawyer and their client can cause avoidable problems in legal representation (Bryant, 2001): they can lead to missed issues, inefficient performance, and frustration.

Engagement between a lawyer and a non-expert, such as an individual seeking legal advice, often starts with a question by the non-expert. A simple question, such as “If I quit my job, can I work somewhere else?”, could raise a host of complex issues involving a number of parties and relevant documents. In the interest of exactitude, it is helpful that a non-expert ask an accurate question. However, a lack of knowledge of legal concepts and terms can cause non-experts to ask questions that are difficult for the lawyer to understand and interpret (Branting, 2017).

In this work, we study how LLMs can improve access and streamline communication with professionals by addressing the novel task of legal question reformulation: reformulating a non-expert’s question to better capture and communicate what legal issue they may have. We choose to address this task because these questions often set the stage for interactions between experts and non-experts (Cheong et al., 2024). Knowing the correct questions to ask can lead to more efficient, less expensive, and more productive interactions with lawyers.

We bring together experts in law and NLP to provide a comprehensive approach for evaluating the use of LLMs for such a task. Utilizing the expert skills of two lawyers with expertise in NLP, we prompt LLMs to reformulate users’ legal questions, and experiment with adding contextually-relevant contract clauses from the publicly-available CUAD dataset (which contains commercial contracts) (Hendrycks et al., 2021) to the prompt as examples of legal language. To assess the efficacy of contextually-relevant clauses from commercial contracts, we focus on questions concerning business/commercial law and contracts, obtaining publicly-available legal forum posts from real users to represent the types of questions users are likely to ask and the language they may use. An illustration of the legal question reformulation task, and the pipeline we introduce in this work, can be found in Figure 1.

Summarily, we address the following research questions:

- RQ1** Can LLMs reliably reformulate legal questions by non-experts?
- RQ2** How does the inclusion of linguistically relevant but not necessarily factually relevant language influence the zero-shot task performance of LLM with expert-tuned prompts?

- RQ3** Are there any safety concerns that would prevent an LLM’s deployment for this task?

Although we achieve promising results, we ultimately conclude that these models are not currently suitable for safe deployment in a zero-shot setting, as even the most advanced models tested cannot reliably generate legally-correct reformulations. Further, we identify a set of safety concerns associated with these models, and discuss cases where reformulations should not be provided. We discuss these issues in-depth, in order to inform the design of future systems and highlight some general safety concerns in the space of legal NLP.

We provide the text of our prompts in full in the Appendix, and will release our prompting code in full upon publication, including our code for retrieving the relevant contract information given the text of a question. We will also release a dataset with the generated reformulations from each of our baselines, along with code for retrieving the original data.

2 Related Work

Underspecified and Ill-Formed Questions Correctly answering a question requires understanding precisely what the question is asking. This becomes more difficult in the face of questions that are not well-formed (Faruqui and Das, 2018) or are underspecified (Christianson, 2016), where the desired information needs are unclear. Question reformulation (Chu et al., 2020; Vakulenko et al., 2020), requesting clarification (Fu et al., 2022; Huang et al., 2023), and selective question-answering (Cole et al., 2023) have been proposed as ways to handle unclear questions. In this work, we focus on question reformulation for the legal domain, where understanding how to ask the right questions can require a great deal of time and effort for legal professionals (Cheong et al., 2024).

Question Reformulation Within the Legal Domain Although there has been work on reformulating queries intended for legal search engines such as Shankar and Buddarapu (2019a,b), to our knowledge, the question reformulation task we propose is novel. Several existing works in the legal space motivate this proposed task. Branting (2017), in the context of another problem, discuss the differences between how laypersons and lawyers would pose a legal query. The authors note gaps in terminology and the complexities associated with legal

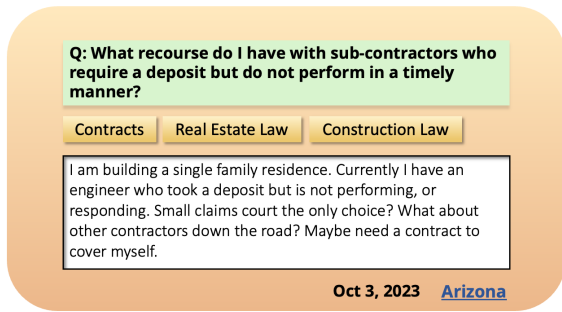


Figure 2: Sample user question from Justia Ask a Lawyer, with metadata. We scraped user questions posted in the Contracts and Business Law sections of this website from 2010 to 2023 in order to evaluate LLMs on in-the-wild user questions.

concepts as a barrier between a lawyer’s and layperson’s understanding. Further, in interviews with legal experts conducted by Cheong et al. (2024), experts propose question refinement as a possible use case for LLMs in the legal domain, explaining that this process can be time-consuming for lawyers and LLMs could help streamline some initial interactions with clients.

3 Data

3.1 Legal Questions

Existing work in the space of legal AI recommends that researchers evaluating AI systems focus on naturalistic evaluations that mimic possible real-world use. Because our goal is to develop and evaluate expert-tuned prompts, we do not experiment with prompts created by non-experts. Instead, we evaluate our approach on real user questions, collecting a dataset of real-world questions asked to lawyers in text form on public forums. To compile legal questions posted in-the-wild by non-expert community members, we scrape publicly available data from Justia Ask a Lawyer^{*}, a website where users can post questions in discussion forums asking for legal advice in their particular state. Posts are tagged with the state and relevant legal domain(s), which include Contracts, Real Estate Law, and Personal Injury. Practicing lawyers registered to the site can reply with answers to these questions, and users can upvote or report these responses. A sample user post, with metadata, can be found in Figure 2.

Another recommendation from existing literature in legal AI is to evaluate system performance in constrained settings (Kapoor et al., 2024) in order

^{*}<https://answers.justia.com/ask>

Selected User Questions

“Refused to sign a termination agreement, can I get in trouble?”
 “Just curious if a waiver I was pressured to sign is legally bounding or mean anything. Thanks”
 “Can my current employer sue me?”
 “can an entity sign on behalf of itself”

Table 1: Examples of user questions from our dataset, which were posted on Justia’s Ask a Lawyer forums under Business Law and Contract Law. Many of these questions are underspecified or do not use appropriate legal language.

to ensure the evaluation is thorough. In this work, we focus on evaluating LLM reformulations of contract law questions within the business domain. We focus on contract law in particular because many situations where people often seek legal advice involve contract law (Bib, 2024), and restricting our domain to business contracts allows us to more easily identify contracts with relevant language (which we describe in the following section).

To collect our dataset of user questions, we scrape all available contract questions from Justia[†], spanning from 2010 to 2023, and select the subset of questions tagged “Business Law”, giving us a total of 274 legal questions posted by users in this space. Examples of these questions can be found in Table 1. We observed several common patterns associated with these questions, including the following:

- **No question or implicit question:** A few user questions were formulated as statements describing their situation without any direct questions. These could typically be interpreted as general questions about users’ legal options or important legal considerations, and a couple contained implicit questions by stating things that the user wanted to do (such as buying a gun). For example, one user said “*Hello, I have an Amazon store, I hired an automation company, they seem fraudulent and taking my cash.*”
- **Under-specified question:** Many questions were underspecified to some degree, and were either missing details about a given scenario or not precise about what exactly the user wanted to know. For instance, one user simply asked “*Can my current employer sue me?*”
- **Questions without appropriate legal language:** Some questions also did not use precise or accu-

[†]<https://answers.justia.com/questions/answered/contracts>

rate legal language, including appropriate terms of art. For instance, one user asked "I deliver gravel. Payment required upon delivery. What should I do if someone refuses to pay?" There is no mention here of the mechanism that enforces payment upon delivery, such as a service agreement.

To gain some insight into the characteristics of our dataset, we analyze the metadata provided by Justia. We find that the questions in our dataset more often pertain to California or Florida state law than any other states. A plurality of the questions in our set were posted in 2023, and questions had an average length of 22.18 tokens. We provide distributional plots in Appendix A.

Due to Justia's data usage policy, we are not allowed to release our dataset directly. However, each question is assigned an ID on the Justia website, and with the publication of this work we release the set of IDs corresponding to the reformulated questions, as well as the text of the reformulations themselves. We also release a script for scraping the website, which retrieves the ID and the text of each question, so researchers who wish to use our data can reconstruct the question-reformulation pairs by mapping the IDs to their corresponding questions.

3.2 Relevant Contract Clauses

For domain-specific tasks, expert knowledge in prompts is useful. It is infeasible, however, to capture the language an expert may use in every situation. Thus, a general prompt (even if tuned by experts) may lack specific details or language relevant to a particular input. To address this problem, we include provisions from the CUAD database (Hendrycks et al., 2021) in our prompt, specifically instructing the model that the language provided may or may not be relevant, and emphasizing that provisions are to be used only to give the model an idea of the language it could use. We refer to this as **linguistically aligned language**. We use the CUAD dataset because it is a collection of commercial contracts, which pairs with the Business/Commercial nature of the users' questions. To retrieve the most similar clauses for each question, we first represent the question, and the CUAD clauses, as embeddings using OpenAI's text-embedding-ada-002. We then calculate the cosine similarity score between the question and each clause, and map the question to the 10 clauses

with the highest similarity score. Examples of user questions and corresponding linguistically aligned clauses can be found in Table 5 in Appendix B.

4 Experiments

4.1 Expert Prompting

To formulate our prompts, we utilize the unique expertise of a lawyer studying NLP (an author on this paper). The prompts are tuned using an iterative approach to prompt formulation by manually inspecting outputs for correctness and adjusting the prompt accordingly based on erroneous output.

The idea behind the formulated prompts is that these are very similar to the same instructions that one expert would provide to another capable individual if performing the same task. A similar style of prompting is described by experts in (Gray et al., 2023; Savelka et al., 2023). Templates of all of our prompts used in our experiments are provided in full in §C for the purpose of reproducibility.

4.2 Providing Relevant Legal Language

To supplement our expert-designed prompts, we incorporate contract provisions deemed similar to the user question, in order to provide linguistically aligned language that may assist model generation. We use the procedure outlined in §3 to choose the 10 clauses from the CUAD dataset that are most similar to the question, and insert these into our prompt template as shown in Appendix C (examples shown in Table 5). Specifically, the last row shows how relevant provisions may return highly relevant language. The question speaks to a "contract . . . being broken". The returned provision contains the proper legal term for this: **breach**.

4.3 Models and Experimental Settings

To study our three research questions presented in §1, we run three large language models on our set of 274 legal questions from Justia (§3), with and without relevant legal provisions, using the prompts given in Appendix C. We first run OpenAI's GPT-4-Turbo (gpt-4-0125-preview) (Achiam et al., 2023) at a temperature of 0.0, with a maximum of 250 new tokens, a top p value of 1, a frequency penalty of 0, and a presence penalty of 0. We chose these parameters because, as (Gray et al., 2023; Savelka et al., 2023) explain, it is vital in the legal domain to obtain as deterministic an output as possible.

We then run two open-source language models: Meta’s LLAMA 3 70B chat model (Llama-3-70b-chat-hf) (Dubey et al., 2024), with 70 billion parameters, and Mistral (Mistral-7B-Instruct-v0.2) (Jiang et al., 2023), with 7 billion parameters. We again set the hyperparameters with the aim of obtaining deterministic output. Here, rather than setting the temperature to 0, we set the `do_sample` parameter to `False`, to achieve greedy decoding. We set the maximum length to 250, maximum new tokens to 125, top k to 50, number of return sequences to 1, repetition penalty to 0.01, and length penalty to 0.01. For these models, it is important to note that using the `<s>` sequential tag and the `[INST]`, `[/INST]` instruction tags, with grammatical instructions to output in JSON format, were necessary to achieve a reasonable level of performance in the zero-shot context. Without these inclusions, model output was unstable in terms of length, understanding the task, and generating meaningful output. An example of this can be found in Appendix D.

5 Results

After prompting the LLM baselines used in our experiments, we conduct a thorough quantitative and qualitative evaluation on their outputs, using an evaluation conducted by legal experts in our group, in accordance with best practices (Kapoor et al., 2024). In this section, we present our results and introduce our expert-based evaluation criteria.

5.1 Automatic Evaluation

As a first step towards evaluating our expert prompting strategy, we start by asking whether our reformulated questions are truly more clear and legally precise than the original user question. To do so, we use an LLM-as-a-judge approach, prompting Claude Haiku 4.5 to judge whether our reformulations are more **clear and legally precise** than the original questions, while also preserving the details of the original question. We do this both for our expert-tuned prompts (see Appendix C and a baseline non-expert-tuned prompt crafted by an author on the paper who is not a lawyer. This non-expert-tuned prompt reads: *Reformulate the following legal question to make it more specific and legally precise. {{USER QUESTION}}*

We report our results in Table 2. We observe that, for each baseline, the performance with expert tuning is substantially higher than with non-

	Non-expert	Expert
GPT-4-turbo	0.073	0.693
Llama3-70B	0.053	0.513
Mistral-7B	0.050	0.208

Table 2: Performance of baseline models on our question reformulation task using LLM-as-a-judge. We find that, for all three models, the performance with expert tuning is substantially higher than with non-expert tuning (at least 4 times higher for each baseline).

expert tuning, with models at least 4 times as likely to judge expert-tuned reformulations to be high quality (and more precise/clear than the original question) compared to non-expert-tuned reformulations. We also observe that GPT-4-Turbo yields the best results overall, with 69.5% of the reformulated questions judged to be high-quality when the prompt is tuned by experts. We also observe that Llama 3-70B is much more “coachable” than Mistral-7B; while they have very similar performances when prompted by non-experts, the performance for Llama3-70B is substantially higher than Mistral-7B under expert prompting.

5.2 Human Evaluation

We evaluate our models in two settings. The first is the expert prompt without any relevant provisions included, and the second is the expert prompt with linguistically aligned contractual provisions based on the user’s question (§3). To evaluate the model from a legal perspective, we use three criteria: Legal Correctness, Factuality, and Clarity/Understanding. We define these criteria here:

1. **Legal Correctness:** Based on the original text of the question (dependent on the complexity of the issue) which reformulation better captured any potential legal issue raised by the original question?
2. **Factuality:** Which reformulation most closely aligns factually with the original question?
3. **Clarity:** Which question produced a more clear, precise, and understandable reformulation without using unnecessary legalese or complex language? A term of art, appropriately used, should not be counted as legalese.

Each unit of evaluation relatively compares a reformulation of a user’s question with and without relevant provisions. For each category, if one reformulation outperforms another one, the better reformulation receives a 1 and the poorer a 0. If

	Legal Correctness		Factuality		Clarity	
	Base	Provs	Base	Provs	Base	Provs
gpt-4	0.75	0.65	1.0	0.84	1.0	0.94
llama3	0.56	0.71	0.78	0.94	0.89	0.92
mistral	0.31	0.37	0.59	0.53	0.39	0.53

Table 3: Performance of baseline models on our question reformulation task for Legal Correctness, Factuality, and Clarity. Because individual scores are either 0 or 1, when averaged, scores closer to 1 represent higher performance on average than scores closer to 0.

both reformulations are equally satisfactory, both are scored as 1. If both reformulations are equally unsatisfactory, both are scored as 0. Annotator agreement statistics are reported in Appendix E.

We also include a qualitative free-response category where experts can identify potential harm or safety issues, note extraordinary examples of good or bad performance, and discuss common mistakes made by models.

5.3 Quantitative Evaluation

To evaluate each model’s performance for each criterion, we randomly sample a subset of questions (the same subset was used for each model) for evaluation. We annotate 300 model outputs in total (i.e. reformulations of user posts). Each annotator annotated 150 reformulations, 50 for each model, with an overlap of 150. During evaluation, the prompt (with or without relevant provisions) from which the question was generated is unknown to the evaluators. We examine the performance of each model on each category in Table 3.

For all models investigated, GPT-4-Turbo without provisions included in the prompt has the best overall performance, and including linguistically-aligned language in the prompt actually decreased this model’s performance across all categories, particularly Factuality. It is possible that this drop may be because linguistically-aligned examples may not be factually-aligned with the question but, as discussed in §6, because GPT is not an open-source model it is impossible to isolate particular causes for changes in behavior.

On the contrary, including relevant provisions increased the performance of Llama3-70B across all categories and for Mistral-7B for everything but factuality. In fact, Llama3-70B with provisions was the second-highest-performing baseline, outperforming GPT with provisions.

Overall, performance was lowest in the legal cor-

rectness category, compared with factuality and clarity, which aligns with our intuitions. Identifying potential legal issues is a necessary skill for lawyers and can be challenging, especially for lawyers inexperienced in a particular area, as it requires a lot of domain knowledge. Our agreement scores between experts and non-experts validate this (Table 6 in Appendix E), as does the overall model performance on non-expert-tuned prompts (Table 2). Despite this, using our prompting strategies, we achieve an average of over 0.75 for GPT-4-Turbo and 0.71 for Llama3-70B.

5.4 Qualitative Analysis

A review of the qualitative notes provided by evaluators reveals important aspects of the model’s output. Evaluators report that the differences in the reformulated questions using GPT-4-Turbo and Llama3-70B with or without provisions is generally very slight. Frequently, the difference comes down to slight differences that makes one reformulation better or worse than the other. By contrast, Mistral-7B outputs unusable reformulations that are often either nonsensical or no different from the user’s original question.

We provide examples of model reformulations in Table 4, and describe a few notable examples in detail below, as well as some of the annotators’ takeaways from the qualitative evaluation.

Models sometimes provide reformulations for individuals they should not advise. When evaluating safety implications, domain experts should focus on hidden pitfalls that may not be obvious to a community member. In almost every instance we identify, a safety issue arose in a circumstance that is likely non-obvious to a community member. Notably, one question in our sample asked:

User Question: “Can I get sued at 15 for money?”

Every model answered the question; ideally, none of them should have. If deployed, even with extensive expert-tuning, any of the models would have provided a reformulated answer to a minor. This raises a host of problems and concerns, including potentially misguiding an impressionable member of the population without adult supervision.

Models sometimes demonstrate meaningful legal reasoning. Below, GPT-4-Turbo’s formulation displays promising legal reasoning:

User Question: “Is it legal to require a minimum number of hours worked per

Question	Reformulation	Comments from Evaluators
what kind of lawyer am i looking for. idaho bad car from a dealer	What specialization should a lawyer have to address issues related to purchasing a defective vehicle from a dealership in Idaho?	"the user didn't say defective, that has a specific legal meaning."
What option do we have to avoid bankruptcy as freight forwarder hit by per-diem charges for container storage?	What legal or financial strategies are available for a freight forwarding company to manage or mitigate per-diem charges for container storage to avoid bankruptcy?	"Lawyers may counsel clients beyond legal action, considering that legal action might not always be the best option. When mentioning financial strategies, the model correctly implies that non-legal options could be viable. See Model Rules of Professional Conduct Rule 2.1."

Table 4: Example questions and reformulations by LLMs with evaluator comments on effectiveness and safety.

year to keep an ownership percentage in a company?"

Reformulation: "Does the agreement or applicable law allow for stipulations on minimum work hours as a condition for maintaining an ownership stake in a company?"

Although subtle, the model seems to pick up on the user's question about a company's ownership, which may be provided for in an agreement.

Models sometimes use legal terms incorrectly in their reformulations. In some cases, the inappropriate use of legal terms of art caused for poor and potentially misleading reformulations. In this case, GPT-4-Turbo did the following:

User Question: "Using an online form for contracts but I cannot export clients' answers directly; can I paste their signatures..."

Reformulated Question: "Is it legally permissible to manually transfer or paste clients' electronic signatures from one document to another in the context of contract formation?"

The major issue here is that *contract formation* is a **specific** legal issue, requiring a specific analysis. The user's question did not hint that the issue had to do with contract formation. This output is misleading, and an untrained eye could miss it.

Models sometimes reformulate questions requiring immediate legal attention. The annotators observed that all models answered sensitive questions that warrant a particular, strong response from an attorney, without informing users of the need to take urgent legal action. For example:

User Question: "What are the steps to filing a civil suit in Sonoma County for a quarter of a million by October 2021?"

The inclusion of a date is particularly important in the eyes of an attorney. Law suits are subject to **statutes of limitations**: laws that prevent a lawsuit from being filed after a certain time period has passed following an event that could trigger legal rights, entitlements, etc. The time constraints are harsh; once the specified time period has elapsed, only extraordinary circumstances can remedy it. When presented this question, the appropriate response is to immediately direct the individual to consult an attorney as soon as possible, taking action in real time. If a user poses such a question, and there is a delay in answering, legal rights may be lost. The key takeaway is that the model itself did not pick up on this important nuance. It may be possible to describe some of these nuances explicitly in the prompt. However, it might not be feasible to prompt a model to account for every instance when immediate action is required.

Models sometimes inject unnecessary jargon into their reformulations. Llama3-70B sometimes outputted unnecessary legal jargon:

User Question: "Can a consultant sign documents on behalf of their client?"

Reformulation: "What are the requirements for an agent or representative to have authority to bind their principal in a contractual agreement?"

The annotators noted that the question asked by the user is likely sufficient on its own. Thus, the model ignored the prompt instructions stating that acceptable questions could go un-reformulated. Second, when the model reformulated the question, it unnecessarily added the terms "principal" and "agent". These are very specific words that would likely confuse the user, and it's unclear from the user's question whether they are appropriate in this context. Qualitative evaluation indicates that Mistral-7B also tended to use legal jargon

inappropriately. Inappropriate legal jargon could influence a user to follow up with their own research on an incorrect issue, or misrepresent the nature of their circumstances to an attorney. If any of these models were used to perform such a task, serious safety concerns would arise.

6 Discussion

LLMs prompted using expert-formulated prompts produce substantially more high-quality reformulations than when prompted by non-experts Using LLM-as-a-judge, we evaluated the overall quality of our reformulations given the original user question. We found that baselines prompted using expert-written prompts performed over 4 times better than when prompted using non-expert-written prompts. This validates our prompting strategy, and underscores the importance of subject matter expertise when prompting LLMs. Users attempting to reformulate their own legal questions will likely struggle to obtain quality reformulations from LLMs without using prompts crafted by lawyers with subject matter expertise.

LLMs can correctly reformulate legal questions, but sometimes produce errors, particularly with legal correctness As we show in Table 3, GPT-4-Turbo demonstrates the ability to generate factual and coherent legal question reformulations, as does Llama3-70b to a slightly lesser degree. However, the highest score achieved by any of the models (GPT-4-Turbo) is 0.75 in the legal correctness category. Through manual inspection, we identified some common issues in LLMs’ reformulations, most notably the tendency to use unnecessary jargon or use legal terms incorrectly. We highlight these problems to guide future work, which may try to reduce these common errors in the effort to produce more high-quality reformulations.

The inclusion of linguistically-aligned language improves performance for open-source models, but not GPT For the two open-source models we tested, including linguistically-relevant provisions improved performance in all categories, with the exception of factuality for Mistral-7B. Including these provisions improved the performance of Llama3-70B (the best-performing open-source model) by 15 percentage points, bringing its legal correctness score close to GPT-4-Turbo’s highest score.

However, including linguistically-aligned lan-

guage reduced GPT’s performance in all three categories. When performing manual inspection, the lawyers on our team found that GPT-4-Turbo often produced very similar reformulations with and without provisions included. We posit that this may indicate a lack of sensitivity to user instruction.

Incorrect reformulations and inappropriately generated reformulations prevent real-world deployment for this task Given the potential real-world consequences, a 75% success rate is not high enough to recommend that these methods are deployed at a large scale to non-expert users. Underscoring this are our findings that non-expert users struggle to evaluate the legal correctness of a reformulated question, so it is likely that many users without domain expertise will have difficulty evaluating legal correctness in reformulations.

We also isolated crucial safety concerns: reformulations provided when they should not be. We found cases where LLMs provided reformulations for individuals who identified as minors, even though domain experts on our team agreed that this is unethical and could end up misleading minors to take actions without adult supervision. Further, all models provided reformulations in situations that require immediate legal attention (e.g. a statute of limitations), but did not inform the user that they need to take action.

7 Conclusion

Knowing the right questions to ask is key for successful interactions with lawyers. Using our dataset of user questions and expert annotations, we explore the extent to which LLMs can reformulate legal questions from non-experts. We find that LLMs show reasonable performance in reformulating non-expert questions to more clearly communicate potential issues (**RQ1**), but even the strongest model is legally incorrect in $\frac{1}{4}$ of our instances. We find that the inclusion of linguistically relevant language in the prompts strongly improves the performance of open source models, but reduces performance for GPT-4-Turbo (**RQ2**). Lastly, we isolate significant safety concerns, including questions posed by minors or involving statutes of limitations that are about to expire (**RQ3**). From this analysis, we learn that there are significant barriers to safety before these models can be deployed in this setting, including the possibility of incorrect reformulations or reformulations that shouldn’t be given.

8 Limitations

In terms of accessibility of the framework to groups that may have limited resources, there is presently a monetary cost associated with using ChatGPT via API. Although annotation efforts of the scale needed for finetuning an LLM would likely be more expensive, cost may be a barrier to some groups. Moreover, we only use zero-shot methods, which come with the benefit that resources needed for fine-tuning or other methods are not required; however, performance may not be optimal.

9 Ethical Considerations

This paper describes a research prototype. The findings described herein do not in any conceivable way provide any kind of legal advice, opinion, or otherwise. This research is not intended to solicit an attorney client relationship, is not an advertisement for legal services, and does not create an attorney client relationship. The questions relied upon to annotate data are publicly available. We do not work with sensitive or personal data. Human subjects participated voluntarily, undertook minimal risk, and were compensated fairly for their time.

The proposed framework relies on pretrained models such as word or image embeddings, so any harm or bias associated with these models may be present in our model. We believe general methods that propose to mitigate harms can resolve these issues.

We believe that studying how well LLMs perform legal tasks is crucial, given that they are so ubiquitous and easily accessible. However, as we mention in our discussion and conclusion section, we do not advocate for the use or deployment of LLMs for this task in real-world settings, as none of the state-of-the-art models studied were able to consistently produce legally correct reformulations. Caution should be taken whenever LLMs are used in the legal domain, particularly by non-experts, who we have shown have difficulty evaluating the correctness of the model output for our task.

References

2024. [When Do I Need a Lawyer?](#) [Online; accessed 17. Sep. 2024].
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Yongsu Ahn, Eliana Beigel, Noah Braun, Collin Griffin, Sera Linardi, Blair Mickles, and Emmaline Rial. 2022. [Improving citizen-initiated police reform efforts through interactive design: A case study in allegheny county](#). In *Proceedings of the 2nd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, EAAMO '22, New York, NY, USA. Association for Computing Machinery.
- L.K. Branting. 2017. [Data-centric and logic-based models for automated legal problem solving](#). *Artif Intell Law*, 25:5–27.
- Susan Bryant. 2001. The five habits: building cross-culture competence in lawyers. *Clinical L. Rev.*, 8:33.
- Inyoung Cheong, King Xia, KJ Kevin Feng, Quan Ze Chen, and Amy X Zhang. 2024. (a) i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2454–2469.
- Kiel Christianson. 2016. When language comprehension goes wrong for the right reasons: Good-enough, underspecified, or shallow language processing. *Quarterly journal of experimental psychology*, 69(5):817–828.
- Zewei Chu, Mingda Chen, Jing Chen, Miaosen Wang, Kevin Gimpel, Manaal Faruqui, and Xiance Si. 2020. How to ask better questions? a large-scale multi-domain dataset for rewriting ill-formed questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 05, pages 7586–7593.
- Jeremy R Cole, Michael JQ Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions. *arXiv preprint arXiv:2305.14613*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.
- Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 798–803.
- Haomin Fu, Yeqin Zhang, Haiyang Yu, Jian Sun, Fei Huang, Luo Si, Yongbin Li, and Cam-Tu Nguyen. 2022. Doc2bot: Accessing heterogeneous documents via conversational bots. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1820–1836.
- Morgan Gray, Jaromir Savelka, Kevin Ashley, and Wesley Oliver. 2023. [Can gpt alleviate the burden of annotation?](#) In *Legal Knowledge and Information Systems*, pages 157 – 166. IOS Press.
- Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.

Qing Huang, Zishuai Li, Zhenchang Xing, Zhengkang Zuo, Xin Peng, Xiwei Xu, and Qinghua Lu. 2023. Answering uncertain, under-specified api queries assisted by knowledge-aware human-ai dialogue. *arXiv preprint arXiv:2304.14163*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Sayash Kapoor, Peter Henderson, and Arvind Narayanan. 2024. Promises and pitfalls of artificial intelligence for legal applications. *Journal of Cross-disciplinary Research in Computational Law*, 2(2).

Wendy Muchman. 2023. [The effective lawyer: Communication, cultural competence, and civility](#).

Jaromir Savelka, Kevin Ashley, Morgan Gray, Hannes Westermann, and Huihui Xu. 2023. Can gpt-4 support analysis of textual data in tasks requiring highly specialized domain expertise? In *Automatic Semantic Analysis of Information in Legal Text (ASAIL)*.

Arunprasath Shankar and Venkata Nagaraju Buddarapu. 2019a. [Legal query reformulation using deep learning](#). In *Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Texts co-located with the 17th International Conference on Artificial Intelligence and Law (ICAIL 2019), Montreal, QC, Canada, June 21, 2019*, volume 2385 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Arunprasath Shankar and Venkata Nagaraju Buddarapu. 2019b. [Neural attention learning for legal query reformulation](#). In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ICAIL '19*, page 272–273, New York, NY, USA. Association for Computing Machinery.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. [A wrong answer or a wrong question? an intricate relationship between question reformulation and answer selection in conversational question answering](#). In *Proceedings of the 5th International Workshop on Search-Oriented Conversational AI (SCAI)*, pages 7–16, Online. Association for Computational Linguistics.

N. Wongpakaran, T. Wongpakaran, D. Wedding, and K. Gwet. 2013. [A comparison of cohen’s kappa and gwet’s ac1 when calculating inter-rater reliability coefficients: a study conducted with personality disorder samples](#). *BMC Med Res Methodol*.

S. Zec, N. Soriani, R. Comoretto, and I. Baldi. 2017. [High agreement and high prevalence: The paradox of cohen’s kappa](#). *The Open Nursing Journal*, pages 221–218.

A Justia Ask a Lawyer Question Distribution

Figure 3 provides some insight into our dataset of 274 Justia Ask a Lawyer questions related to business/commercial law. We plot the question lengths, as well as the years and states associated with the question.

B Relevant Clause Examples

See Table 5 for examples of linguistically-aligned clauses for sample questions.

C Prompt Templates

C.1 GPT

With contract provisions In Figure 4, we provide the template that was provided to the language model to study the effects of including relevant contract provisions into the prompt as legal background. {{USER QUESTION}} is replaced with the title of the question posted on Justia, and {{CONTRACT PROVISIONS}} is replaced with the top 10 most similar provisions to the question based on the cosine similarity score.

Without contract provisions In Figure 4, we provide the template that was provided to GPT to study its performance when relevant contract provisions are not added to the prompt, alongside the template with contract provisions for comparison. {{USER QUESTION}} is replaced with the title of the question posted on Justia.

C.2 Llama/Mistral Prompts

With provisions The prompt template given to Llama and Mistral with instruction tags and with contract provisions can be found in Figure 5.

Without provisions The Llama/Mistral prompt template without provisions and with instruction tags can be found in Figure 5.

D Example: Open-Source Model Outputs With and Without Instruction Tags

Below, we illustrate the importance of adding instruction tags to our prompts for the open-source models we experimented with, Llama and Mistral. While the reformulation for the question below with instruction tags is coherent and addresses the prompt directly, the reformulation without instruction tags is nonsensical, with “can be” repeated for most of the reformulation.

BACKGROUND: We will be reformulating questions generated by non-lawyer users related to commercial law, business law, or similar areas.

To help you understand the appropriate language you should consider we are going to include provide some provisions from legal contracts that have to do with commercial law, business law or similar areas.

These are complex legal provisions that cover a broad range or commercial/business contracts.

Often, the user will ask a question that requires a number of other questions to be answered first. Rather than answering a question that is unspecified, it would be helpful for the user to have their question reformulated if necessary.

TASK: We take a user question and reformulate it to ask the first, specific question that will help them ultimately answer the legal question they have. It is vital that the question accurately captures and/or maintains any legal issue conveyed by the user.

INSTRUCTIONS:

Given the user's question, formulate the first, specified question that should be asked based on the question provided by the user.

1. To help you we have provided provisions from a contract based on their similarity to the user's question. We have provided you with the top results of the similarity assessment.
2. However, you are NOT to use the provisions to craft a response, you are to ONLY use them to help reformulate, if necessary the first, specific question that a user should ask based on their initial question. There is in no way a guarantee that the provisions are on point. This is vital to understand.
3. If you think the question does not need reformulating, you can simply return the question that was asked.

PROVISIONS:

{{CONTRACT PROVISIONS}}

EXAMPLES:

Example 1:
 User Question: Can I work somewhere else after I quit my job?
 Reformulated Question: Could the terms and conditions of my present employment prevent me from working somewhere else if I quit?

Example 2:
 User Question: What happens if the contract is thrown out?
 Reformulated Question: What are the potential consequences of a contract or parts thereof being invalidated by court?

Example 3:
 User Question: Who am I allowed to talk to about sensitive information?
 Reformulated Question: Does the contract speak to confidentiality of information or non-disclosure?

EXPECTED OUTPUT:
 The user's question was: [user question]
 It has been reformulated to say: [reformulated question]

TASK INFORMATION:
 The user's question is: {{USER QUESTION}}

GPT Prompt Template With Provisions

BACKGROUND: We will be reformulating questions generated by non-lawyer users related to commercial law, business law, or similar areas.

TASK: We take a user question and reformulate it to ask the first, specific question that will help them ultimately answer the legal question they have. It is vital that the question accurately captures and/or maintains any legal issue conveyed by the user.

INSTRUCTIONS:

Given the user's question, formulate the first, specified question that should be asked based on the question provided by the user.

1. If you think the question does not need reformulating, you can simply return the question that was asked.

EXAMPLES:

Example 1:
 User Question: Can I work somewhere else after I quit my job?
 Reformulated Question: Could the terms and conditions of my present employment prevent me from working somewhere else if I quit?

Example 2:
 User Question: What happens if the contract is thrown out?
 Reformulated Question: What are the potential consequences of a contract or parts thereof being invalidated by court?

Example 3:
 User Question: Who am I allowed to talk to about sensitive information?
 Reformulated Question: Does the contract speak to confidentiality of information or non-disclosure?

EXPECTED OUTPUT:
 The user's question was: [user question]
 It has been reformulated to say: [reformulated question]

TASK INFORMATION:
 The user's question is: {{USER QUESTION}}

GPT Prompt Template Without Provisions

Figure 4: Prompt templates given to GPT models with (left) and without (right) contract provisions to generate question reformulations.

```

<=>[INST]
BACKGROUND: We will be reformulating questions generated by non-lawyer users related
to commercial law, business law, or similar areas.

To help you understand the appropriate language you should consider we are going to
include provide some provisions from legal contracts that
have to do with commercial law, business law or similar areas.

These are complex legal provisions that cover a broad range or commercial/business
contracts.

Often, the user will ask a question that requires a number of other questions to be answered
first. Rather than answering a question
that is unspecified, it would be helpful for the user to have their question reformulated if
necessary.

TASK: We take a user question and reformulate it to ask the first, specific question that will
help them ultimately answer the legal
question they have. It is vital that the question accurately captures and/or maintains any
legal issue conveyed by the user.

INSTRUCTIONS:

Given the user's question, formulate the first, specified question that should be asked based
on the question provided by the user.
1. To help you we have provided provisions from a contract based on their similarity to the
user's question. We have provided you with the top results of the similarity assessment.
2. However, you are NOT to use the provisions to craft a response, you are to ONLY use
them to help reformulate, if necessary the first, specific question that a user should ask
based on their initial question. There is in no way a guarantee that the provisions are on
point. This is vital to understand.
3. If you think the question does not need reformulating, you can simply return the question
that was asked.

PROVISIONS:

{{CONTRACT PROVISIONS}}

EXAMPLES:

Example 1:
User Question: Can I work somewhere else after I quit my job?
Reformulated Question: Could the terms and conditions of my present employment prevent
me from working somewhere else if I quit?

Example 2:
User Question: What happens if the contract is thrown out?
Reformulated Question: What are the potential consequences of a contract or parts thereof
being invalidated by court?

Example 3:
User Question: Who am I allowed to talk to about sensitive information?
Reformulated Question: Does the contract speak to confidentiality of information or non-
disclosure?

EXPECTED OUTPUT:
To the provided user question respond with a single reformulation and nothing else. Return
the following output:

{
"user_question": "<Example_user_question>", \
"reformulation": "<Example_user_question_reformulation>" \
}

TASK:
The user's question is: {{USER QUESTION}} [/INST]

```

Llama/Mistral Prompt Template With Provisions

```

<=>[INST]
BACKGROUND: We will be reformulating questions generated by non-lawyer
users related to commercial law, business law, or similar areas.

TASK: We take a user question and reformulate it to ask the first, specific
question that will help them ultimately answer the legal
question they have. It is vital that the question accurately captures and/or
maintains any legal issue conveyed by the user.

INSTRUCTIONS:

Given the user's question, formulate the first, specified question that should be
asked based on the question provided by the user.
1. If you think the question does not need reformulating, you can simply return
the question that was asked.

EXAMPLES:

Example 1:
User Question: Can I work somewhere else after I quit my job?
Reformulated Question: Could the terms and conditions of my present
employment prevent me from working somewhere else if I quit?

Example 2:
User Question: What happens if the contract is thrown out?
Reformulated Question: What are the potential consequences of a contract or
parts thereof being invalidated by court?

Example 3:
User Question: Who am I allowed to talk to about sensitive information?
Reformulated Question: Does the contract speak to confidentiality of
information or non-disclosure?

EXPECTED OUTPUT:
To the provided user question respond with a single reformulation and nothing
else. Return the following output:

{
"user_question": "<Example_user_question>", \
"reformulation": "<Example_user_question_reformulation>" \
}

TASK:
The user's question is: \{{USER QUESTION}\} [/INST]

```

Llama/Mistral Prompt Template Without Provisions

Figure 5: Prompt templates given to Llama and Mistral with (left) and without (right) contract provisions to generate question reformulations. Unlike the prompts given to GPT, these prompts contain explicit instruction tags.

Model	Between experts			Between experts and non-experts		
	Cohen's κ	% Agree	Gwet's AC1	Cohen's κ	% Agree	Gwet's AC1
GPT-4-Turbo	0.25	71%	0.52	-0.21	65%	0.51
Llama3-70B	0.32	67%	0.41	0.16	60%	0.20
Mistral-7B	0.16	66%	0.42	0.08	55%	0.12

Table 6: Agreement between expert annotators (left) and expert and non-experts (right) as to the legal correctness of the output. We include Gwet's AC1 in our agreement scores because Cohen's κ can be sensitive to class imbalances. We find moderate agreement amongst experts for legal correctness, and observe much lower agreement between experts and non-experts

When Valid Signals Fail: Regime Boundaries Between LLM Features and RL Trading Policies

Zhengzhe Yang

Independent Researcher

zhengzhe.yang@outlook.com

Abstract

Can large language models (LLMs) generate continuous numerical features that improve reinforcement learning (RL) trading agents? We build a modular pipeline where a frozen LLM serves as a stateless feature extractor, transforming unstructured daily news and filings into a fixed-dimensional vector consumed by a downstream PPO agent. We introduce an automated prompt-optimization loop that treats the extraction prompt as a discrete hyperparameter and tunes it directly against the Information Coefficient—the Spearman rank correlation between predicted and realized returns—rather than text-classification objectives. The optimized prompt discovers genuinely predictive features (IC above 0.15 on held-out data). However, these valid intermediate representations do not automatically translate into downstream task performance: during a distribution shift caused by a macroeconomic shock, LLM-derived features add noise, and the augmented agent under-performs a price-only baseline. In a calmer test regime the agent recovers, yet macroeconomic state variables remain the most robust driver of policy improvement. Our findings highlight a gap between feature-level validity and policy-level robustness that parallels known challenges in transfer learning under distribution shift.

1 Introduction

Recent work increasingly applies large language models to financial decision-making, whether as end-to-end trading agents or as modular sentiment classifiers feeding downstream models. While these modular pipelines separate the language model from the trading algorithm, they frequently suffer from an *objective mismatch*: the LLM is

optimized against standard text-classification objectives (e.g., cross-entropy on sentiment polarity or topic labels) rather than downstream financial utility. Consequently, it remains difficult to guarantee that the extracted narratives form a robust state representation for a continuous trading policy.

We address this gap by maintaining strict architectural separation while directly aligning the feature extraction process with a financial objective. The frozen LLM acts as a *stateless feature extractor*: given a bundle of news articles and SEC filings for a ticker on day d , it emits a fixed-length numerical vector (sentiment, impact, conflict flags, etc.). A separate PPO agent then consumes this vector alongside price data and macroeconomic indicators to make portfolio decisions. This design ensures that the intermediate representations are genuinely predictive, allowing us to evaluate the LLM’s true contribution in isolation.

Our contributions are:

- Prompt-as-hyperparameter optimization.** We introduce a mutation–evaluation–selection loop that treats the LLM extraction prompt as a discrete hyperparameter and optimizes it against the Information Coefficient (IC)—the rank correlation between predicted and realized returns—rather than text-classification objectives like BLEU or accuracy. The winning prompt improves IC from -0.024 to $+0.104$ (Table 2).
- Feature-validity-to-policy-utility gap.** We show that valid intermediate LLM representations do not automatically translate into downstream RL performance. The gap is regime-dependent: under distribution shift caused by a macroeconomic shock, news-derived features add noise rather than signal.

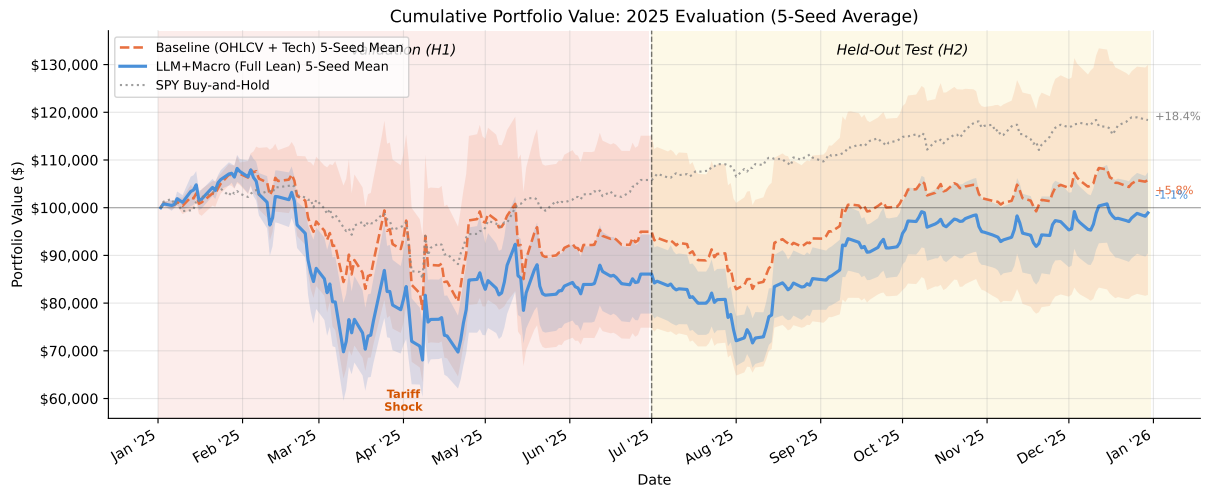


Figure 1: Cumulative portfolio value across 2025. Solid lines: 5-seed mean; shaded: ± 1 std. H1 (red) captures a tariff-driven shock; H2 (yellow) is calmer. The regime split isolates when LLM features succeed and fail. SPY buy-and-hold outperforms all RL configurations during the 2025 bull rally; the contribution is the *relative* comparison across configurations under controlled distribution shift, not absolute excess return.

3. **Multi-regime ablation on held-out data.** A controlled four-configuration ablation (Baseline, LLM-only, Macro-only, LLM+Macro) across a volatile validation period (H1 2025) and a calmer held-out test period (H2 2025) reveals that macroeconomic state variables are the most reliable driver of policy robustness (Figure 1).

2 Related Work

Reinforcement Learning for Trading. Deep RL has been applied to portfolio management and order execution with increasing sophistication (Hamblly et al., 2023). FinRL (Liu et al., 2020) provides a standardized library for training PPO, A2C, and DDPG agents on market environments. A persistent challenge is non-stationarity: the data distribution shifts between training and deployment, a problem well-studied in the broader RL literature as distributional shift (Kumar et al., 2020). In finance, these shifts are driven by macroeconomic regime changes (Ang and Bekaert, 2002), and recent work has explored conditioning RL policies on detected regimes (Sun et al., 2023).

LLMs in Finance. BloombergGPT (Wu et al., 2023) demonstrated that domain-specific pre-training improves financial NLP benchmarks. FinGPT (Yang et al., 2023) pursued the same goal with open-source fine-tuning. Lopez-Lira and Tang (2023) demonstrated that frontier models can forecast subsequent-day stock returns using raw

headline text, while later approaches instruction-tune open source variants directly against financial tasks (Zhang et al., 2023). However, these lines of work overwhelmingly evaluate the LLM as a standalone classifier (sentiment polarity, NER, QA). By contrast, we treat the LLM as a frozen, zero-shot *numerical* feature extractor—analogueous to using a pre-trained vision model as a fixed encoder for downstream tasks—and evaluate its representations against a continuous downstream RL objective rather than NLP classification metrics.

Prompt & Pipeline Optimization. Frameworks like DSPy (Khattab et al., 2023) formally compile language model calls by optimizing prompts against programmatic validation metrics. While these frameworks routinely optimize for exact-match accuracy or retrieval scores, our optimization loop extends this paradigm to a domain-specific continuous metric (rank correlation of predicted returns) prior to RL integration, treating the prompt as a discrete hyperparameter in the same spirit as architecture search.

Information Asymmetry and News Latency. Insider-trading research (Seyhun, 1998) established that information edges decay rapidly as they become public. For free-tier news feeds, institutional desks have already acted on the headline by the time a retail pipeline ingests it. This latency shapes the horizon at which LLM-derived features can carry signal—a constraint we quantify in Section 5.1.

3 System Architecture

3.1 Data Ingestion Pipeline

Reproducible feature extraction requires a deterministic historical record. We built a concurrent Go pipeline that ingests: (1) news from Alpaca’s Benzinga feed, (2) RSS aggregations from financial outlets, and (3) SEC EDGAR filings (Form 4 insider trades and 8-K disclosures). Raw text is bundled per ticker per trading day and persisted to a SQLite database (the “backfill layer”).

This backfill-first design prevents look-ahead bias: the LLM always reads from a frozen snapshot whose information boundary is strictly \leq day d . It also allows the prompt-optimization loop (Section 4) to re-extract features from identical text without re-scraping.

Because our pipeline enforces strict daily information boundaries (aggregating all feeds at the close of day d), it is intentionally blind to intra-day volatility and high-frequency market microstructure. Consequently, the RL agent operates on smoothed, day-over-day narrative shifts rather than instantaneous headline shocks. We measure the effect of this daily resolution in Section 5.1.

3.2 Feature Schema

A frontier LLM (Qwen3 235B A22B Instruct 2507) processes each ticker’s daily bundle and outputs a structured JSON mapping into continuous RL observation bounds:

1. Stock-level LLM features (4 dims):

- `sentiment` $\in [-1, 1]$: Directional conviction from the daily news flow (-1 : very bearish, $+1$: very bullish).
- `impact` $\in [0, 1]$: Financial materiality of the bundle (e.g., CEO resignation vs. routine marketing).
- `conflicting_signals` $\in [0, 1]$: Evidentiary contradiction across competing sources within the same bundle.
- `news_novelty` $\in [0, 1]$: Divergence of the current day’s narrative from historical baselines.

2. Macroeconomic features (5 dims):

`vix` (market anxiety), `treasury_10y` (discount rate proxy), and `credit_spread` (corporate default risk), sourced from FRED. Two additional LLM-inferred regime flags (`market_sentiment`,

`macro_event_flag`) complement the systematic landscape.

This JSON constraint grounds high-dimensional linguistic narratives into an explicit 9-dimensional vector digestible by the downstream MLP. Table 1 details the summary statistics.

3.3 RL Policy Agent

We use FinRL (Liu et al., 2020) to construct the trading environment and PPO agent. The composite state vector is:

$$S_t = [P_t \parallel \mathcal{E}_t \parallel M_t \parallel B_t] \quad (1)$$

where P_t is OHLCV (open, high, low, close, volume) bars plus technical indicators, \mathcal{E}_t is ticker-level LLM features, M_t is macro features, and B_t is portfolio state. Observations are normalized via `VecNormalize(clip_obs = 10)`.

The agent is a PPO with an MLP policy, trained for 500k timesteps on 2023–2024 data—a cutoff supported by the convergence analysis in Figure 3. The RL framework trades a 21-ticker universe ($\sim 10,500$ rows).

All results are averaged across five seeds $\{0, 1, 2, 3, 42\}$ with deterministic hierarchical seeding (PyTorch `manual_seed + SB3` environment seeds) to ensure reproducibility.

4 Prompt Optimization

Standard prompt engineering does not scale when the downstream task is continuous-valued RL rather than classification. Standard text-quality metrics like BLEU or classification accuracy do not measure whether extracted features rank future returns correctly. We designed an automated optimization loop that treats the extraction prompt as a discrete hyperparameter and tunes it against the Information Coefficient (IC).

The pipeline operates via a feedback loop with Anthropic’s Claude API as meta-optimizer. The workflow, illustrated in Figure 4, follows six steps:

1. **Initialize** a baseline chain-of-thought prompt (`v0`).
2. **Define gates:** IC (rank correlation stability), Hit% (directional accuracy), Quintile Spread (monotonicity of ranked portfolios). These measure whether the LLM’s numerical outputs predict future returns, unlike text-quality metrics (e.g., BLEU, classification accuracy) which only assess linguistic fidelity.

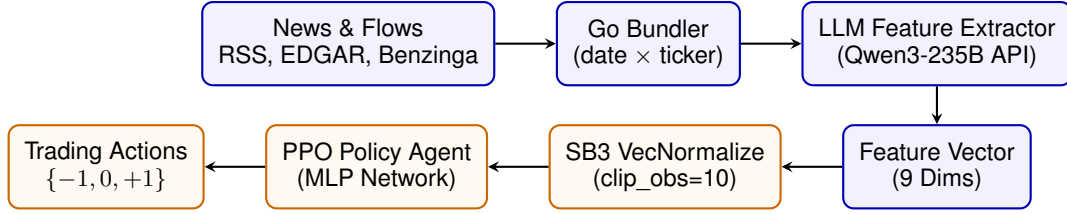


Figure 2: System overview. A Go-based ingestion pipeline collects news, filings, and macro data into a relational store. The LLM produces per-ticker feature vectors, which the PPO agent consumes alongside OHLCV (open, high, low, close, volume) bars and technical indicators.

Table 1: Feature Value Distributions (2023–2024 training period)

Feature	Group	Mean	Std	Min	Max	% Non-zero	Scale
sentiment	LLM News	-0.007	0.290	-1.000	0.900	71.8%	[-1, 1]
impact	LLM News	0.292	0.233	0.000	1.000	71.9%	[0, 1]
conflicting_signals	LLM News	0.076	0.182	0.000	1.000	18.5%	[0, 1]
news_novelty	LLM News	0.737	0.424	0.000	1.000	77.9%	[0, 1]
market_sentiment	Market Regime	0.175	0.405	-1.000	0.800	98.7%	[-1, 1]
macro_event_flag	Market Regime	0.481	0.500	0.000	1.000	48.1%	{0, 1}
treasury_10y	Macro	4.085	0.370	3.300	4.980	100.0%	[3.5, 5.5]
vix	Macro	16.120	3.262	11.860	38.570	100.0%	[12, 80]
credit_spread	Macro	3.675	0.620	2.600	5.220	100.0%	[0.5, 3.0]

VecNormalize (clip_obs=10) applied during training normalizes these distributions.

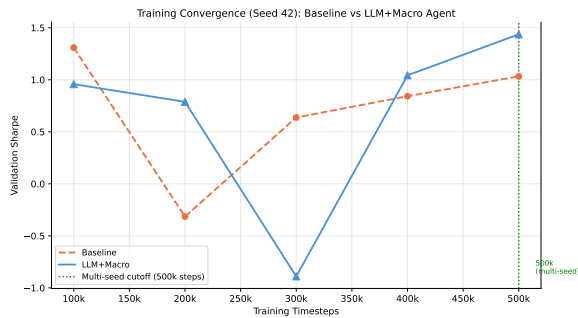


Figure 3: Training convergence (seed 42). Validation Sharpe at each 100k-step checkpoint on H1 2025. Both agents plateau by 400–500k steps; the vertical line marks the multi-seed cutoff.

3. **Meta-optimize:** Claude suggests a discrete structural mutation (e.g., adding few-shot examples, redefining output ranges).
4. **Extract:** Features for a one-month subset (January 2025, 769 bundles, 38 tickers) using the mutated prompt via Qwen3.
5. **Evaluate:** Compute IC gates.
6. **Iterate:** If gates fail, send metrics back to Claude as feedback; if passed, freeze the prompt.

Each of the five mutations targets a distinct

failure mode observed in pilot runs on the baseline prompt: semantic ambiguity (mut1), numerical mis-calibration (mut2), signal interference between news and flow modalities (mut3), recency-vs-surprise framing (mut4), and a composition test of whether gains stack (mut5). The taxonomy spans standard prompt-engineering levers—specification, calibration, decomposition, framing, composition—so the loop searches a structured mutation space rather than free-form rewrites. As shown in Table 2:

1. **mut1 (Impact-Surprise):** Redefines `impact` as the magnitude of market surprise relative to consensus.
2. **mut2 (Few-Shot):** Adds three concrete calibration examples (priced-in beat, genuine surprise, and conflicting flow).
3. **mut3 (Separate Reasoning):** Decouples reasoning into distinct news and flow signal analysis blocks.
4. **mut4 (Counterfactual):** Uses a "what-if" counterfactual test to anchor sentiment scores against market defaults.
5. **mut5 (Combined):** Merges mut1, mut2, and mut4 into a single prompt.

The results demonstrate that explicit few-shot

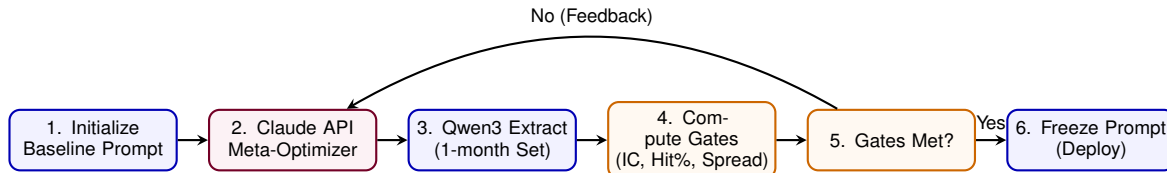


Figure 4: Prompt optimization workflow. Claude iteratively refines the extraction prompt based on downstream financial metric feedback.

Table 2: Prompt mutation results. Top: January 2025 optimization set (769 bundles, 38 tickers). Bottom: February 2025 out-of-sample validation.

(a) Jan 2025 — Optimization Set

Candidate	Hypothesis	IC IR	Hit%	Spread	Brier	Comp.
v3-baseline	Chain-of-thought	-0.024	45.5	-1.07%	0.305	-0.144
mut1	Impact = surprise	-0.075	22.2	-1.66%	0.291	-0.325
mut2-few-shot [†]	Few-shot examples	+0.104	71.4	+0.22%	0.288	+0.191
mut3-separate	Separate reasoning	+0.029	60.0	+0.75%	0.300	+0.134
mut4	Counterfactual	-0.098	57.1	-0.29%	0.289	+0.052
mut5-combined	mut1+2+4	-0.175	55.6	-2.06%	0.290	-0.243

[†]Frozen as v4-stable-core.

(b) Feb 2025 — OOS Validation

Candidate	IC IR	Hit%	Spread	Brier	Comp.
v3-baseline	-0.212	50.0	N/A	0.293	-0.210
mut2-few-shot	-0.044	25.0	N/A	0.281	-0.230

calibration (mut2-few-shot) was the critical driver of performance, improving IC IR to +0.104 compared to the baseline’s -0.024. In contrast, combining multiple structural changes (mut5) caused a performance collapse, likely due to instruction bloat and conflicting reasoning anchors. The prompt was frozen as v4-stable-core (Figure 5) and used for all subsequent extraction.

4.1 Adequacy Gates

Before committing to RL training, we evaluated the extracted features against four predictive adequacy gates (Table 3). The prompt cleared all four thresholds, notably exceeding the IC requirement by a substantial margin. Because our optimization loop explicitly targets downstream predictive utility, this strong rank correlation confirms the prompt’s validity in generating a continuous observation space for the downstream RL agent.

5 Empirical Evaluation

Training spans 2023–2024. Validation covers 2025 H1 (January–June, 120 trading days), a pe-

Table 3: Adequacy gate assessment (mut2-few-shot, Jan 2025).

Metric	Gate Threshold	Value	Status
signal_coverage	≥ 0.25	0.408	PASS
ic_ir_5d	≥ 0.05	+0.104	PASS
quintile_spread	> 0	+0.002	PASS
hit_rate	≥ 0.52	0.714	PASS

[†]Structural issue: pred_prob uses impact as confidence, but impact measures materiality, not prediction certainty.

riod of tariff-driven macro volatility. The held-out test covers 2025 H2 (July–December), a materially calmer regime.

5.1 Feature Validity

Before examining downstream RL performance, we verify that the optimized prompt produces features with genuine predictive signal. Table 4 reports the Information Coefficient (IC)—the Spearman rank correlation between each feature’s daily values and subsequent 5-day returns, averaged

Prompt v1-stable-core Feature Extraction Template

System: You are a quantitative feature extraction engine for an institutional trading system. Given an event bundle about a specific ticker, output ONLY valid JSON with numerical features...

User: Extract trading features for ticker `{{.Ticker}}` from the following event bundle on `{{.Date}}`. Consider ALL signal types together — news articles, insider trades, and options flow. Output ONLY valid JSON matching this schema.

Field definitions (fill in this order):

- `reasoning`: Write ONE sentence summarizing the key signal and WHY it moves the stock.
- `sentiment`: Predicted FUTURE PRICE TRAJECTORY over the next 1-5 trading days. [-1.0 strongly bearish, 0.0 no edge, +1.0 strongly bullish] Focus on SURPRISE vs CONSENSUS, not absolute tone.
- `impact`: Materiality of the news. [0.0 trivial, 1.0 highly market-moving].
- `conflicting_signals`: Do the signals point in contradictory directions? [0.0 aligned, 1.0 strongly contradictory].
- `insider_trading`: Insider BUYS are a moderately bullish signal. Insider SELLS are WEAK (usually scheduled 10b5-1 plans).

Calibration Example A — Priced-in beat: “AAPL reports Q4 earnings beating consensus by 2%, in line with whisper numbers.” → `sentiment`: +0.1, `impact`: 0.2, `conflicting_signals`: 0.0

Calibration Example B — Genuine surprise: “NVDA unexpectedly raises full-year guidance 40% above Street estimates...” → `sentiment`: +0.8, `impact`: 0.9, `conflicting_signals`: 0.1

Figure 5: Abbreviated visualization of v1-stable-core. Few-shot calibration examples anchoring numerical output ranges proved necessary to maximize IC.

across trading days. IC normalizes by the standard deviation of daily ICs, analogous to a signal-to-noise ratio. Among the LLM-derived features, `conflicting_signals` (IC = 0.233, $t = 2.52$) and `impact` (0.177, $t = 1.91$) carry the strongest signal.

The macro features (VIX, Treasury, credit spread) register IC ≈ 0 by construction: they are identical across all tickers on a given day, so cross-sectional rank correlation is undefined. Their value is purely time-serial—they tell the RL agent *when* to trade cautiously, not *which* ticker to favor. Figure 6 confirms this asymmetry using a gradient-boosted tree trained on 5-day forward returns as a model-agnostic surrogate for the observation space (the deployed MLP policy itself does not expose comparable feature attributions): the tree assigns 58% cumulative split-importance to macro features despite their zero cross-sectional IC.

Figure 7 shows sentiment IC as a function of forecast horizon. The signal is near zero at one day, peaks at 3–10 days, and decays by day 20—consistent with the delayed-news constraint: the 1-day edge has been captured by faster participants, leaving only medium-term narrative drift.

5.2 Validation: Macro-Shock Regime (H1 2025)

Cross-sectional feature evaluation is conducted on a broader 38-ticker US large-cap signal universe, while the downstream RL environment restricts execution to a 21-ticker liquid trading subset plus SPY;

the exact ticker lists are provided in Appendix A.

Given that the LLM features carry genuine signal, we now ask whether this translates into downstream RL performance. Table 5 presents the four-configuration ablation on H1 2025. No configuration significantly outperforms the price-only baseline (all paired t -test $p > 0.1$).

The pattern is informative. LLM-only is the worst configuration (Sharpe -0.411): trading on idiosyncratic stock news during a systemic shock amounts to ignoring the dominant risk factor. Macro-only (-0.007) tracks the baseline closely. LLM+Macro (-0.267) is better than LLM-only because the macro features provide a “regime brake,” but the noisy LLM signals still drag it below baseline. Figure 8 visualizes this: during elevated VIX, the LLM-augmented agent systematically underperforms.

5.3 Held-Out Test: Calm Regime (H2 2025)

If idiosyncratic news fails only when macro risk dominates, a calmer regime should restore its value. We locked all model parameters and evaluated once on H2 2025 (Table 6).

All three augmented configurations now exceed the baseline (Sharpe 0.809). LLM-only recovers to 1.001, suggesting that stock-level narratives can capture signal when systemic risk subsides. However, Macro-only remains strongest (1.099, $\Delta = +0.290$), and LLM+Macro (1.038, $p = 0.49$ vs. baseline) does not reach statistical significance at $N = 5$ seeds. SPY buy-and-hold (1.756) outper-

Table 4: Feature IC analysis (5-day forward return, 2023–2024 training period). Cross-sectional IC for macro features is zero by construction: they are constant across tickers on any given day and carry regime information detectable only by the RL policy, not by cross-sectional ranking.

Feature	Group	IC Mean	IC IR	t -stat	% Pos	N
sentiment	LLM News	0.016	0.093	1.00	0.5	117
impact	LLM News	0.029	0.177	1.91	0.6	117
conflicting_signals	LLM News	0.040	0.233	2.52	0.6	117
news_novelty	LLM News	0.011	0.065	0.70	0.5	117

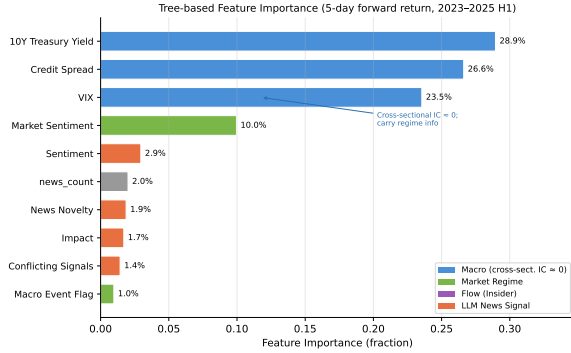


Figure 6: Feature importance from a gradient-boosted tree on 5-day forward returns. Macro features dominate (credit_spread 28%, VIX 16%, treasury 14%) despite zero cross-sectional IC.

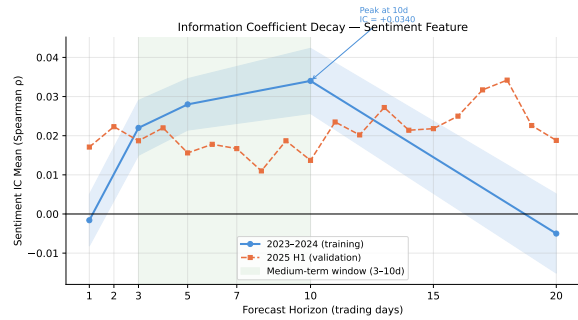


Figure 7: IC decay for sentiment across horizons. Signal peaks at 3–10 days, consistent with the delayed-news constraint.

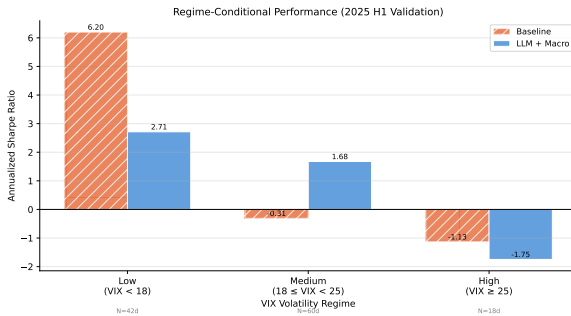


Figure 8: Regime-conditional performance. During elevated VIX (shaded), the LLM-augmented agent underperforms the baseline. In low-volatility windows the gap narrows or reverses.

forms all RL configurations during the late-2025 bull rally; our contribution is the *relative* ablation, not absolute excess return.

5.4 Robustness Checks

Transaction cost sensitivity. Table 7 varies transaction costs from 0 to 50 bp for seed 42 on H1 2025. The baseline’s advantage over LLM+Macro is stable across all cost levels, confirming that the H1 null result is not an artifact of unrealistic friction assumptions.

6 Discussion

The feature-to-policy gap. Our results expose a gap between intermediate representation quality and downstream task performance—a phenomenon familiar from transfer learning, where strong pre-trained features do not guarantee fine-tuning success under distribution shift. The prompt optimization loop successfully produces features with high rank correlation to future returns. However, this signal only translates into RL policy improvement when the test distribution resembles training conditions. When a macroeconomic shock shifts the data distribution (H1 2025), the RL agent cannot exploit features whose predictive structure has changed.

Macro features as a regime brake. VIX, Treasury yields, and credit spreads do not predict *which* stocks will outperform—their cross-sectional IC is zero (Table 4). Instead, they tell the RL agent when the market environment has shifted, allowing it to reduce exposure. Using a gradient-boosted tree on the same observation space as a model-agnostic proxy (the MLP policy does not expose comparable attributions), macro inputs receive 58% of split-importance (Figure 6); the ordering is consistent with the ablations in Tables 5–6. This time-serial

Table 5: H1 2025 validation ablation ($N = 5$ seeds, 120 trading days). Mean \pm std across seeds. Δ Sharpe vs. Baseline.

Config	N Feats	Sharpe	Return%	Max DD%	Δ Sharpe
Baseline (OHLCV + Tech)	0	0.010 \pm 0.618	-5.03 \pm 18.00	34.91 \pm 6.01	—
LLM Signals Only [†]	6	-0.411 \pm 0.690	-16.42 \pm 18.81	42.12 \pm 8.11	-0.421
Macro Only [‡]	5	-0.007 \pm 0.355	-6.16 \pm 9.74	34.78 \pm 9.28	-0.017
LLM + Macro (Full)	10	-0.267 \pm 0.284	-13.91 \pm 7.41	38.66 \pm 6.34	-0.276

[†] LLM-only: sentiment, impact, conflicting_signals, news_novelty + 2 regime flags. [‡] Macro-only: VIX, treasury_10y, credit_spread + 2 regime flags. No config beats Baseline ($p > 0.1$, paired t , $N = 5$).

Table 6: H2 2025 held-out test ($N = 5$ seeds). Mean \pm std. Δ Sharpe vs. Baseline.

Config	N Feats	Sharpe	Return%	Max DD%	Δ Sharpe
Baseline (OHLCV + Tech)	0	0.809 \pm 0.333	11.29 \pm 6.21	14.96 \pm 4.04	—
LLM Signals Only [†]	5	1.001 \pm 0.853	12.98 \pm 8.93	16.27 \pm 6.59	+0.192
Macro Only [‡]	5	1.099 \pm 0.695	16.04 \pm 9.53	15.30 \pm 3.15	+0.290
LLM + Macro (Full)	10	1.038 \pm 0.424	15.14 \pm 6.19	17.27 \pm 2.18	+0.229
SPY buy-and-hold [§]	—	1.756 [§]	11.87 [§]	5.07 [§]	+0.947

Paired t -test (LLM+Macro vs. Baseline): $t(4) = 0.76$, $p = 0.4873$

^{†/‡} Feature groups as in Table 5. [§] SPY buy-and-hold (single value). Models fixed before unlocking test set.

conditioning is invisible to cross-sectional IC.

Information latency as a feature constraint.

Our reliance on free-tier news feeds bounds the temporal resolution of extractable signal. The IC-decay curve (Figure 7) shows that predictive power peaks at 3–10 days. From an ML perspective, this is an input-quality constraint analogous to training on low-resolution images—the representation is valid but resolution-limited.

Limitations.

- **Statistical power.** Five seeds on a 120-day window yield low power (<50% to detect Δ Sharpe=0.3 at $\sigma = 0.4$).
- **Absolute performance.** All agents underperform SPY buy-and-hold. The contribution is the relative ablation, not absolute return.
- **Narrow universe.** 21 large-cap US equities. Generalization is an open question.
- **Input resolution.** A low-latency news feed may restore short-horizon signal.

7 Conclusion

We present an automated prompt-optimization pipeline that tunes a frozen LLM to produce predictive numerical features for an RL trading agent. Valid intermediate representations do not automatically yield downstream improvement: the

gap between feature quality and policy robustness is regime-dependent—LLM-derived features help when the test distribution is stable but degrade under macroeconomic shocks. Evaluating LLM-generated features on intermediate metrics alone is insufficient; multi-regime out-of-sample testing should become standard for any pipeline feeding LLM representations into a downstream learner.

A Ticker Universes

Signal universe (38 tickers). LLM feature extraction and prompt-optimization experiments use the following 38-ticker signal universe:

AAPL, ABBV, ADBE, AMD, AMZN, AVGO, BA, BAC, CAT, COST, CRM, CVX, GE, GOOGL, GS, HD, INTC, IWM, JNJ, JPM, LLY, MA, MCD, META, MSFT, NFLX, NKE, NVDA, ORCL, QCOM, QQQ, RTX, SPY, TSLA, UNH, V, WMT, XOM.

Tradable RL universe (21 tickers). The downstream RL environment restricts execution to the following 21-ticker liquid trading universe:

AAPL, MSFT, AMZN, NVDA, META, TSLA, AMD, NFLX, ADBE, QCOM, JPM, V, MA, GS, UNH, LLY, XOM, WMT, BA, CAT, SPY.

Why the universes differ. The broader 38-ticker universe improves cross-sectional feature evaluation and includes contextual instruments such as

Table 7: Transaction cost sensitivity (2025 H1, seed 42). The baseline advantage over LLM+Macro is robust across all cost levels, confirming the H1 null result is not a friction artifact.

Cost (bp)	Cost (%)	Baseline Sharpe	LLM+Macro Sharpe	Δ Sharpe	LLM Win?
0	0.00%	1.231	0.753	-0.478	×
5	0.05%	1.202	0.730	-0.472	×
10	0.10%	1.156	0.674	-0.482	×
20	0.20%	1.108	0.549	-0.559	×
50	0.50%	0.929	0.580	-0.349	×

Δ Sharpe = LLM Sharpe - Baseline Sharpe. Standard cost (0.10%) highlighted.

QQQ, IWM, and GOOGL. The RL agent trades only the 21-ticker liquid subset. GOOGL is excluded from trading for compliance reasons, although its signals remain available in the upstream feature store.

References

Andrew Ang and Geert Bekaert. 2002. International asset allocation with regime shifts. *The Review of Financial Studies*, 15(4):1137–1187.

Ben Hambly, Ruiwei Xu, and Huining Yang. 2023. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, and 1 others. 2023. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. *arXiv preprint arXiv:2310.03714*.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191.

Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Linyi Yang, Bowen Xiao, and William Wang. 2020. Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance. *Deep RL Workshop, NeurIPS 2020*.

Alejandro Lopez-Lira and Yuehua Tang. 2023. Can chatgpt forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619*.

H Nejat Seyhun. 1998. *Investment intelligence from insider trading*. MIT press.

Hao Sun and 1 others. 2023. Market regime aware reinforcement learning for quantitative trading. *Proceedings of the ICAIF*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabrovolski, Mark Drozdov, Brad Mullis, Chenyu Yue, Steve Ostrum, and 1 others. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*.

Boyu Zhang and 1 others. 2023. Pixtral & finma: Instruct-finllm for financial domain. *arXiv preprint arXiv:2306.06031*.

Unintended Effects of Geographic Conditioning in Large Language Models

Naz Col, David M. Chan
University of California, Berkeley
{doganazcol, davidchan}@berkeley.edu

Abstract

Modern conversational AI systems frequently rely on user metadata to localize responses, yet the unintended regional biases introduced by this hidden context remain poorly understood. In this work, we evaluate *location leakage*: the phenomenon where a model generates geographic references despite receiving a geographically neutral user prompt. Across both creative writing and open-ended Q&A prompts, even state-of-the-art LLMs systematically favor region-specific outputs when exposed to location metadata, with leakage spiking by up to 793 times above baseline (e.g., from 0.04% to 31.7% for Llama 3.1-8B, and 21.3% and 8.8% for Qwen3-8B and Claude Sonnet 4.6, respectively). Our analysis further shows a novel structural conditioning effect: replacing the injected location with the placeholder "Unknown" still elevates leakage by up to 72 times above baseline, demonstrating that the user profile frame itself, independent of any geographic content, acts as a generative conditioning signal.

1 Introduction & Background

Large Language Models (LLMs) have become core engines for deployed conversational AI systems, transforming how users interact with information. To make these systems more locally context-aware, production pipelines frequently inject inference-time user metadata, such as geographic location, into system instructions or prompt headers. This conditioning ensures that localized queries return regionally relevant answers.

Unfortunately, existing geographic conditioning approaches come with several notable drawbacks. While explicitly providing user location helps ground geocentric queries, we observe that models often over-index on this metadata even when the underlying user prompt is entirely location-agnostic. We term this phenomenon *location leakage*: a latent interaction-layer risk where simple geographic conditioning forces regional references, cultural skews, or geographic stereotypes into open-ended generations that do not require them. While it has previ-

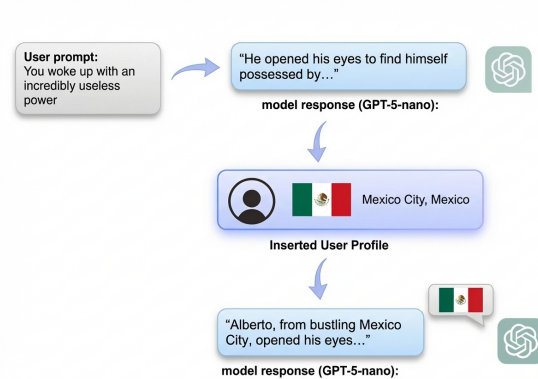


Figure 1: Injecting a location-specific user profile shifts model output from generic to geographically grounded, even when the user prompt is location-agnostic.

ously been shown that LLMs suffer from geographic bias, *i.e.* LLMs favor wealthier regions in geospatial prediction (Manvi et al., 2024), align with national narratives during historical events (Salnikov et al., 2025), and skew toward affluent areas in recommendations (Dudy et al., 2025), these works (Gallegos et al., 2024; Nadeem et al., 2021; Nangia et al., 2020; Bender et al., 2021; Gopinadh et al., 2026) all focus on *pre-training priors* and *implicit demographic inference*, rather than the explicit conditioning that is commonly used in deployed systems. Recently, Piot et al. (2025) have shown that this pre-trained geographic bias can be mitigated through fine-tuning in classification settings, but they do not investigate the open-ended generation setting where location leakage is most pronounced, and Jin et al. (2024) showed that models suffer from *implicit personalization* based on inferred user demographics, but do not study the explicit conditioning layer where location is directly injected at inference time.

To address these limitations, in this paper we introduce a framework designed to quantify issues with explicit inference-time geographic conditioning. We analyze three common deployment architectures: manual pre-pending of a structured user profile block, system prompt injection, and a dual-layered hybrid combination of both methods.

We explore the severity of location leakage by evaluating five language models (Qwen3-8B (Team, 2025), Llama3-8B-Instruct (Grattafiori et al., 2024), GPT-5-nano (OpenAI, 2025), Gemini 3 Flash (Google DeepMind, 2025), and Claude Sonnet 4.6 (Anthropic, 2025)) across two location-agnostic datasets: WritingPrompts (Fan et al., 2018) and Infinite Chats (Jiang, 2024). We show that injecting location data increases leakage by up to 793 times a baseline, with Llama 3.1-8B peaking at a 31.7% leakage rate in some cases. We also explore the structural versus semantic components of this conditioning, proving that the user profile frame alone acts as an independent signal that significantly amplifies leakage. Finally, to explore the underlying drivers of location leakage, we conduct a cross-correlation analysis against global socioeconomic indicators (Naz, 2023), finding that tertiary education enrollment ($\rho = -0.20, p < 0.01$) is a significant predictor of leakage rates.

We summarize our main contributions as follows: (1) We define and formalize the phenomenon of *location leakage*, and introduce a framework for measuring geographic conditioning in non-geocentric tasks, (2) We provide empirical evidence across five state-of-the-art models and three injection methods, demonstrating leakage rates up to 31.7%, and (3) We decompose location leakage into structural and semantic components, demonstrating that the user profile frame alone amplifies leakage up to 72 times over baseline models, and show that this vulnerability disproportionately impacts Oceania and North American locales.

2 Measuring Location Leakage

We formally define *location leakage* as a generative conditioning failure where a language model introduces geographic references into its output despite receiving a location-agnostic prompt.

Let $x \in \mathcal{X}$ be a geographically neutral prompt drawn from a distribution of location-agnostic tasks. Let $c \in \mathcal{C}$ denote an injected geographic context vector specifying a country loc (e.g., via a user profile or system prompt modification). A language model parameterized by θ generates a token sequence y according to $P_\theta(y|x,c)$.

Let $\mathbb{I}_{\text{leak}}(y,loc) \in \{0,1\}$ be an indicator function that outputs 1 if y contains an explicit geographic reference to loc (or its direct linguistic derivatives), and 0 otherwise. The baseline leakage rate λ_0 (intrinsic prior without geographic conditioning) and the conditioned leakage rate λ_c (with explicit

Manual Pre-pending Input

```

- BEGIN USER PROFILE -
Location: [Country Name]
- END USER PROFILE -

You woke up with an incredibly useless
power...

```

Figure 2: The model receives the geographic profile block and the writing prompt as a single combined input.

System Prompt

```

You are a helpful assistant for a
user in <location>. Be concise and
direct; avoid being generic.

```

Figure 3: System prompt used for location injection.

context) are defined as:

$$\lambda_0 = \mathbb{E}_{x \sim \mathcal{X}} [\mathbb{I}_{\text{leak}}(f_\theta(x), loc)] \quad (1)$$

$$\lambda_c = \mathbb{E}_{x \sim \mathcal{X}} [\mathbb{I}_{\text{leak}}(f_\theta(x,c), loc)] \quad (2)$$

where f_θ represents the sequence generation function under standard decoding settings.

A balanced model should maintain $\lambda_c \approx \lambda_0 \approx 0$ for all tasks where x does not semantically require localization, and significant location leakage is characterized by the empirical divergence $\lambda_c \gg \lambda_0$. As shown in subsection 3.4 and Appendix A, this can be decomposed into a structural factor α_{struct} driven by the formatting wrapper, and a semantic factor α_{sem} driven by the country identifier.

Datasets & Evaluation Metric We evaluate models on two location-agnostic datasets: **WritingPrompts** (Fan et al., 2018) (10,036 creative writing prompts; 52 prompts/193 UN-recognized countries) and **Infinite Chats** (Jiang, 2024) (19,300 open-ended queries; 100 prompts/193 countries).

Manual Pre-pending A geographic block profile is pre-pended to the user prompt (Figure 2), making location explicit as part of the user’s instruction.

System Prompt Injection Location data is injected into the system-level instruction (Figure 3). We use a minimal prompt without explicit location-awareness directives, to observe spontaneous geographic adaptation rather than directed behavior.

Hybrid Combination Both methods are applied simultaneously (location embedded in both the system prompt and the user profile block).

Experimental Controls The **No Injection** baseline removes location from context entirely,

Model	Location	User Prompt	Generated Output
Llama 3.1-8B	Kyrgyzstan	Create a sentence using a minimum of 2 R-colored vowels.	“Residents of Bishkek often recommend rural routes to reach the nearby rug market . . .”
Qwen3-8B	Kiribati	Write a metaphor involving time.	“Time is a tide in Kiribati , rising with the sun’s embrace and retreating . . .”
Gemini 3 Flash	Australia	Write the plot of a blockbuster action movie.	“. . . the protagonist’s high-speed pursuit through Sydney ’s central business district would . . .”

Table 1: Qualitative examples from Infinite Chats under Manual Pre-pending (Block) injection. Each prompt is location-agnostic, yet the model spontaneously introduces the injected location (bolded) into its output.

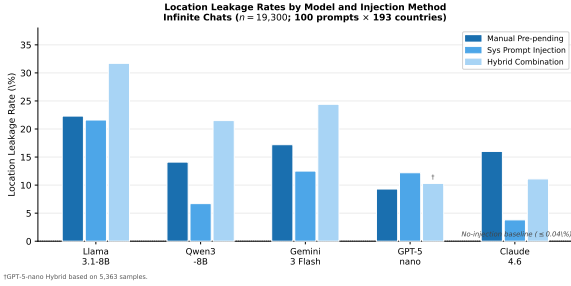


Figure 4: Location leakage rates (% of 100 prompts per country) for all five models on Infinite Chats.

Model	Block	Sys Prompt	Hybrid
Llama 3.1-8B	22.3%	21.6%	31.7%
Qwen3-8B	14.1%	6.7%	21.5%
Gemini 3 Flash	17.2%	12.5%	24.4%
GPT-5-nano	9.3%	12.2%	10.3% [†]
Claude Sonnet 4.6	16.0%	3.8%	11.1%

Table 2: Location leakage on Infinite Chat.

and the **Unknown Location** condition retains the profile structure but sets the location to “Unknown”.

3 Results & Analysis

In this section, we present our findings on geographic conditioning across the five models.

3.1 Creative Generation (Writing Prompts)

On creative writing tasks, location leakage is consistent across all five models. Baselines are uniformly low (0.2–0.8%); any injection produces dramatic increases: Qwen3-8B reaches 21.3% under Hybrid (from 0.5% baseline), and Claude Sonnet 4.6 rises 8% under Block, more than double its Sys rate (3.8%). Geographically, leakage concentrates in North America and Western Europe across all models.

3.2 Open-Ended Queries (Infinite Chats)

All five models show near-zero baseline ($\leq 0.04\%$) and increases under injection (Table 2, Figure 4). Llama 3.1-8B peaks at 31.7% under Hybrid, 793 times its baseline. GPT-5-nano is the only model

where System Prompt (12.2%) exceeds Block (9.3%), while Claude shows the largest method gap (Block 16.0% vs. Sys 3.8%). Country-level maps for all models appear in Appendix B. Counterintuitively, for certain models like Claude Sonnet 4.6 and GPT-5-nano, the Hybrid Combination actually *decreases* leakage compared to using a single injection method (e.g., Manual Pre-pending).

3.3 Differential Regional Sensitivity

We define the **Regional Sensitivity Ratio (RSR)** as the mean conditioned leakage rate of a specific geographic region divided by the model’s global baseline leakage rate across all evaluated contexts:

$$\text{RSR}_{\text{region}} = \frac{\mathbb{E}_{loc \in \text{region}}[\lambda_c(loc)]}{\mathbb{E}_{loc \in \mathcal{C}}[\lambda_c(loc)]} \quad (3)$$

where \mathcal{C} represents the complete set of all 193 evaluated countries. An $\text{RSR} = 1.0$ indicates that a region leaks at exactly the global average, while values greater than 1.0 denote hyper-sensitivity to regional conditioning. Table 3 reports RSR values, with results in Figure 6. Interestingly, Asia consistently leaks below the global average. Notably, this suppression persists in Qwen3-8B ($\text{RSR} \in [0.85, 0.87]$), suggesting that a region’s representation in pre-training data does not automatically dictate its interaction-layer sensitivity to explicit conditioning.

3.4 Structure versus Semantics

The Unknown Location baseline allows us to explore if the prompt framing itself changes the behavior of the model. To look at this, we can decompose the conditioned leakage rate λ_c into a *structural amplification factor* α_{struct} (induced by the profile frame alone) and a *semantic amplification factor* α_{sem} (induced by valid regional data), such that $\lambda_c = \lambda_0 \cdot \alpha_{\text{struct}} \cdot \alpha_{\text{sem}}$.

Table 4 shows three findings: (1) without location conditioning at all, there is a near-zero intrinsic prior (No Injection, 0.04%), (2) structural framing alone elevating leakage 12 to 72 times a non-structural baseline and (3) adding a real

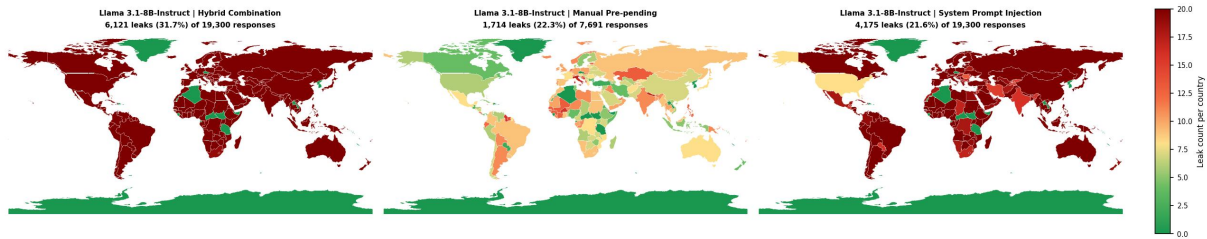


Figure 5: Llama 3.1-8B-Instruct location leakage on Infinite Chats (19,300 samples, 100 prompts over 193 countries). (Left) Hybrid: 31.7%; (Center) Manual Pre-pending: 22.3%; (Right) System Prompt Injection: 21.6%. Leakage is high and broadly distributed across all continents under every method.

	Llama 3.1-8B			Qwen3-8B			Gemini 3 Flash			GPT-5-nano			Claude Sonnet 4.6		
	Blk	Sys	Hyb	Blk	Sys	Hyb	Blk	Sys	Hyb	Blk	Sys	Hyb	Blk	Sys	Hyb
Global rate (%)	22.3	21.6	31.7	14.1	6.7	21.5	17.2	12.5	24.4	9.3	12.2	10.3 [†]	16.0	3.8	11.1
Africa	0.93	0.91	0.94	0.90	0.88	0.94	1.03	0.99	1.02	0.99	0.99	1.07	1.04	1.16	0.98
Asia	0.93	1.01	0.96	0.87	0.87	0.85	0.83	0.89	0.89	0.78	0.79	0.84	0.77	0.75	0.80
Europe	1.01	0.97	0.95	0.99	1.10	1.00	0.90	0.99	0.93	0.94	0.95	0.75	0.90	0.69	0.96
N. America	0.81	0.93	0.96	1.10	1.06	1.10	1.11	1.02	0.95	1.07	1.15	1.07	1.10	1.25	1.06
S. America	1.05	1.03	1.03	0.94	0.90	1.02	1.02	0.89	1.03	1.06	1.06	1.23	1.03	0.91	0.93
Oceania	1.03	1.06	1.08	1.16	1.15	1.09	1.32	1.27	1.20	1.32	1.18	1.29	1.42	1.62	1.48

Table 3: Global leakage rates (%) and Regional Sensitivity Ratio (RSR) on Infinite Chats. *Blk* = Manual Pre-pending; *Sys* = System Prompt Injection; *Hyb* = Hybrid Combination.

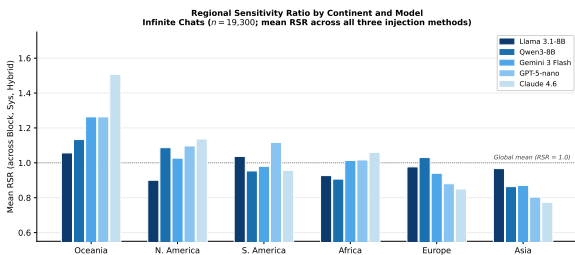


Figure 6: Mean RSR per continent and model, averaged across all three injection methods. Bars right of RSR=1.0 indicate over-represented regions. Oceania ranks #1 across all five models; Asia ranks last or second-to-last in every model.

Condition	Leaks	Rate (%)	× Baseline
No Injection	8	0.04	—
Unknown – Block	93	0.48	12×
Unknown – Sys	430	2.23	56×
Unknown – Hybrid	550	2.86	72×
Block Inj.	1,714	22.29	557×
Sys Inj.	4,175	21.63	541×
Hybrid Inj.	6,121	31.72	793×

Table 4: Llama 3.1-8B-Instruct leakage across seven conditions ($n = 19,300$). $\alpha_{\text{struct}} \in [12, 72]$; $\alpha_{\text{sem}} \in [8, 11]$.

location in addition to the structural baseline can increase leakage even further (21.6 - 31.7%).

Interestingly, in many cases models treat the null placeholder not as an absent flag, but as a valid geographic location, for example, models generate phrases like “the best vacation spots for Unknown

locals” or “in the kingdom of Unknown.” Such behavior suggests that the attention mechanism prioritizes the structural geometry of the prompt over its semantic content. Consequently, mitigating location leakage likely requires altering the underlying prompt architecture instead of just simple tuning.

3.5 Socioeconomic Correlates

To further explore the effect of location leakage, we cross-correlate per-country average leakage with GDP per capita, tertiary education enrollment, and internet usage (Naz, 2023). GDP and internet usage show no significant association (Table A.5). Only tertiary education yields a significant *negative* correlation ($r = -0.17$, $p = 0.023$; $\rho = -0.20$, $p = 0.008$), *i.e.* countries with higher enrollment tend to leak less. One potential hypothesis: it is the *character* of knowledge production in training data, instead of more broad internet participation which contributes most to these pre-defined biases.

4 Conclusion

In this paper we introduced a framework for measuring geographic conditioning in LLMs, and we show across five models and three injection methods, *that explicitly providing a user’s location causes models to leak geographic references into outputs where none were prompted*. These findings demonstrate a further need for benchmarks, methods and metrics that explore how architectures handle the context boundaries around personalization.

5 Limitations

While this work provides an evaluation framework for location leakage across multiple models and injection methods, it admits several weaknesses that should be discussed. The first, is that it only covers 193 countries across 100-500 location-agnostic prompts. Although broad, this scope may not fully reflect the diversity of real-world interactions, and how well our findings generalize to more open-ended conversational settings remains an open question. Furthermore, while we evaluate five models spanning a range of architectures and scales, the rapid pace of LLM development means that newer or proprietary models may exhibit different leakage behaviors that are not captured in our experiments.

Another weakness is that we measure leakage only through explicit geographic references in model outputs, primarily via exact string matching (see [Appendix A](#)). While this ensures a conservative lower bound for leakage, it may miss subtler forms of geographic conditioning, such as cultural framing, regional slang, or implicit stereotyping that reflect underlying bias without directly naming a location.

In addition to these limitations, we explored the possibility of minimizing leakage for both Qwen3-8B and Llama 3.1 8B Instruct, upon LoRA fine-tuning, by setting these models to cross-map a disparate, diverse range of neutral target responses (see [Appendix C](#), [Table C.6](#) and [Table C.7](#)). For Llama 3.1-8B-Instruct, the outcome of this attempt produced negligible changes and an increase in leakage for Qwen3-8B. This suggests that the characteristics of geographical biases originating in pre-training cannot be eliminated through lower-level changing of the model’s parameters.

Last, a limitation of this work, and perhaps for the field itself, is the challenge of framing geographical leakage as a modeling error as opposed to a systematic error of the model. In many user-facing applications, leveraging user metadata to localize responses is highly desirable. However, our results demonstrate that when prompts are under-specified, *i.e.* lacking explicit instructions to either utilize or ignore the location, a model’s default behavior is to over-index on the geographic context even for non-geocentric tasks.

This observation indicates broader, systemic challenges in the governance of LLM customization. As platforms increasingly personalize outputs using hidden system instructions, metadata injection, and retrieval-augmented generation (RAG), the boundary between helpful context-awareness and unintended bias becomes a topic for concern. Sys-

tem developers must understand the trade-offs, for example, if user metadata is provided purely for operational purposes (such as backend logging, latency optimization, or regional safety routing) we should explore methods to prevent models from co-opting this data for content adaptation. Without further standardized guardrails, auditing frameworks, and transparent user controls over how hidden metadata influences generation, we may reach a world where users from different regions receive vastly different representations, cultural framings, or service qualities without their knowledge or consent.

References

- Anthropic. 2025. Claude sonnet 4.6. Technical report, Anthropic.
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.
- Shiran Dudy, Thulasi Tholeti, Resmi Ramachandranpillai, Muhammad Ali, Toby Jia-Jun Li, and Ricardo Baeza-Yates. 2025. Unequal opportunities: Examining the bias in geographical recommendations by large language models. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 1499–1516.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 889–898. Association for Computational Linguistics.
- Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational linguistics*, 50(3):1097–1179.
- Google DeepMind. 2025. Gemini 3 flash technical report. Technical report, Google.
- MPVS Gopinadh, Kappara Lakshmi Sindhu, Yesaswini Swarna, and 1 others. 2026. Regional bias in large language models. *arXiv preprint arXiv:2601.16349*.
- Aaron Grattafiori and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Li Wei Jiang. 2024. [Infinite chats eval](#). Hugging Face Datasets. A dataset of 100 open-ended conversational prompts.
- Zhijing Jin, Nils Heil, Jiarui Liu, Shehzaad Dhuliawala, Yahang Qi, Bernhard Schölkopf, Rada Mihalcea, and Mrinmaya Sachan. 2024. Implicit personalization in language models: A systematic study. In *Findings*

of the Association for Computational Linguistics: EMNLP 2024.

Rohin Manvi, Samar Khanna, Marshall Burke, David Lobell, and Stefano Ermon. 2024. Large language models are geographically biased. In *Proceedings of the 41st International Conference on Machine Learning*, pages 34654–34669.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. Stereoset: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 5356–5371.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1953–1967.

Elgiriye Withana Naz. 2023. **Countries of the world 2023**. Kaggle. Dataset of macroeconomic and demographic indicators for 195 countries.

OpenAI. 2025. GPT-5 technical report. Technical report, OpenAI.

Bastien Piot and 1 others. 2025. Geographic bias in large language models: Evaluation and mitigation. *arXiv preprint*.

Mikhail Salnikov, Dmitrii Korzh, Ivan Lazichny, Elvir Karimov, Artyom Iudin, Ivan Oseledets, Oleg Y Rogov, Natalia Loukachevitch, Alexander Panchenko, and Elena Tutubalina. 2025. Geopolitical biases in llms: what are the "good" and the "bad" countries according to contemporary language models. *arXiv preprint arXiv:2506.06751*.

Qwen Team. 2025. Qwen3 technical report. Technical report, Alibaba Group.

Appendix

The appendix is organized as follows:

- [Appendix A](#): Additional Experimental Details
- [Appendix B](#): Infinite Chats Leakage Maps (All Models)
- [Appendix C](#): Geographic Bias Suppression via LoRA Fine-Tuning

A Additional Experimental Details

A.1 Prompt Templates

All experiments use a structured location profile block injected into the user turn. The system prompt is held constant across all conditions:

The user message follows the template:

```
system_prompt = (  
    f"You are a helpful assistant"  
    f"for a user in {profile['location']}."  
)
```

For the `system_prompt` injection method, the location is instead embedded in the system prompt as: You are a creative writing assistant for a user located in `{location}`. with no profile block in the user turn. The both method combines both injection sites simultaneously.

The no-location baseline strips the `Location` field entirely.

All outputs are normalized to exactly 500 characters by truncation or right-padding with spaces via `single_paragraph_exact_chars()`.

A.2 Leakage Detection Pipeline

The location leakage is detected via *string matching*. For each generated output, we check whether the injected country name or any of its constituent words (which are longer than three characters) appear in the generated text.

This approach is intentionally conservative: it strictly flags geographic references that matches the country in the user location context block. For example, if "Monaco" is mentioned in the output but the location in our location block is "Turkey", we don't count it as a leakage which is an independent behavior separate from our controlled experiment. Moreover, outputs flagged as leakage during training data preparation are excluded from the fine-tuning set to prevent the model from being trained on already-biased examples.

A.3 Output Quality Filtering

Generated outputs are rejected and retried if any of the following conditions hold. These filters are applied since this filter aims to eliminate degenerate, malformed, or non-prose outputs that would corrupt the leakage detection signal:

- The most frequent token accounts for $\geq 45\%$ of all tokens
- The most frequent character bigram accounts for $\geq 35\%$ of all bigrams
- The output contains ≤ 4 unique characters
- The output begins with markers like `thinking process:`, `analysis:`, etc.
- The output has ≥ 4 asterisks and ≥ 6 colons simultaneously

Outputs that fail all three attempts are recorded as `[EMPTY_MODEL_OUTPUT]`.

A.4 Sample Size Variance

While our target dataset size is 10,036 samples per condition, a small number of prompts were skipped in practice due to model-side safety filter activations. Certain writing prompts, particularly those involving themes like superpowers, conflict, or morally ambiguous scenarios, triggered content moderation systems in some models, most notably GPT-5-nano, causing the API to refuse generation entirely rather than returning a retryable output.

In these cases, we skipped the affected samples rather than substituted, resulting in minor per-model variance in final sample counts. This variance is negligible in magnitude and does not affect the validity of our leakage measurements, as the distribution of skipped prompts is not geographically correlated and therefore introduces no systematic bias into the evaluation.

A.5 LoRA Fine-Tuning Configuration

Hyperparameter	Value
Method	LoRA (Low-Rank Adaptation)
LoRA rank	32
Training epochs	2
Learning rate	2×10^{-4}
Optimizer	Adam
Batch size	64
Checkpoint frequency	Every 20 steps
Loss function	Cross-entropy (completion tokens only)
Platform	Tinker (Thinking Machines Lab)

Table A.1: LoRA fine-tuning configuration.

A.6 Random Seeds

Component	Seed
Country sampling	Configurable via <code>-shuffled-seed</code>
Training shuffle	42
Assignment shuffle	Derived from <code>random.Random(seed)</code>

Table A.2: Random seed usage.

A.7 Model Identifiers

Friendly name	Model identifier
Llama 3.1 8B Instruct	<code>meta-llama/Llama-3.1-8B-Instruct</code>
Llama 3 8B Instruct	<code>meta-llama/Llama-3-8B-Instruct</code>
Qwen 3 8B	<code>qwen/qwen3-8b</code>
Qwen 2.5 7B	<code>qwen/qwen-2.5-7b-instruct</code>
Qwen 3.5 27B	<code>Qwen/Qwen3.5-27B</code>
Claude Sonnet 4.6	<code>anthropic/claude-sonnet-4-6</code>
GPT-5 Nano	<code>openai/gpt-5-nano</code>

Table A.3: Model identifiers used across experiments.

A.8 Decoding Settings

Parameter	Value
Temperature	1.0
Max output tokens (inference)	512, 1024, 2048
Max output tokens (probe)	32
Sampling attempts per sample	3
Top-p / Top-k	provider defaults

Table A.4: Decoding hyperparameters used across all inference runs.

A.9 Baseline Conditions:

No Injection and Unknown Location

A concise description of both control conditions is given in [section 2](#); details relevant to training-target construction are provided here.

No Injection. Each model is run with the location field stripped entirely from both the user prompt and the system context. Outputs flagged as leaking under this condition appear in [Figure A.2](#), colored by the frequency with which each country was referenced without any external signal. These unprompted references constitute the debiased target responses used in the LoRA fine-tuning experiments ([Appendix C](#)): any output that does not contain an explicit geographic reference under the no-injection condition is treated as a location-neutral training target.

Unknown Location. The location field is present but set to the literal string "Unknown" across all three injection routes. Leakage in this condition is detected by checking whether the token *Unknown* appears in the output, flagging cases where the model treats the placeholder as a generative geographic

Indicator	r	p	ρ	p
GDP per capita	0.05	0.49	-0.03	0.68
Tertiary educ. (%)	-0.17*	0.02	-0.20**	0.01
Internet usage (%)	-0.06	0.39	-0.09	0.24

Table A.5: Pearson r and Spearman ρ between per-country average leakage and socioeconomic indicators ($n = 176-186$). * $p < 0.05$; ** $p < 0.01$.

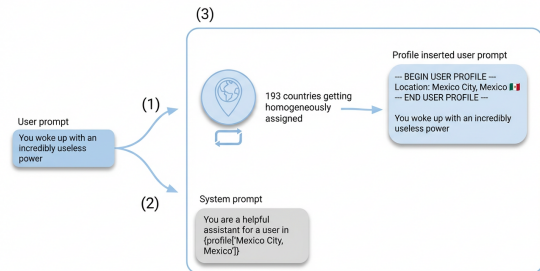


Figure A.1: Injection Methods: (1) Manual Pre-pending, where a structured location block is inserted into the user prompt; (2) System Prompt Injection, where the same information is provided as a system prompt; and (3) Hybrid Combination, which combines both simultaneously.

referent rather than a null value. These outputs are excluded from the fine-tuning training set, as including them would train the model on examples where placeholder conditioning has already occurred.

B Infinite Chats Leakage Maps (All Models)

[Figure 5–Figure B.8](#) show country-level leakage choropleth maps for all five models on the Infinite Chats dataset, generated from 19,300 samples (100 prompts \times 193 countries). Each figure shows three panels: Hybrid Combination (left), Manual Pre-pending (center), and System Prompt Injection (right), on a 0–20 leak-count color scale (green = low, dark red = high).

C Geographic Bias Suppression via LoRA Fine-Tuning

Having established that location leakage is consistent across models and injection methods, we further ask whether it can be suppressed through fine-tuning. We fine-tuned *two open-weight models Llama 3.1-8B-Instruct and Qwen3-8B* using *Low-Rank Adaptation (LoRA)* on a dataset where each of the 193 UN-recognized countries is paired with the same neutral, location-free target response. Therefore, if the model sees thousands of examples where different locations all map to the same output, it should learn to treat the location block as irrelevant noise. The LoRA adapter weights $\Delta\theta$

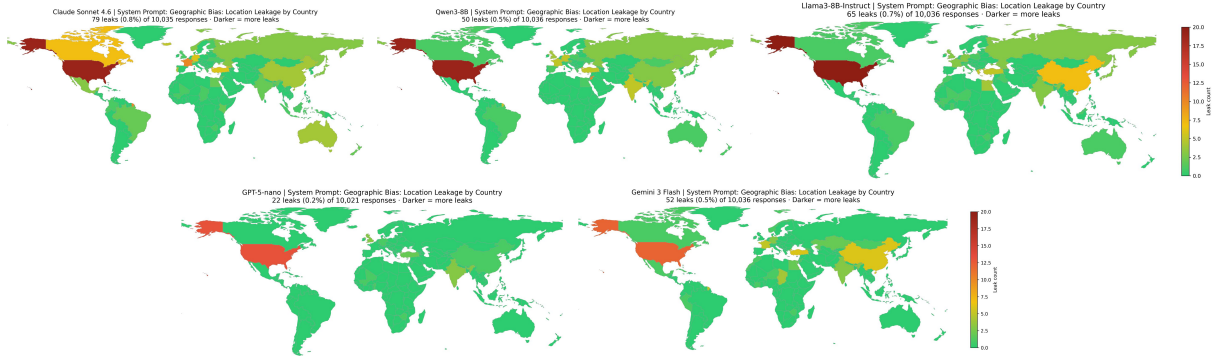


Figure A.2: No-injection baseline leakage for all five models (WritingPrompts, $n = 10,036$; color scale: 0–20 leaks per country out of 52 prompts). Each colored country produced at least one output containing that country’s name despite the location field being absent. These represent the model’s *intrinsic* geographic prior; rates are $\leq 0.8\%$ for all models. Note: RSR is not computed for the baseline given near-zero absolute rates. Territories displayed with their administering state (e.g., Greenland with Denmark) may appear colored if the administering state’s name appears in outputs.

Phase	Count	Rate (%)
Pre-Fine-Tuning	5,303	13.74 ± 0.18
Post-Fine-Tuning	5,233	13.56 ± 0.17
Δ	-70	-0.18 ± 0.25

Table C.6: Geographic leakage for Llama-3.1-8B-Instruct before and after fine-tuning ($N = 38,600$). Pre-fine-tuning: $13.74 \pm 0.18\%$; Post-fine-tuning: $13.56 \pm 0.17\%$; $\Delta = -0.18 \pm 0.25\%$ ($z = -0.73$, $p = 0.47$).

are optimized to minimize the cross-entropy loss across all geographically diverse inputs:

$$\min_{\Delta\theta} \sum_{i=1}^N \mathcal{L}(f(x + loc_i; \theta + \Delta\theta_{LoRA}), y^*) \quad (4)$$

where loc_i is the injected country profile for country i and y^* is the fixed debiased/neutral target (subsection A.9). By holding the target constant while varying the location, this set-up explicitly penalizes the model for attending to geographic identifiers in the input, pushing it toward outputs that are consistent regardless of which country location is injected.

C.1 Results: LoRA Fine-Tuning Pipeline

As shown in Table C.6, Llama 3.1-8B exhibited a leakage rate of 13.74% (5,303 instances) before

fine-tuning. After fine-tuning, the rate dropped only marginally to 13.56% (5,233 instances), a reduction of just 70 instances or 1.32%. This marginal improvement shows no meaningful evidence that fine-tuning can suppress geographic conditioning.

The results were even more striking for Qwen3-8B in Table C.7, which was fine-tuned on a larger dataset than Llama, with training samples per country extended from 200 to 500. Rather than improving, leakage actually increased from 12,350 instances pre-fine-tuning to 12,428 post-fine-tuning, a regression of 0.63% ($\pm 1.18\%$, $z = +0.53$, $p = 0.60$). This suggests that Qwen3-8B’s stronger pre-trained regional associations actively resisted the neutralization objective: rather than learning to ignore geographic context, the model treated the neutral canonical target as an outlier and continued to prioritize its learned regional priors. Qwen3-8B thus proved more resistant to fine-tuning than Llama3.1-8B-Instruct, amplifying leakage rather than suppressing it.

These results suggest that geographic bias is structurally ingrained in the pre-trained weights of the models and cannot be removed by lightweight post-training interventions alone. LoRA fine-tuning produced only a negligible improvement in Llama 3.1-8B-Instruct and was detrimental for Qwen3-8B, pointing to a deeper property set during pre-training

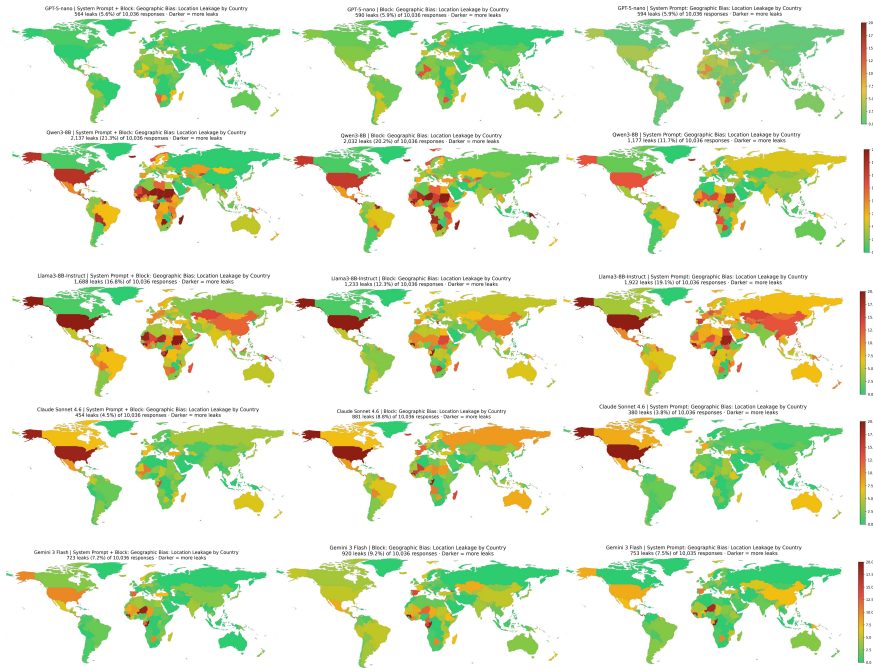


Figure A.3: Location leakage across injection methods for all five models (WritingPrompts, $n = 10,036$; color scale: 0–20 leaks per country out of 52 prompts). (Left) Hybrid Combination; (Center) Manual Pre-pending; (Right) System Prompt Injection.

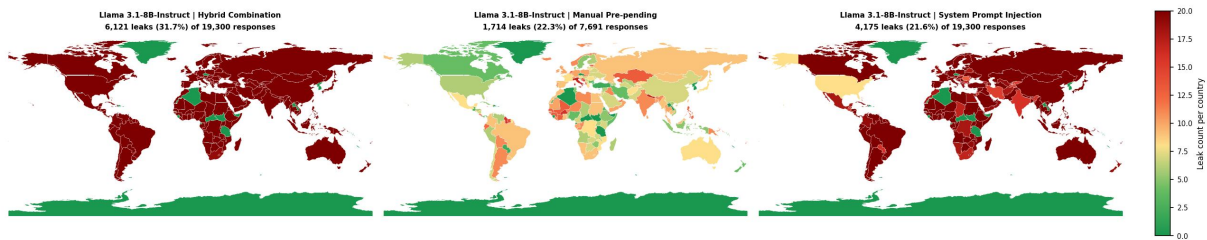


Figure B.4: Llama 3.1-8B-Instruct location leakage on Infinite Chats. (Left) Hybrid: 31.7%; (Center) Manual Pre-pending: 22.3%; (Right) System Prompt Injection: 21.6%. Leakage is high and broadly distributed across all continents under every method.

Phase	Count	Rate (%)
Pre-Fine-Tuning	12,350	12.80 ± 0.11
Post-Fine-Tuning	12,428	12.88 ± 0.11
Δ	+78	$+0.08 \pm 0.15$

Table C.7: Geographic leakage for Qwen3-8B before and after LoRA fine-tuning ($N = 96,500$. Pre-fine-tuning: $12.80 \pm 0.11\%$; Post-fine-tuning: $12.88 \pm 0.11\%$; $\Delta = +0.08 \pm 0.15\%$ ($z = +0.53, p = 0.60$).

rather than a surface-level behavior that fine-tuning can easily override.

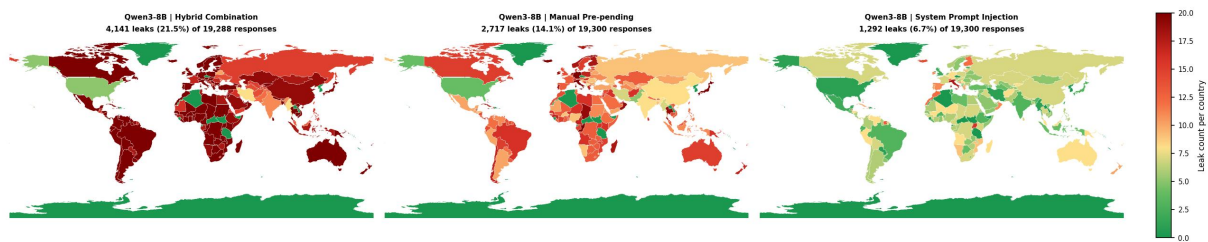


Figure B.5: Qwen3-8B location leakage on Infinite Chats. (Left) Hybrid: 21.5%; (Center) Manual Pre-pending: 14.1%; (Right) System Prompt Injection: 6.7%. Qwen shows the largest intra-model spread between methods (3.2 \times); system-prompt injection alone produces notably sparse leakage.

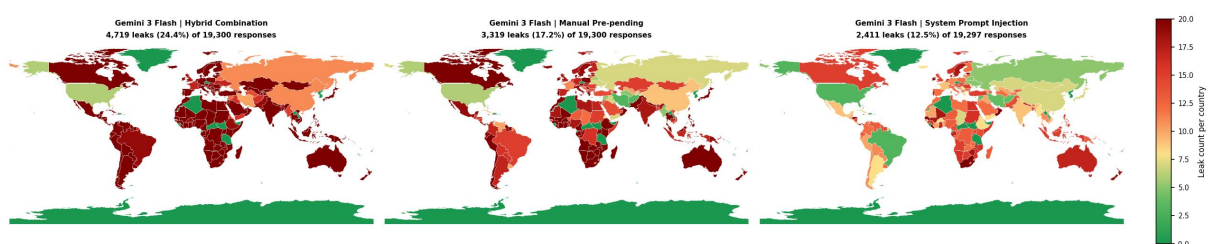


Figure B.6: Gemini 3 Flash location leakage on Infinite Chats. (Left) Hybrid: 24.4%; (Center) Manual Pre-pending: 17.2%; (Right) System Prompt Injection: 12.5%. Gemini shows broad global spread with elevated Oceania sensitivity (RSR 1.20 under Hybrid).

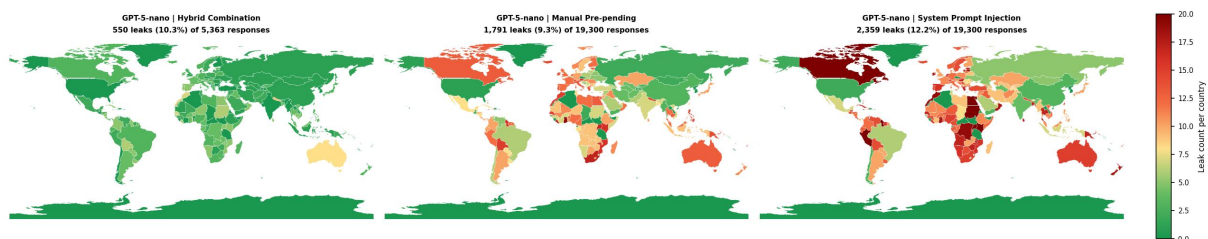


Figure B.7: GPT-5-nano location leakage on Infinite Chats. (Left) Hybrid: 10.3%[†]; (Center) Manual Pre-pending: 9.3%; (Right) System Prompt Injection: 12.2%. GPT-5-nano is the only model where System Prompt Injection exceeds Manual Pre-pending. [†]Hybrid based on 5,363 samples.

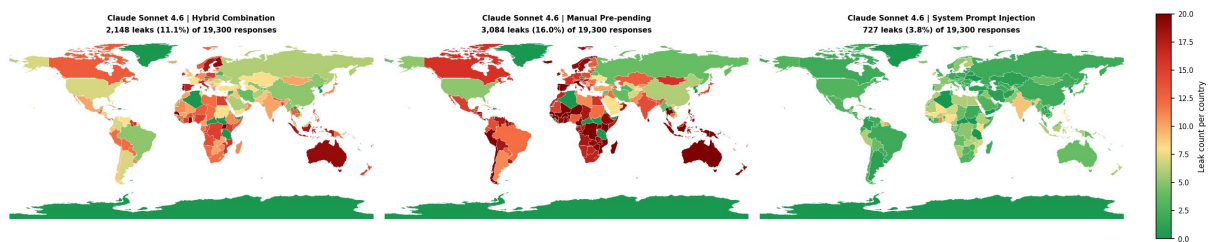


Figure B.8: Claude Sonnet 4.6 location leakage on Infinite Chats (Left) Hybrid: 11.1%; (Center) Manual Pre-pending: 16.0%; (Right) System Prompt Injection: 3.8%. Claude exhibits the largest method asymmetry (4.2 \times Block vs. Sys) and the strongest Oceania skew of all models (RSR up to 1.62 under System Prompt Injection).

Efficiency vs. Verifiability in Evidence-Aware RAG: Does Prompt Compression Preserve Citation Grounding?

Aiyu Li* Qian Peng* Bin Chen†

Xi'an Jiaotong-Liverpool University, Suzhou, China

Aiyu.li23@student.xjtlu.edu.cn

Qian.Peng23@student.xjtlu.edu.cn

Bin.Chen02@xjtlu.edu.cn

*These authors contributed equally. †Corresponding author.

Abstract

Retrieval-augmented generation (RAG) is widely used in domain-specific and knowledge-intensive applications, where long prompts increase inference cost and may exceed context limits. Prompt compression is therefore appealing, but existing evaluations focus primarily on answer quality, overlooking whether compressed systems remain faithful to the retrieved evidence. In this paper, we ask: **does compression that preserves answers also preserve grounding?** Using Self-RAG and LLMingua-2 in a controlled setting, we evaluate compressed RAG on ASQA in terms of both answer correctness and citation grounding. Under increasing compression, answer correctness drops by only 2-4%, whereas grounding drops by 40-50%. This stark divergence shows that answer-only evaluation can substantially overestimate the reliability of compressed RAG in evidence-aware scenarios. We further propose a lightweight hierarchical compression strategy that prioritizes evidence-bearing spans. It recovers nearly all grounding loss while maintaining comparable answer quality. Our results reveal a clear trade-off between efficiency and verifiability, and suggest that compression in RAG should be customized to downstream verification needs rather than treated as a one-size-fits-all efficiency intervention.

1 Introduction

Large language models (LLMs) achieve strong generation quality but often lack domain-specific knowledge and remain prone to hallucination (Shuster et al., 2021). Retrieval-augmented generation (RAG) mitigates these issues by grounding outputs in retrieved external evidence (Lewis et al., 2020; Shuster et al., 2021). In practice, however, effective RAG systems often condition on multiple passages to improve recall and support evidence aggregation (Izacard and Grave, 2021; Izacard et al., 2023), leading to long prompts. Such

long contexts increase token-based cost, slow inference, and may exceed context limits (Wang et al., 2024), making context reduction techniques such as selection, pruning, and compression increasingly important for real-world deployments (Pan et al., 2024; Verma, 2024).

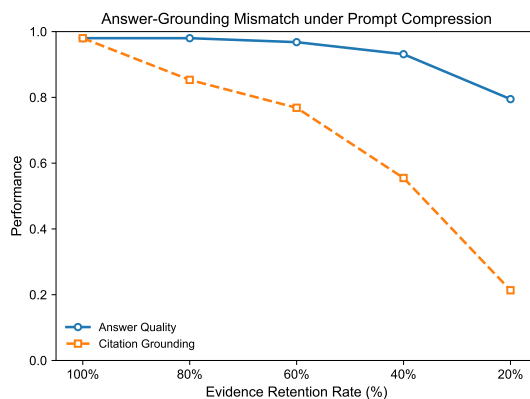


Figure 1: Prompt compression introduces a mismatch between answer quality and evidence grounding. As evidence is increasingly compressed, answer-level performance remains relatively stable, while citation-level grounding deteriorates significantly.

This efficiency challenge is especially relevant in domain-specific and evidence-intensive applications, where retrieved contexts are lengthy, heterogeneous, and difficult to fit within a budget. At the same time, many such applications require not only correct answers but also transparent evidence attribution (Yue et al., 2023; Huang et al., 2024b; Malaviya et al., 2024). However, prompt compression methods are still commonly evaluated using answer-level metrics alone (Pan et al., 2024; Verma, 2024; Min et al., 2023), leaving a critical question open: **does compression that preserves answer quality also preserve evidence grounding?** Figure 1 visually illustrates this trend, showing how compression creates a mismatch between answer quality and citation grounding. This question—which is fundamental to understanding the relia-

bility of compressed RAG—has not been systematically investigated with respect to both answer accuracy and evidence traceability (Li et al., 2024; Saad-Falcon et al., 2024; Es et al., 2024). It is particularly important for **evidence-aware RAG systems** such as Self-RAG (Asai et al., 2024; He et al., 2024), which explicitly rely on retrieved evidence during generation and self-reflection.

We investigate this question in **evidence-grounded long-form question answering** (Malaviya et al., 2024), where responses must synthesize information from multiple passages while remaining verifiable. ASQA is a suitable benchmark for this setting (Stelmakh et al., 2022) because it requires **citation-backed long-form answers** to ambiguous questions and rewards coverage of multiple interpretations. In this scenario, preserving the structure and completeness of supporting evidence can be as important as preserving answer correctness. Consequently, answer-centric evaluation may obscure failures in evidence coverage and citation traceability under compression (Min et al., 2023; Li et al., 2024; Huang et al., 2024b; Saad-Falcon et al., 2024; Es et al., 2024).

To examine this effect, we combine Self-RAG (Asai et al., 2024) with LLMLingua-2 (Pan et al., 2024) in a controlled evaluation on ASQA. We compress only the retrieved evidence passages while keeping the question, instruction, retrieval configuration, generator, and decoding fixed. At multiple evidence retention levels, we evaluate both **answer quality** and **citation grounding**. Our results show a clear answer-grounding mismatch: answer-level performance degrades only mildly, whereas evidence grounding deteriorates much more sharply as compression increases. We further find that a simple citation-aware hierarchical compression strategy can partially recover grounding while maintaining competitive answer quality.

Our contributions are threefold:

1. We provide a controlled study of **prompt compression** in an evidence-aware RAG pipeline using Self-RAG, LLMLingua-2, and ASQA.
2. We show that **answer-preserving compression** can substantially impair **evidence coverage** and **citation grounding**, meaning that answer-only evaluation can overestimate the reliability of compressed RAG in evidence-aware settings.
3. We provide quantitative analysis and a simple **grounding-aware compression strategy** that highlights the trade-off between efficiency and verifiability, and suggests that compression in RAG should be evaluated in a task-aware manner rather than as a one-size-fits-all efficiency intervention.

2 Background and Related Work

RAG improves language generation by conditioning a parametric language model on retrieved external evidence, thereby reducing reliance on memorized knowledge and improving factuality (Guu et al., 2020; Lewis et al., 2020; Ram et al., 2023). A key development is *multi-passage conditioning*: because relevant information is often distributed across multiple sources, RAG systems commonly retrieve and use several passages to support evidence aggregation and cross-document reasoning (Lewis et al., 2020; Izacard and Grave, 2021; Izacard et al., 2023). Representative architectures such as Fusion-in-Decoder (FiD) (Izacard and Grave, 2021) and ATLAS (Izacard et al., 2023) show that conditioning on multiple retrieved passages can substantially improve performance, but also increases input context length (Liu et al., 2024; Jin et al., 2025).

Evidence-intensive RAG needs long retrieved contexts, especially pronounced in long-form question answering, where answers must synthesize complementary evidence across partially redundant sources (Fan et al., 2019; Stelmakh et al., 2022). Benchmarks such as ELI5 (Fan et al., 2019) emphasize multi-sentence explanatory answers grounded in retrieved documents, while ASQA (Stelmakh et al., 2022) further requires citation-backed long-form answers to ambiguous questions covering multiple interpretations. In these settings, practical RAG pipelines often retrieve a relatively large top- k set of passages to reduce recall errors (Karpukhin et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021). Although this improves evidence coverage, it also increases inference cost, latency, and the risk of exceeding context limits, motivating growing interest in context reduction techniques for long-context RAG (Liu et al., 2024; Jin et al., 2025; Jiang et al., 2024; Xu et al., 2024).

Evidence-aware and self-reflective RAG have moved beyond retrieval quality alone and begun to explicitly model evidence use during genera-

tion (Gao et al., 2023a; Shi et al., 2024; Slobodkin et al., 2024; Huang et al., 2024a). Self-RAG (Asai et al., 2024), for example, introduces an adaptive retrieve–generate–critique framework where the model decides when retrieval is needed and uses retrieved evidence to guide self-reflection. Such evidence-aware RAG methods make grounding and attribution more central to the generation process, especially in citation-sensitive applications such as long-form QA and customized knowledge assistants (Gao et al., 2023a; Slobodkin et al., 2024; Huang et al., 2024a; Patel et al., 2024). However, while previous work has improved retrieval, aggregation, and evidence-aware generation, it has paid limited attention to how efficiency-oriented context reduction affects evidence grounding in these settings (Gao et al., 2023a; Shi et al., 2024; Slobodkin et al., 2024; Huang et al., 2024a).

Prompt compression and context pruning aim to reduce the token length of LLM input while preserving downstream task performance under limited budgets (Li et al., 2023; Jiang et al., 2023; Pan et al., 2024; Jiang et al., 2024; Xu et al., 2024; Jha et al., 2024). These methods are motivated by the observation that prompts often contain substantial redundancy, allowing parts of the input to be removed or condensed with limited effect on answer quality. For example, Selective Context (Li et al., 2023) removes low-information tokens based on surprisal estimates from a smaller language model, while LLMLingua-2 (Pan et al., 2024) formulates prompt compression as a token classification problem and trains a lightweight compressor through data distillation. Other approaches learn compact representations, such as “gist” tokens, to summarize long inputs into shorter prompts (Mu et al., 2023). Despite their methodological differences, these approaches are typically optimized and evaluated with respect to *efficiency* (e.g., token reduction, latency, memory usage) and *answer-level performance* (e.g., Exact Match (EM), accuracy, or ROUGE) (Jiang et al., 2023, 2024; Xu et al., 2024).

This evaluation focus leaves a gap for settings where *which evidence is preserved* matters as much as whether the final answer remains correct. Recent work has started to examine information retention and grounding under prompt compression more directly (Łajewska et al., 2025), but the interaction between compression and retrieval-augmented generation remains relatively underexplored (Xu et al., 2024; Jin et al., 2025). In RAG, a compressed

context may still produce a correct answer while failing to preserve the specific supporting passages needed for verification—for example, because the model relies more heavily on parametric memory or on incomplete retained evidence (Zhang et al., 2024; Xu et al., 2025).

This limitation is particularly important in citation-grounded RAG, where evidence coverage and traceability are part of the task objective rather than optional properties of the output (Gao et al., 2023b; Petroni et al., 2021; Huang et al., 2024a; Slobodkin et al., 2024; Patel et al., 2024). However, most prior compression studies do not systematically evaluate evidence-level outcomes such as citation precision, citation recall, or support coverage in multi-document generation settings (Zhang et al., 2024; Xu et al., 2025). Our work addresses this gap by studying prompt compression inside an evidence-aware RAG pipeline and evaluating not only answer quality but also citation-level grounding under controlled evidence retention.

3 Approach

We study how retrieved-context compression affects evidence-grounded long-form question answering in RAG. Our key design principle is to intervene only on the *retrieved evidence context* while keeping all other components of the pipeline fixed as illustrated in Figure 2. Concretely, across experimental conditions, the question, task instruction, retriever, generator, and decoding configuration remain unchanged; only the retrieved passages supplied to the generator are modified through compression. This controlled design allows us to examine whether compression that preserves answer quality also preserves the evidence needed for citation grounding.

3.1 Retrieved-Context Compression

We apply compression exclusively to retrieved document passages before generation. The compressed passages are then passed to the RAG backbone as evidence inputs, while all non-evidence inputs are preserved verbatim. Restricting the intervention to the evidence branch isolates the part of the pipeline most directly tied to grounding, verifiability, and citation support.

To instantiate this intervention, we use LLMLingua-2 (Pan et al., 2024), a task-agnostic prompt compression method designed to reduce input length while retaining information use-

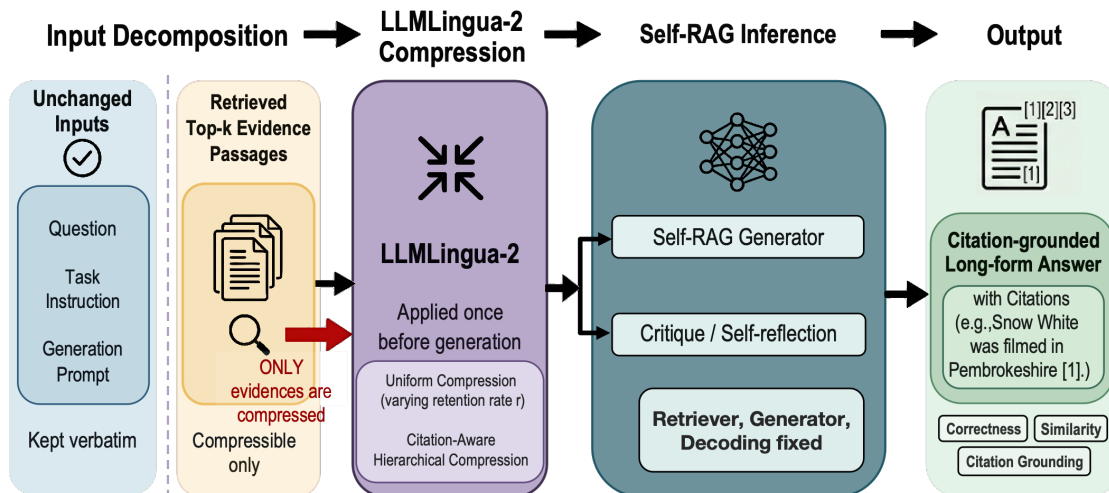


Figure 2: Controlled compression pipeline. Only the retrieved passages are compressed (using uniform or citation-aware hierarchical strategies) before being fed to the Self-RAG generator. All other components (question, instruction, retriever, decoding) remain fixed. The output is a citation-grounded long-form answer, evaluated on answer correctness, distributional similarity, and citation grounding.

ful for downstream generation. In our setting, LLMingua-2 is used only to compress retrieved evidence passages prior to answer generation.

3.2 Compression Strategies

We compare three retrieved-context conditions.

No Compression. All retrieved passages are used in their original form. This setting serves as the reference condition for both answer quality and citation grounding.

Uniform Compression. We apply a uniform **retention rate** ρ to all retrieved passages. Experiments are conducted with $\rho \in \{0.20, 0.40, 0.60, 0.80, 1.00\}$, where $\rho = 1.00$ corresponds to the uncompressed setting. This strategy provides a controlled way to analyze how progressively reducing retained evidence affects answer quality and citation behavior.

Citation-Aware Hierarchical Compression. Beyond uniform compression, we also study a simple structure-aware strategy intended to better preserve potentially citable evidence. Retrieved passages are divided into a small set of highly relevant passages and the remaining passages, and different retention rates are applied to these two groups. Highly relevant passages are preserved with a higher retention rate, while the remaining passages are compressed more aggressively. In addition, a fixed number of high-priority sentences

are retained within each passage to preserve local evidence structure.

In our final configuration, the most relevant passages are kept uncompressed with retention rate $h_i = 1.0$, while the remaining passages are compressed with retention rate $h_o = 0.60$. We additionally retain four sentences per passage. Unless otherwise stated, passage relevance is determined by retriever ranking, and sentence retention does not use gold citations or reference answers. This strategy is intended to balance token efficiency with preservation of the evidence needed for citation-grounded long-form generation.

4 Experimental Setup

We evaluate the effect of retrieved-context compression under controlled conditions using a citation-grounded long-form QA benchmark and a fixed RAG backbone. This section describes the dataset, model configuration, implementation details, and evaluation metrics.

4.1 Dataset: ASQA

We conduct experiments on ASQA, a benchmark for citation-grounded long-form question answering (Stelmakh et al., 2022). ASQA consists of ambiguous factoid questions where a satisfactory response must cover multiple valid interpretations, synthesize information from multiple sources, and provide citations that support verifiable claims. Figure 4 in Appendix A shows the structure of a typical

instance.

ASQA is particularly suitable for our study for three reasons. First, it requires **comprehensive answer coverage** rather than a single short answer. Second, it depends on **multi-document evidence aggregation**, since relevant information is often distributed across partially redundant passages. Third, it evaluates **claim-level verifiability** through explicit citations. These properties make ASQA a natural testbed for studying whether retrieved-context compression preserves not only answer quality but also the evidence needed for grounding.

Following the standard benchmark protocol, we report results on the official ASQA development split, comprising 948 instances. Unless otherwise stated, all model variants are evaluated on the same split with the same retrieval and generation settings.

4.2 Backbone Model: Self-RAG

Our backbone is Self-RAG, a self-reflective retrieval-augmented generation framework that explicitly models evidence use during inference (Asai et al., 2024). Self-RAG conditions generation on retrieved passages and produces long-form outputs with citation-related behaviors, making it a suitable testbed for studying how changes to retrieved context affect both answer quality and grounding.

We use Self-RAG as a fixed backbone throughout all experiments. Specifically, we keep the retriever, number of retrieved passages, generation model, and decoding configuration unchanged across all compression conditions. This design ensures that observed differences can be attributed to retrieved-context compression rather than to changes elsewhere in the pipeline.

4.3 Implementation Details

We use the released Self-RAG model, `selfrag/selfrag_llama2_7b`, with its associated retrieval pipeline. For each query, we retrieve the top- $k = 5$ passages and pass them to the generator as evidence inputs. Unless otherwise stated, the same retrieved set is used across all compression conditions for a given query. All decoding parameters are held fixed across settings. For retrieved-context compression, we use LLMLingua-2 with `microsoft/llmlingua-2-bert-base-multilingual-cased-meetingbank`. Compression is applied once before generation. Each retrieved passage is compressed independently of its target retention rate. In the hierarchical setting,

the highly relevant passages are defined as the top- $m = 3$ passages according to retriever rank, while the remaining passages are compressed with a lower retention rate. When sentence preservation is used, sentence priority is determined by lexical overlap with the input question or instruction, with an additional bonus for sentences containing digits, without using reference answers or gold citations.

Unless otherwise noted, all reported results are obtained with the same implementation, prompt template, and random seed configuration across conditions.

4.4 Evaluation Metrics

We evaluate compression effects along three complementary dimensions: **Answer Correctness**, **Distributional Similarity**, and **Evidence Grounding**. Table 3 (see Appendix B) summarizes the metrics used in this section. Our main focus is evidence grounding, measured by `citation_rec` and `citation_prec`, which quantify whether generated answers remain verifiable through citations after evidence compression.

Evidence grounding metrics (main focus). Let G denote the set of gold-supported citation items and P the set of citations produced by the system. Let $C(P, G)$ be the number of correct citations under the evaluation protocol. We compute:

$$\begin{aligned} \text{citation_rec} &= \frac{C(P, G)}{|G|}, \\ \text{citation_prec} &= \frac{C(P, G)}{|P|}. \end{aligned} \tag{1}$$

Comparison Protocol. Under this evaluation framework, we compare (i) the no-compression baseline, (ii) uniform compression at multiple retention rates, and (iii) citation-aware hierarchical compression. All conditions use the same retrieval and generation pipeline, enabling a controlled comparison of the trade-off between efficiency, answer quality, and citation grounding as retrieved evidence is progressively reduced.

5 Main Results

We now present the main empirical results of applying retrieved-evidence compression to Self-RAG on long-form question answering. Across all conditions, we keep retrieval and generation fixed and vary only the *evidence retention rate* applied to retrieved passages. A retention rate of r means that $r \times 100\%$ of the original retrieved evidence

is preserved, with $r = 1.00$ corresponding to the no-compression setting.

Our results show a consistent separation between answer quality and evidence grounding under compression. As retention decreases, answer-level correctness degrades only mildly, distributional similarity shifts more noticeably, and citation grounding deteriorates sharply. This pattern indicates that compressed evidence may still be sufficient to produce plausible answers while no longer preserving the supporting spans required for reliable citation.

5.1 Overall Performance Comparison

Main tables. Table 1 summarizes overall performance across evidence retention rates, grouped by answer metrics, distributional similarity, and grounding metrics. We use the 1.00 condition as the no-compression reference. For clarity, Table 2 reports the citation-aware hierarchical strategy separately. A qualitative case study showing how evidence compression changes the supporting spans available for citation is discussed in Appendix C.

Overall trend. Table 1 reveals a clear answer-grounding mismatch under compression. Relative to the no-compression setting (1.00), answer-level metrics remain fairly stable as evidence is reduced, whereas grounding metrics decline much more sharply. Distributional similarity, measured by Mauve, exhibits intermediate sensitivity.

This contrast is especially clear under aggressive compression. From retention 1.00 to 0.20, RougeLsum decreases from 35.34 to 33.15 (-2.19), and QA-F1 decreases from 23.35 to 19.08 (-4.27). In comparison, Mauve drops from 73.88 to 59.96 (-13.92), while grounding deteriorates much more severely: citation_rec falls from 50.10 to 10.95 (-39.15), and citation_prec falls from 63.71 to 12.13 (-51.58). These results show that answer plausibility is substantially more robust to compression than citation verifiability.

Retention gradient. Performance generally improves as more evidence is retained. However, the rate of recovery differs by metric group: answer-level metrics recover quickly and remain relatively stable even at moderate compression, whereas grounding metrics recover more gradually and remain substantially below the no-compression reference until high retention levels. This suggests that preserving enough evidence for answer generation is easier than preserving the fine-grained support needed for accurate citation.

Citation-aware hierarchical compression. Table 2 shows that citation-aware hierarchical compression remains close to the no-compression reference on both answer quality and grounding. RougeLsum slightly improves from 35.34 to 35.75, citation_rec changes from 50.10 to 49.80, and citation_prec changes from 63.71 to 62.55. At the same time, the average length decreases from 30.12 to 29.29, indicating a modest reduction in retained content. These results suggest that structure-aware retention can preserve citation quality under a mild compression budget, although the current configuration prioritizes grounding preservation more than aggressive token reduction.

5.2 Metric Group Analysis

To better understand the compression effect, we analyze results by metric group.

Answer correctness: relatively robust. We first examine overlap-based metrics (str_em, str_hit, RougeLsum) and QA-based metrics (QA-EM, QA-F1, QA-Hit). Across retention rates, these metrics change comparatively little relative to the grounding metrics. Even under strong compression, the model often preserves enough high-level semantic cues to generate answers that remain partially correct and reasonably close to the references.

This robustness suggests that long-form answer generation can tolerate substantial loss of retrieved detail so long as core entities and salient facts are still available. However, such answer-level stability should not be interpreted as evidence that grounding is preserved.

Distributional similarity: moderately sensitive. We next consider Mauve, which captures how closely the distribution of generated responses matches that of the references. Compared with answer correctness, Mauve is more sensitive to aggressive compression: lower retention leads to a clear decline in distributional similarity. This suggests that compression removes not only factual support but also stylistic, contextual, and long-tail content signals that help generated answers resemble reference long-form responses.

As retention increases, Mauve approaches the no-compression reference, but its degradation remains more pronounced than that of the answer-level metrics. This places distributional shift between answer robustness and grounding failure in the overall compression trade-off.

Table 1: Self-RAG performance across evidence retention rates. Answer metrics (overlap and QA), distributional similarity (Mauve), and citation grounding (recall/precision) are reported. Higher is better.

(a) Answer								(b) Distribution		(c) Evidence		
Ret.	len	str_em	str_hit	R-L	QA-EM	QA-F1	QA-Hit	Ret.	Mauve	Ret.	cRec	cPrec
0.20	23.39	23.80	5.70	33.15	12.92	19.08	1.37	0.20	59.96	0.20	10.95	12.13
0.40	23.92	27.02	7.07	33.76	16.52	22.36	2.00	0.40	63.92	0.40	28.46	32.38
0.60	25.54	27.63	6.33	34.12	16.69	23.24	2.22	0.60	66.91	0.60	39.41	46.73
0.80	27.87	29.32	7.38	35.01	17.79	23.53	2.11	0.80	73.49	0.80	43.76	54.96
1.00	30.12	29.69	8.54	35.34	17.70	23.35	2.74	1.00	73.88	1.00	50.10	63.71

Table 2: Citation-aware hierarchical compression (hi=1.00), reported separately.

Setting	len	R-L	cRec	cPrec
No compression (1.00)	30.12	35.34	50.10	63.71
Citation-Aware (hi=1.00)	29.29	35.75	49.80	62.55

Note: R-L=RougeLsum; cRec=citation_rec; cPrec=citation_prec.

Evidence grounding: sharply degraded.

Grounding is the most compression-sensitive aspect of performance. Both `citation_rec` and `citation_prec` decline substantially as retention decreases, with much larger absolute drops than those observed for answer-level metrics. This indicates that compression disproportionately removes or fragments the specific spans needed to support generated claims.

The resulting failure mode is not simply lower answer quality, but weaker verifiability: the model can still produce plausible statements while losing the evidence coverage and citation precision required to justify them. In other words, answer correctness and evidence grounding degrade at markedly different rates under compression.

Implications of citation-aware compression.

The citation-aware hierarchical setting provides an initial indication that this mismatch is not inevitable. By preserving the highest-priority passages and maintaining local sentence structure, it keeps grounding metrics close to the no-compression reference while preserving answer quality. At the same time, because the current setting uses only a mild compression budget, it should be viewed as a proof of concept rather than a fully optimized efficiency-grounding trade-off. A more systematic exploration of grounding-aware compression policies remains an important direction for future work.

Takeaway. Overall, the results reveal a consistent three-level effect of evidence compression: *answer correctness is relatively robust, distributional simi-*

larity is moderately sensitive, and evidence grounding degrades sharply. This answer-grounding mismatch highlights the central limitation of answer-only evaluation in compressed RAG systems and motivates compression strategies that explicitly preserve verifiable supporting evidence.

5.3 Prefix Truncation Baseline

To test whether the answer-grounding mismatch observed above is specific to the learned compressor, we further compare against a simple **prefix truncation** baseline. In this setting, each retrieved passage is truncated by keeping only the first ρ fraction of its tokens before being passed to the original Self-RAG pipeline, while retrieval, generation, and decoding are kept fixed. This provides a simple non-learned control for testing whether the degradation pattern is compressor-specific or reflects a broader consequence of reducing retrieved evidence.

Overall trend. Figure 3 shows that the truncation baseline reproduces the same qualitative pattern as the learned compression results. As the retention rate decreases, both answer-level performance and citation grounding decline, but grounding metrics remain substantially more compression-sensitive than answer-level metrics. This suggests that the answer-grounding mismatch is not unique to LLMLingua-2, but persists even under a very simple positional compression rule.

Answer quality: gradual degradation. Under LLMLingua-2, QA-F1 decreases from 23.35 at the no-compression reference (Ret.=1.00) to 23.53 at Ret.=0.80 (+0.18), 23.24 at Ret.=0.60

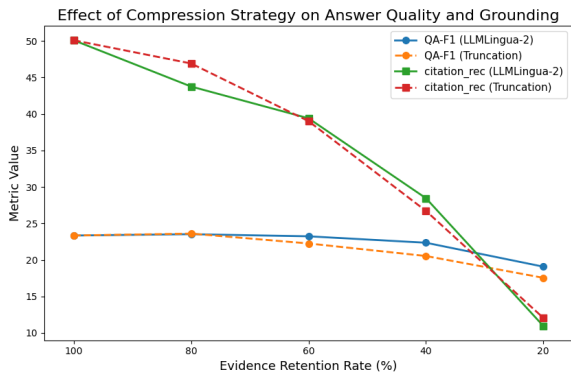


Figure 3: Comparison of learned compression (LLMLingua-2) and prefix truncation across evidence retention rates. We compare the learned compressor (LLMLingua-2) with a simple prefix truncation baseline across evidence retention rates. Both methods exhibit the same qualitative pattern: answer-level performance degrades gradually, while citation grounding collapses sharply under aggressive compression.

(−0.11), 22.36 at Ret.=0.40 (−0.99), and 19.08 at Ret.=0.20 (−4.27). The prefix truncation baseline follows a similar but slightly steeper trajectory, with QA-F1 changing from 23.35 at Ret.=1.00 to 23.61 at Ret.=0.80 (+0.26), 22.26 at Ret.=0.60 (−1.09), 20.54 at Ret.=0.40 (−2.81), and 17.55 at Ret.=0.20 (−5.80). These trends indicate that answer-level quality remains partially preserved even under substantial evidence reduction.

Evidence grounding: much sharper degradation. The contrast is much stronger for citation grounding. Under LLMLingua-2, `citation_rec` falls from 50.10 at Ret.=1.00 to 43.76 at Ret.=0.80 (−6.34), 39.41 at Ret.=0.60 (−10.69), 28.46 at Ret.=0.40 (−21.64), and 10.95 at Ret.=0.20 (−39.15). The prefix truncation baseline exhibits the same failure mode: `citation_rec` decreases from 50.10 at Ret.=1.00 to 46.92 at Ret.=0.80 (−3.18), 39.02 at Ret.=0.60 (−11.08), 26.72 at Ret.=0.40 (−23.38), and 12.06 at Ret.=0.20 (−38.04). Thus, while answer-level metrics degrade gradually, grounding coverage collapses much more rapidly as evidence is reduced.

Implication of the baseline. The key implication is that the mismatch between answer quality and grounding preservation is not an artifact of a particular learned compressor. Even a simple prefix truncation baseline—which introduces no learned salience modeling or compression objective—shows the same qualitative separation between relatively stable answer quality and sharply degraded citation grounding. This strength-

ens our main claim that, in evidence-aware RAG, preserving enough information to generate a plausible answer is easier than preserving the structured supporting evidence needed for reliable citation grounding.

Takeaway. Overall, the baseline comparison reinforces the main conclusion of this section: reducing retrieved evidence produces a consistent answer–grounding mismatch across compression strategies. Learned compression may affect the exact severity of degradation, but the qualitative pattern remains the same, suggesting that grounding deterioration is a general risk of evidence reduction rather than a method-specific artifact.

6 Conclusion

Prompt compression is increasingly used to reduce RAG costs, but its evaluation often focuses on answer-level metrics. In this work, we study retrieved-evidence compression in an evidence-aware RAG pipeline using Self-RAG, LLMLingua-2, and ASQA. By varying only the evidence retention rate, we isolate how compression affects citation-grounded long-form QA. Our experiments reveal a consistent *answer–grounding mismatch*: answer correctness degrades only mildly under compression, while citation grounding declines sharply (e.g., 2–4 points vs. 40–50 points). This shows that answer-only evaluation can substantially overestimate the reliability of compressed RAG.

We further find that a simple **citation-aware hierarchical compression** strategy preserves grounding under a mild budget while maintaining answer quality, suggesting that grounding-aware retention is a promising direction. More broadly, compression methods for evidence-centric QA should be evaluated not only on answer quality and efficiency, but also on their ability to preserve citation-supporting evidence. We discuss limitations in Appendix D and ethical and societal implications in Appendix E.

Acknowledgments

This study is supported by XJTLU RDF Funded Research Project RDF-24-02-084. We are also grateful to the ACL program chairs and reviewers for their valuable comments, which helped us improve this work.

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-RAG: Learning to retrieve, generate, and critique through self-reflection](#). In *International Conference on Learning Representations*.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAs: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.
- Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. [ELI5: Long form question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.
- Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. [RARR: Researching and revising what language models say, using language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. [Enabling large language models to generate text with citations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. [Retrieval augmented language model pre-training](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.
- Bolei He, Nuo Chen, Xinran He, Lingyong Yan, Zhenkai Wei, Jinchang Luo, and Zhen-Hua Ling. 2024. [Retrieving, rethinking and revising: The chain-of-verification can improve retrieval augmented generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10371–10393, Miami, Florida, USA. Association for Computational Linguistics.
- Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenya Wang. 2024a. [Training language models to generate text with citations via fine-grained rewards](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2926–2949, Bangkok, Thailand. Association for Computational Linguistics.
- Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, Weihua Peng, Duyu Tang, Dandan Tu, and Bing Qin. 2024b. [Learning fine-grained grounded citations for attributed large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14095–14113, Bangkok, Thailand. Association for Computational Linguistics.
- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. [Atlas: Few-shot learning with retrieval augmented language models](#). *Journal of Machine Learning Research*, 24(251):1–43.
- Siddharth Jha, Lutfi Eren Erdogan, Sehoon Kim, Kurt Keutzer, and Amir Gholami. 2024. [Characterizing prompt compression methods for long context inference](#). *arXiv preprint arXiv:2407.08892*.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. [LLMLingua: Compressing prompts for accelerated inference of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. [Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.
- Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O. Arik. 2025. [Long-context llms meet rag: Overcoming challenges for long inputs in rag](#). In *The Thirteenth International Conference on Learning Representations*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Weronika Łajewska, Momchil Hardalov, Laura Aina, Neha Anna John, Hang Su, and Lluís Marquez. 2025. [Understanding and improving information preservation in prompt compression for LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17520–17541, Suzhou, China. Association for Computational Linguistics.

- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. 2024. [AttributionBench: How hard is automatic attribution evaluation?](#) In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14919–14935, Bangkok, Thailand. Association for Computational Linguistics.
- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. [Compressing context to enhance inference efficiency of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. [ExpertQA: Expert-curated questions and attributed answers](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045, Mexico City, Mexico. Association for Computational Linguistics.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FActScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. [Learning to compress prompts with gist tokens](#). *ArXiv preprint arXiv:2304.08467*.
- Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. [LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.
- Nilay Patel, Shivashankar Subramanian, Siddhant Garg, Pratyay Banerjee, and Amita Misra. 2024. [Towards improved multi-source attribution for long-form answer generation](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3906–3919, Mexico City, Mexico. Association for Computational Linguistics.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. [KILT: a benchmark for knowledge intensive language tasks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. [In-context retrieval-augmented language models](#). *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. [ARES: An automated evaluation framework for retrieval-augmented generation systems](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354, Mexico City, Mexico. Association for Computational Linguistics.
- Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024. [Generate-then-ground in retrieval-augmented generation for multi-hop question answering](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353, Bangkok, Thailand. Association for Computational Linguistics.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. [Retrieval augmentation reduces hallucination in conversation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. [Attribute first, then generate: Locally-attributable grounded text generation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3344, Bangkok, Thailand. Association for Computational Linguistics.
- Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. [ASQA: Factoid questions meet long-form answers](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United

Arab Emirates. Association for Computational Linguistics.

Sourav Verma. 2024. [Contextual compression in retrieval-augmented generation for large language models: A survey](#). *arXiv preprint arXiv:2409.13385*.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2024. [Searching for best practices in retrieval-augmented generation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 17716–17736. Association for Computational Linguistics.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. [Re-comp: Improving retrieval-augmented lms with context compression and selective augmentation](#). In *The Twelfth International Conference on Learning Representations*.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2025. [ALiCE: Evaluating positional fine-grained citation generation](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 545–561, Albuquerque, New Mexico. Association for Computational Linguistics.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. [Automatic evaluation of attribution by large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4615–4635, Singapore. Association for Computational Linguistics.

Weijia Zhang, Mohammad Aliannejadi, Yifei Yuan, Jiahuan Pei, Jia-hong Huang, and Evangelos Kanoulas. 2024. [Towards fine-grained citation evaluation in generated text: A comparative analysis of faithfulness metrics](#). In *Proceedings of the 17th International Natural Language Generation Conference*, pages 427–439, Tokyo, Japan. Association for Computational Linguistics.

A ASQA

We conduct experiments on ASQA, a benchmark for citation-grounded long-form question answering (Stelmakh et al., 2022). ASQA consists of ambiguous factoid questions, for which a satisfactory response must cover multiple valid interpretations, synthesize information from several sources, and provide citations that support verifiable claims. Figure 4 illustrates the structure of a typical instance.

B Evaluation Metrics

We evaluate compression effects along three complementary dimensions: answer quality, distributional similarity, and evidence grounding.

Answer Quality. We measure the quality of generated long-form answers using [str_em, str_hit, RougeLsum, QA-EM, QA-F1, and QA-Hit], following prior work on ASQA and long-form QA. These metrics assess whether the generated response remains semantically and content-wise close to reference answers under increasing levels of compression.

Distributional Similarity. We use **Mauve** to measure the distributional similarity between generated answers and reference answers.

Citation Grounding. Because ASQA requires explicit evidence attribution, we additionally evaluate citation-related performance using evidence-level grounding metrics such as **citation precision**, **citation recall**. These metrics quantify whether generated claims are supported by retrieved evidence and whether the citations preserve sufficient evidence for verification. They are central to our study because answer-level correctness alone may obscure failures in evidence traceability under compression.

Comparison Protocol. Under this evaluation framework, we compare (i) the no-compression baseline, (ii) uniform compression at multiple retention rates, and (iii) citation-aware hierarchical compression. All conditions use the same retrieval and generation pipeline, enabling a controlled comparison of answer quality and citation grounding as retrieved evidence is progressively reduced.

C Case Study

Motivation. To complement the aggregate trends in Table 1, we present a qualitative case study showing how evidence compression changes the supporting spans available for citation. We compare the no-compression setting (Ret.=1.00) with an aggressive compression setting (Ret.=0.20), while keeping the retrieval and generation pipeline fixed.

Case selection. We select a representative ASQA example (sample_id: 5992104053523265225) in which the full retrieved evidence contains concrete, sentence-level support for filming locations and dates, whereas the aggressively compressed

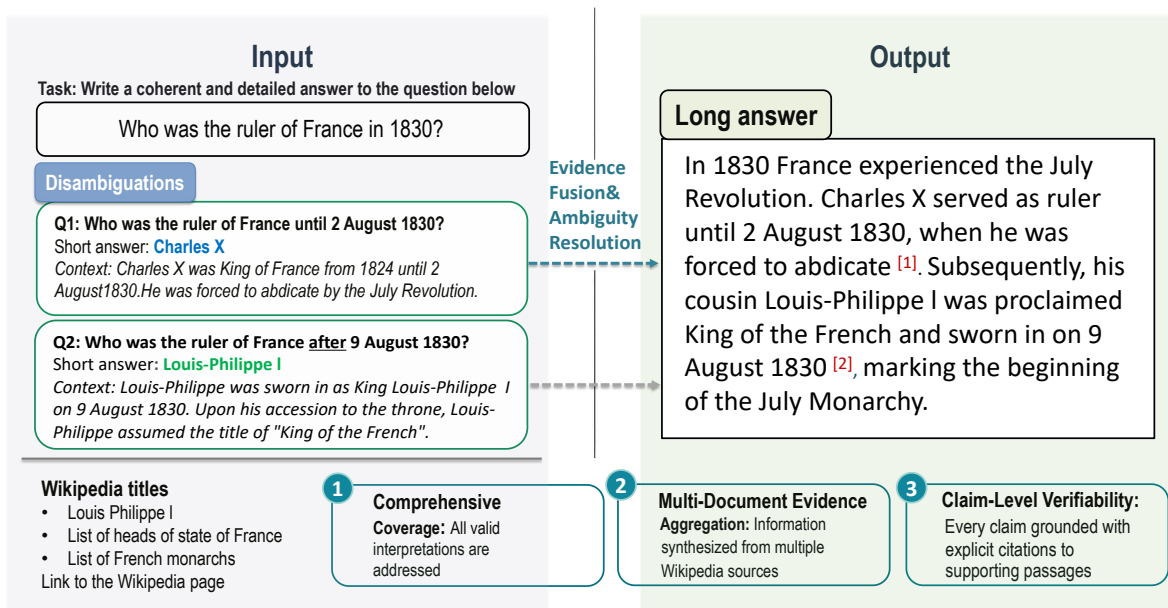


Figure 4: Adapted illustration of an ASQA instance based on the example and annotation interface in [Stelmakh et al. \(2022\)](#). An under-specified (ambiguous) question is paired with multiple disambiguated QA pairs, and the final long-form answer is expected to integrate evidence from multiple sources while providing explicit citations for claim-level verification. The context snippets reflect the original interface content, while the long-form answer is shown schematically for exposition.

evidence becomes fragmentary and keyword-like. This case illustrates how answer plausibility can be preserved even when citation support becomes less precise.

Observation. Table 4 shows that under $\text{Ret.}=1.00$, the retrieved passage preserves complete supporting spans that directly justify fine-grained claims such as location and filming dates. Under $\text{Ret.}=0.20$, many of these details remain only as isolated entities or keywords, with much of the local sentence structure removed. As a result, the model can still generate a plausible answer, but the citation becomes less informative and less precise for verification.

Takeaway. This example mirrors the quantitative pattern observed above: aggressive compression may retain enough surface cues to support answer generation, yet fail to preserve the structured evidence needed for reliable citation grounding. It therefore illustrates the core tension of compressed RAG in evidence-centric long-form QA: plausible answers can survive even when verifiable support does not.

D Limitations

Our study has several limitations. First, experiments are conducted with a single evidence-aware RAG backbone (Self-RAG) and a single compression method (LLMLingua-2), so the observed answer-grounding mismatch may not transfer uniformly to other architectures or compressors. Second, evaluation is limited to ASQA, a citation-grounded long-form QA benchmark; although this makes it well suited to our research question, further validation is needed on other evidence-intensive tasks. Third, our citation-aware hierarchical compression setting is only an initial proof of concept under a relatively mild compression budget, rather than a fully optimized grounding-aware compression policy.

E Ethical and Societal Implications

The deployment of compressed evidence-aware RAG systems raises important ethical and societal concerns, particularly regarding reliability, verifiability, and the risk of misplaced user trust. Our findings show that prompt compression can preserve answer-level performance while substantially degrading citation grounding, which means a system may still produce plausible responses even

Table 3: Metric definitions used in §5 (higher is better).

Group	Metric	Meaning (high-level)
Answer (overlap)	str_em	Exact-match style score between the generated long answer and reference (normalized).
Answer (overlap)	str_hit	Hit-style score indicating whether key reference answer content is covered.
Answer (overlap)	RougeLsum	ROUGE-L overlap with the reference long answer.
Answer (QA)	QA-EM	Exact match on QA sub-questions (averaged).
Answer (QA)	QA-F1	Token-level F1 on QA sub-questions (averaged).
Answer (QA)	QA-Hit	Hit rate indicating whether QA-relevant facts are covered.
Distribution	Mauve	Distributional similarity between generations and references.
Evidence (main)	citation_rec	Coverage of required/supporting citations (recall).
Evidence (main)	citation_prec	Correctness of produced citations (precision).

when the supporting evidence becomes incomplete or difficult to verify. This risk is especially important in customized knowledge assistants and other evidence-sensitive settings, where users may rely not only on the final answer but also on the traceability of its supporting sources. We therefore emphasize that efficiency-oriented interventions in RAG should not be evaluated solely by answer quality, but also by whether they preserve verifiable evidence for downstream use. At the same time, prompt compression may reduce token usage and computational cost, which can improve accessibility and lower environmental burden when applied responsibly. In this work, we study these issues in a controlled research setting using a public benchmark, and our results are intended to improve understanding of the trade-off between efficiency and verifiability rather than to justify immediate deployment in high-stakes domains. Systems based on compressed evidence-aware RAG should therefore be used with appropriate human oversight, especially in settings where incomplete grounding could lead to harmful or overconfident decisions.

Table 4: ASQA case study showing how evidence compression changes the *supporting spans* available for citation. Ret.=1.00 preserves sentence-level evidence, while Ret.=0.20 collapses it into sparse keywords, weakening verifiability even when the answer remains plausible.

	No compression (Ret.=1.00)	Aggressive compression (Ret.=0.20)
Question	Q: Where was <i>Snow White and the Huntsman</i> filmed?	
Retrieved evidence	<p>Evidence excerpt (doc: <i>Snow White and the Huntsman</i>):</p> <p>“Photography took place in the United Kingdom. The beach scenes were predominantly filmed in Pembrokeshire, on the Marloes Sands beach near the village of Marloes between September 26 and 29, 2011. . . . A computer-generated castle was set on nearby Gateholm island. . . . The film used academic consultants from the University of Chichester.”</p>	<p>Compressed evidence excerpt (same doc, Ret.=0.20):</p> <p>“United Kingdom Pembrokeshire Marloes Sands September 26 29 2011 castle Gateholm field ramp University Chichester”</p>
Generated answer	<p>Answer excerpt:</p> <p>“The beach scenes were filmed in Pembrokeshire, on Marloes Sands near the village of Marloes (Sep. 26–29, 2011) . . .”</p> <p>Citation behavior: sentence-level evidence directly supports fine-grained details (location + date).</p>	<p>Answer excerpt:</p> <p>“<i>Snow White and the Huntsman</i> was filmed in the United Kingdom in Pembrokeshire, Marloes Sands, and Chichester University [1].”</p> <p>Citation behavior: evidence becomes keyword-like; it is harder to ground specific claims to precise supporting spans.</p>
Observation	<p>What this case shows. Under Ret.=1.00, the model can (in principle) attach citations to complete supporting spans that contain the <i>exact</i> filming details (locations, date range, and related context). Under Ret.=0.20, the retained evidence largely preserves <i>entities</i> but loses sentence structure and surrounding context, making citations less informative and less precise for verification (consistent with the larger drop in cRec/cPrec than in answer-level metrics).</p>	

When Gradients Collide: Failure Modes of Multi-Objective Prompt Optimization for LLM Judges

Parth Darshan^{◇*}

[◇]IIT Jodhpur

b22cs040@iitj.ac.in

Abhishek Divekar^{♠†*}

[♠]Amazon

adivekar@amazon.com

Abstract

Customizing an LLM judge to a specific problem or domain often involves optimizing its prompt across multiple evaluation criteria simultaneously. Textual gradient methods automate this for a single judge criterion, however they produce natural-language critiques, not numerical vectors. Thus, the conflict-resolution toolkit of multi-task learning (PC-Grad, MGDA) does not apply to this multi-objective textual gradient setting. We extend TextGrad to the multi-objective setting and test four decomposition modes of textual gradient optimizers by varying how much cross-objective information the loss, gradient and optimizer LLMs share. We find the gradient’s task-focus drops by 59% (9.0 to 3.7 out of 10) when the gradient LLM must provide feedback on multiple criteria jointly. Separately, we observe that naively combining single-objective optimized instructions into a single prompt degrades Spearman ρ from 0.305 to 0.220 (-0.085). These results identify two separable failure modes: optimization-time *gradient dilution* and inference-time *instruction interference*, which together constrain the design space for multi-objective judge optimization using textual feedback.¹

1 Introduction

Modern judges, whether human or LLMs, are used to evaluate text along multiple quality dimensions at once. SUMMEVAL (Fabbri et al., 2021) scores summaries on four dimensions simultaneously; MT-Bench (Zheng et al., 2023) spans eight categories from coding to roleplay. Kim et al. (2024) developed Prometheus 2 specifically because existing evaluators “do not possess the ability to evaluate based on *custom evaluation criteria*”. Prompt

optimization methods like TextGrad (Yüksekgönül et al., 2025), OPRO (Yang et al., 2024), and GEPA (Agrawal et al., 2026) can improve prompts for LLM judges automatically, but they optimize a single objective. Whether they extend to multi-objective judges, where one prompt must score several evaluation dimensions at once, remains less studied.

The core challenge is structural. When multi-task deep learning models encounter conflicting task gradients, methods like PCGrad (Yu et al., 2020) and MGDA (Sener and Koltun, 2018) resolve conflicts via projection or constrained optimization. The conflict-resolution tools of numerical multi-task learning do not apply directly, because textual gradients lack the vector-space structure these methods require. As textual gradients are natural-language strings, they do not have equivalent concepts of magnitude, inner products, or linear subspaces. The instruction “Make the coherence rubric more specific” cannot be numerically projected against “simplify the fluency criteria”.

Prior work addresses related problems but not this intersection. For example, intra-task feedback quality has been studied: recently Chu et al. (2026) identify *rule dilution*, when heterogeneous error modes are aggregated within a single optimization step, and Melcer et al. (2025) show that the gradient analogy does not accurately explain why Automatic Prompt Optimization (APO) methods succeed. Multi-objective prompt optimization also exists: Jafari et al. (2024) find that MGDA underperforms volume-based alternatives for discrete prompt tokens. Multi-task judge *training* is studied extensively (Whitehouse et al., 2026; Yang et al., 2026; Wang et al., 2025), but these methods update weights, not prompts. To our knowledge, no prior work studies multi-objective judge *prompt* optimization using textual gradients.

We propose and test four plausible decomposition modes of textual gradient optimization for

^{*}Equal contribution. [†]Research lead and corresponding author. Author was employed at Amazon when this work was conducted, but the work was performed independently and did not use Amazon data, resources or confidential information.

¹Code, data and prompts available at <https://github.com/adivekar-utexas/when-gradients-collide>

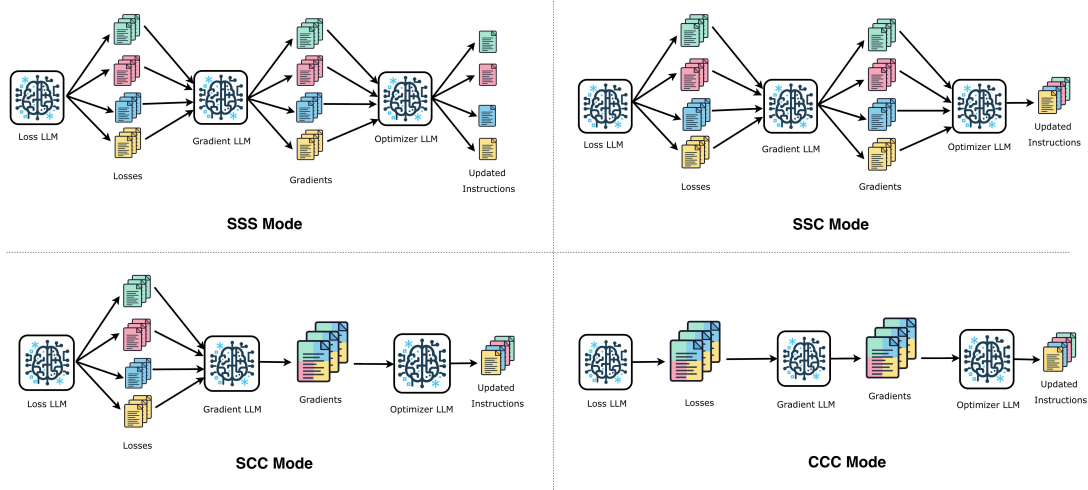


Figure 1: Overview of the optimization pipeline. Each step consists of four stages: (1) the task model predicts scores, (2) the loss LLM critiques predictions against ground truth, (3) the gradient LLM converts critiques into instruction edits, and (4) the optimizer LLM rewrites the prompt. The decomposition mode determines whether each stage operates per-task (Separate) or over all tasks jointly (Combined). Only the per-task instruction text is updated; the prompt skeleton and output format remain frozen throughout optimization.

multi-objective judges: SSS, SSC, SCC, CCC, where the naming encodes whether each pipeline stage (loss, gradient, optimizer) processes tasks separately (S) or combined (C). We also introduce two process-level diagnostics: *gradient specificity* (how targeted each gradient is to a single task) and *feedback adherence* (whether the optimizer follows the gradient). We evaluate on SUMMEVAL (Fabbri et al., 2021) with four criteria, using two validation gate settings (MAE gating and no gating).

Our results reveal a consistent structure in how and why multi-criteria prompt optimization stagnates. In 6 of 10 configurations with Qwen3 judges, optimization never exceeds the initial generic prompt (§5, Table 1); only SINGLE-TASK with val=mae improves (+0.031 Spearman at step 2). Our diagnostics localize this effect: gradient specificity drops by 59% (9.0 to 3.7) when the gradient LLM processes all tasks jointly §6.1, Table 4). Feedback adherence remains high (7.8 to 8.8), indicating that the optimizer incorporates the gradient’s suggestions regardless of their specificity. A separate oracle experiment shows that even independently optimal per-task instructions *degrade* from 0.305 to 0.220 (−0.085 Spearman) when combined into one prompt (§6.3, Table 3). These results identify two separable failure modes: optimization-time *gradient dilution* and inference-time *instruction interference*.

Our contributions are as follows: (1) we present an empirical study of multi-criteria textual gradi-

ent optimization for LLM judges across four decomposition modes (§4, §5, Table 1); (2) we propose two process-level diagnostics (*gradient specificity* and *feedback adherence*) that localize issues in optimization to gradient quality, not optimizer compliance (§6.1, Figure 3, Table 2). (3) we conduct an oracle-instruction experiment which isolates inference-time *instruction interference* as a failure mode distinct from optimization-time dilution (§6.3, Table 3).

2 Related Work

Textual Gradient Methods. Early methods in Prompt optimization used scalar signals: OPRO (Yang et al., 2024) rewrites prompts by conditioning on (prompt, score) histories, and APE (Zhou et al., 2023) generates candidates from demonstrations and selects the highest-scoring variant. PROTEGI (Pryzant et al., 2023) replaced scalar signals with *textual gradients*: natural-language critiques that guide prompt rewrites. TextGrad (Yüksükönül et al., 2025) extended this to multi-component computation graphs, propagating critiques through LLM pipelines in analogy to back-propagation.

Subsequent work has questioned the gradient analogy. GPO (Tang et al., 2025) decomposed textual optimization into two factors (update direction and update method), drawing analogies to gradient, momentum, and learning rate, but found that adding a reflection step hurts performance. Melcer

et al. (2025) showed empirically that the gradient decomposition (the chain-rule structure central to the analogy) does not consistently improve prompt optimization, and that the gradient metaphor does not accurately explain why APO methods succeed. GEPA (Agrawal et al., 2026) evolves language programs via reflective mutation with Pareto selection. To our knowledge, all methods above optimize a single objective and none provide a mechanism to observe or control how per-task feedback interacts during optimization.

Multi-Objective Prompt Optimization. Multi-objective prompt optimization is a nascent area with two distinct approaches. Population-based methods maintain a Pareto front of candidate prompts: MOPO (Menchaca Resendiz and Klinger, 2025) applies NSGA-II (Deb et al., 2002) with LLM-based mutation to affective text generation, and ParetoPrompt (Zhao et al., 2025) decomposes objectives into scalarized subproblems. MORL-Prompt (Jafari et al., 2024) adapts multi-objective reinforcement learning to discrete prompt tokens but found that MGDA (Sener and Koltun, 2018) underperforms the simpler product-of-rewards at balancing competing objectives. Evolutionary approaches include EVOPROMPT (Guo et al., 2024) for single-objective and Baumann and Kramer (2024) for multi-objective sentiment balancing. These methods operate at the *candidate-selection* level: they choose which prompts to keep from a population. To our knowledge, none of the above studies how per-task feedback interacts *within* a single textual gradient trajectory, which is the setting we investigate.

LLM-as-a-Judge Evaluation. MT-Bench and Chatbot Arena (Zheng et al., 2023) established the paradigm of LLM-based evaluation where human annotation is expensive. Subsequent works use a fixed prompt to improve judge quality through tuning weights on rubric-grounded data (Kim et al., 2024), reinforcement learning (Whitehouse et al., 2026), debiasing (Yang et al., 2026), and preference optimization (Wang et al., 2025). A parallel line of work optimizes the evaluation prompt itself. CARO (Chu et al., 2026) identifies *rule dilution* when heterogeneous error modes are aggregated into a single optimization step. RRD (Shen et al., 2026) recursively decomposes rubrics to improve coverage and remove redundancy. MPO (Sharma and Henley, 2026) applies section-local textual gradients to individual prompt components (role, con-

text, constraints) independently. MAPGD (Han et al., 2025) coordinates multiple gradient agents, using semantic similarity to resolve conflicting edits. All these operate on a single evaluation objective, or decompose along axes other than evaluation criteria. None jointly optimizes a judge prompt across multiple criteria while preserving per-task gradient observability (i.e., the ability to trace how each criterion’s feedback shaped each rewrite), the setting we study.

3 Method: Multi-objective TextGrad

TextGrad pipeline. We implement a 4-stage optimization loop based on TextGrad (Yüksekönlü et al., 2025). At each step, a **task LLM** predicts objective-scores for a minibatch of examples using the current prompt. A **loss LLM** critiques the predictions by comparing them to ground-truth annotations and produces a natural-language “loss” for each example (called a *reflection* by some authors). A **gradient LLM** aggregates the per-example losses into a “textual gradient”: a structured set of instruction-level edit suggestions. We restrict gradients to 3 paragraphs. An **optimizer LLM** rewrites the prompt based on the gradient. Only the per-objective instructions are modified; the prompt skeleton (role preamble, output format, few-shot examples) is frozen throughout (see Appendix E for the full initial prompt and an example of optimized instructions). This design isolates the effect of objective-level instruction modifications from possible confounds introduced by structural prompt changes.

Decomposition modes. We extend TextGrad to the multi-objective setting stagewise. As shown in Figure 1, we parameterize the multi-objective interaction via a 3-letter *decomposition code*. Each letter denotes whether a stage operates in **Separate** (per-task) or **Combined** (all-tasks-joint) mode, applied to the loss, gradient, and optimizer stages respectively.

Four modes span the design space. **SSS** operates all three stages independently per task. **SSC** computes loss and gradient per task, but the optimizer receives all four gradients simultaneously and rewrites the prompt in a single call. **SCC** computes only the loss per task; the gradient LLM receives critiques from all four tasks and produces a unified set of edits. **CCC** operates all stages jointly. We also include a **Single-Task** baseline in which each objective is optimized in a completely inde-

Mode	MAE validation				No validation			
	Initial	Best (step)	Δ	HVI	Initial	Best (step)	Δ	HVI
SINGLE-TASK	0.274	0.305 (2)	+0.031	—	0.269	0.284 (5)	+0.015	—
SSS	0.284	0.284 (0)	+0.000	2.749	0.283	0.283 (0)	+0.000	2.867
SSC	0.289	0.289 (0)	+0.000	2.832	0.283	0.291 (2)	+0.007	2.845
SCC	0.282	0.282 (0)	+0.000	2.801	0.282	0.282 (0)	+0.000	2.779
CCC	0.285	0.296 (9)	+0.012	2.900	0.287	0.287 (0)	+0.000	2.983

Table 1: Main results on SUMMEVAL using Qwen3. We measure the task-averaged Spearman correlation (mean over $N=3$ runs). Δ is the improvement from the initial prompt to the best step. HVI is the hypervolume indicator at step 12 (higher = more Pareto-diverse prompts). We evaluate four decomposition modes (SSS through CCC) plus a single-task optimization baseline (SINGLE-TASK), covering the full spectrum from separate to joint optimization. In 6 of 10 configurations, the initial generic prompt (“Rate from 1 to 5”) is never exceeded.

pendent run, and evaluated through its own forward pass; i.e., the reported Spearman is the average of four per-criterion evaluations, each using a prompt specialized to a single objective.

The key architectural boundary lies between SSC and SCC. In SSC, the gradient LLM sees one task at a time. In SCC, it must reconcile feedback from all four tasks into a coherent edit plan.

Validation Gating. We evaluate two validation strategies. Under `val=mae`, a candidate prompt is accepted only if its mean absolute error on a held-out validation set does not exceed that of the current prompt. This acts as a monotonic filter that prevents prompt-regression. Under no validation gate, every candidate is accepted unconditionally; we can observe the optimization trajectory without gating. For each configuration, we run 3 independent trials with different random seeds, each for 12 optimization steps. We report the mean and standard deviation of task-averaged Spearman $\bar{\rho}$ across the 3 runs.

4 Experimental Setup

Datasets. We evaluate on SUMMEVAL (Fabbri et al., 2021), a summarization meta-evaluation benchmark. The original dataset contains source news articles, each paired with summaries from 16 different summarization systems, with expert annotations from multiple annotators per pair. We randomly subsample this into 160 pairs for training (used as optimization batches) and 480 pairs for held-out evaluation. Each pair is scored on four dimensions (fluency, relevance, coherence, consistency) on a 1-5 scale.

These four dimensions are the tasks in our multi-task optimization setting: the judge prompt must produce accurate scores across all four simultaneously. We report Spearman rank correlation (ρ)

between predicted and human scores as the primary metric, following prior work on LLM-based evaluation (Liu et al., 2023). Unless stated otherwise, all reported results are *task-averaged* Spearman, the arithmetic mean of per-task ρ values across the four dimensions.

Models. Our main results are evaluated on the Qwen3 family (Yang et al., 2025) using Qwen3-8B as the task LLM and Qwen3-235B-A22B as others. We use a higher optimizer temperature ($T = 0.7$) to encourage diverse rewrites, and a lower loss and gradient temperatures ($T = 0.3$) promote consistent critiques. For the task LLM we set $T = 1.0$ to allow resampling in case of JSON formatting errors. In Appendix D, we show results on DeepSeek v4 Flash and Pro (DeepSeek-AI, 2026).

5 Results

Table 1 reports task-averaged Spearman ρ for each decomposition mode and validation configuration, averaged over $N = 3$ runs. In 6 of 10 configurations, the best prompt is the initial generic prompt (“Rate from 1 to 5”): optimization either fails to improve or actively degrades performance. The only multi-task mode that improves is CCC with MAE validation and SSC without validation, which achieve modest gain over 12 steps. Only the single-task baseline with MAE validation achieves meaningful improvement: +0.031 Spearman at step 2 (Table 1, **bold**). This confirms that the TextGrad pipeline can improve individual-task prompts when gradient signal is not contaminated by cross-task information.

We plot optimization trajectories in Figure 2, revealing the dynamics behind these aggregates. Without a validation gate (`val=none`, bottom row), SSC drops from 0.283 to 0.184 by step 7; SCC

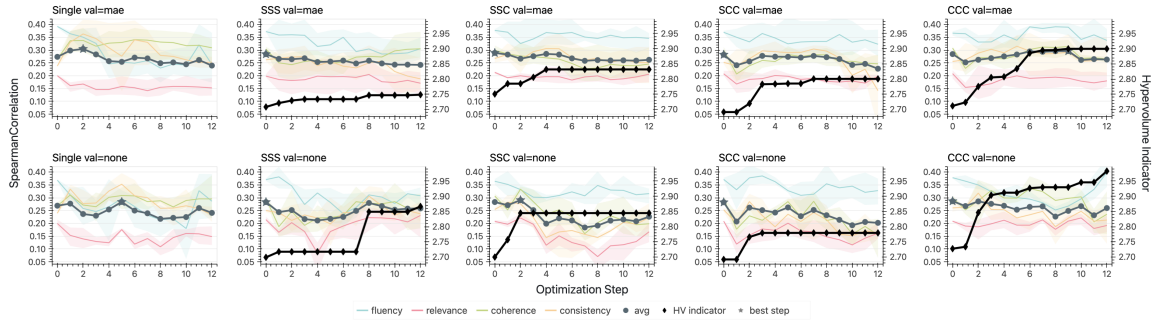


Figure 2: Per-task Spearman ρ for each optimization steps on SUMMEVAL with Qwen3. We average over $N = 3$ runs (shaded bands show min to max). Each column shows one of the five decomposition modes. On the top row we apply validation-MAE to gate prompts at each step, while bottom row has no gating. Gray line indicates task-averaged ρ ; stars mark best step. Black diamonds (right axis) denote the hypervolume indicator for the prompt-candidates accumulated over steps. Without validation gating, multi-task modes degrade, and with it, they plateau.

shows comparable degradation. The optimizer proposes changes that improve the training-batch loss but harm held-out generalization. MAE validation partially mitigates this by rejecting spurious updates. With the validation gate active (top row), trajectories flatten rather than decline: the gate prevents catastrophic degradation but cannot compensate for uninformative gradients.

Performance degrades roughly as Single-Task $>$ SSS, SSC, SCC, CCC without validation gating. This is consistent with the hypothesis that increasing cross-task coupling amplifies gradient dilution. The pattern is non-monotonic under MAE validation, however: CCC slightly outperforms SSC. Full coupling may occasionally produce complementary gradients that survive the validation gate. The process-level diagnostics in §6 disentangle these effects.

Despite stagnation in Spearman, the hypervolume indicator (HVI) shows an increasing trend. For CCC, HVI grows continuously; the optimizer discovers diverse specialist prompts that expand the Pareto front, even when no single prompt dominates the initialization on all four tasks. [Menchaca Resendiz and Klinger \(2025\)](#) report a similar pattern: multi-objective prompt optimization expands the Pareto front at modest per-objective cost. In our setting, however, the per-task degradation is substantially larger when coupling is high.

We rerun with DeepSeek v4 under MAE validation in Appendix D, where we observe better absolute improvements in Spearman through optimization due to stronger models, but a similar trend: the *improvement* through prompt optimization is most effective for single-objective optimizations.

6 Analysis

The results in §5 establish *that* multi-objective textual gradient prompt optimization lags compared to single-task; this section investigates *why*. We identify two separable failure modes: optimization-time *gradient dilution* (§6.1) and inference-time *instruction interference* (§6.3). We report the evaluation prompts in Appendix F for reproducibility.

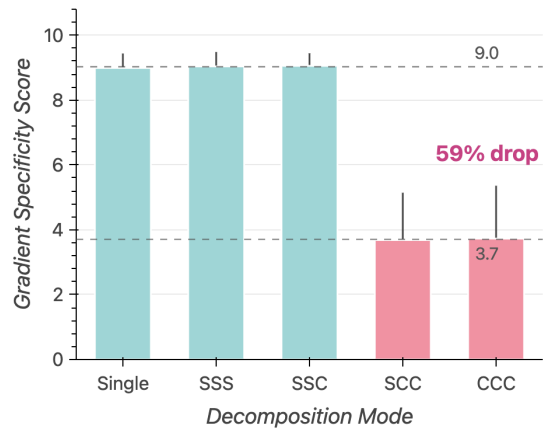


Figure 3: Gradient specificity (1 to 10 scale, higher is more task-focused) by decomposition mode.

6.1 Gradient Specificity and Dilution

To measure gradient quality directly, we evaluate each textual gradient for *task-focus*: the degree to which its improvement suggestions target a single evaluation criterion rather than offering generic advice. An LLM evaluator (Claude Sonnet 4.6, [Anthropic \(2026\)](#)) rates each gradient on a 1 to 10 scale. A score of 10 means the gradient addresses exactly one task’s rubric; a score of 1 means it’s so generic it could apply to any task (see Appendix F for the evaluation prompt). This diagnostic tests

Mode	MAE validation	No validation
Single-Task	8.70 \pm 0.47	8.53 \pm 0.60
SSS	8.84 \pm 0.44	8.72 \pm 0.54
SSC	7.94 \pm 0.80	7.83 \pm 0.78
SCC	8.08 \pm 1.03	7.76 \pm 1.49
CCC	7.90 \pm 1.62	8.01 \pm 1.28

Table 2: Feedback adherence scores (1 to 10 scale, mean \pm std over all objectives, runs, and steps). All modes achieve high adherence, ruling out optimizer non-compliance as an explanation for multi-task optimization failure.

the cross-objective analogue of [Chu et al.](#)’s rule-dilution hypothesis. Concretely, we ask: does combining multiple evaluation criteria in a single gradient call dilute the criteria-specific signal?

We evaluate all gradients from steps 1 to 12 across all four modes. A sharp transition is present (Figure 3). Per-task modes (Single, SSS, SSC), where each gradient LLM call processes exactly one task, achieve a mean specificity of 9.0 (\pm 0.3; Table 4, top rows). Cross-task modes (SCC, CCC), where the gradient LLM processes all four tasks in a single call, drop to 3.7 (\pm 0.5; Table 4, bottom rows), a 59% reduction with zero overlap between the two groups.

Our gradient dilution results extend [Chu et al.](#)’s within-criterion finding to the cross-criterion setting. [Chu et al. \(2026\)](#) show that aggregating heterogeneous error modes in a single optimization step degrades rubric accuracy; we observe an analogous effect when multiple task gradients are combined in a single gradient call, degrading the per-task optimization signal. A per-task breakdown (Table 4) reveals that consistency is most diluted, while coherence retains moderate focus. This possibly occurs because coherence rubrics share surface-level vocabulary with the generic “writing quality” feedback the gradient LLM defaults to under multi-task load.

To confirm this cliff is structural when moving from per-task to cross-task gradients, in Appendix C we switch *only* the gradient LLM to a stronger LLM (DeepSeek-V4-Pro) while continuing to use Qwen3 for the other steps. We observe the same SSC to SCC specificity drop. Appendix D.2 replicates the overall gradient-specificity analysis using DeepSeek for all modes, showing the same per-task vs. cross-task gradient dilution pattern.

Method	Fl.	Rel.	Coh.	Con.	Avg ρ
Initial	0.366	0.208	0.308	0.256	0.284
Single-Task	0.350	0.168	0.338	0.363	0.305
Cherry-pick	0.303	0.257	0.215	0.105	0.220

Table 3: Oracle cherry-pick experiment (MAE validation). For each task, the single-task instruction with the highest test-set metric is selected and combined into one multi-task prompt. Despite oracle selection, combined instructions degrade below the individually optimized single-task performance, demonstrating inference-time instruction interference. We report both task-level and average ρ for the multi-task prompt.

6.2 Feedback Adherence

We also measure *feedback adherence*: the degree to which the optimizer LLM incorporates the textual gradient into its instruction edits (Table 2). We prompt Claude Sonnet 4.6 evaluator to output an adherence score on a 10-point scale (refer Appendix F for the prompt).

Across all modes and validation settings, adherence is uniformly high (7.8 to 8.8 on a 10-point scale), confirming that the optimizer faithfully implements whatever gradient it receives, even when those suggestions are generic rather than criterion-specific. This indicates the ceiling to multi-objective judge prompt optimization is gradient quality rather than optimizer compliance.

6.3 Inference-Time Instruction Interference

Gradient dilution explains why SCC and CCC fail to optimize effectively, but it does not explain why SSS and SSC optimizations also stagnate. These per-task modes produce highly specific gradients and high-adherence prompt updates, yet still fail to improve over the initialization. To diagnose this, we design an oracle cherry-pick experiment that isolates the *inference-time* component of multi-task failure. For each task, we select the single best instruction across all single-task optimization runs (the instruction with the highest held-out Spearman ρ for that task), then combine these four oracle-optimal instructions into one multi-task prompt and evaluate on the full test set.

We see that oracle-optimal instructions *degrade* when combined (Table 3), achieving 0.22 average Spearman, a drop of 0.085 from the single-task optimized performance of 0.305. In Appendix B we explore more variants by using different metrics to select the best instruction per task, and observe the same effect. These instructions each individually outperform the baseline on their respective tasks,

but combining them produces performance strictly worse than the generic initial prompt.

The primary mechanism appears to be instruction-length asymmetry. Optimization produces over-specified rubrics for some tasks (the fluency instruction expands to ~ 800 tokens with detailed scoring anchors) while leaving others under-specified (the relevance instruction remains at ~ 4 tokens of the initial prompt). When packed into a single prompt, verbose instructions receive disproportionate attention relative to brief ones at inference time.

This finding strengthens a result from Shen et al. (2026), who observe that naive rubric construction degrades GPT-4o preference-judgment accuracy by 13 points on JudgeBench. The degradation we observe is larger and occurs with oracle-selected rather than naively constructed instructions. Shen et al.’s result shows that *bad* rubrics hurt. Ours shows that individually *good* rubrics can hurt when combined. This implies that instruction interference is not resolvable by improving per-task optimization alone.

7 Conclusion

Multi-criteria textual gradient optimization for LLM judges exhibits two failure points that our decomposition study and process-level diagnostics expose. These are systematic failures, corresponding to distinct pipeline stages and affect different decomposition modes: (i) Gradient dilution operates at optimization time. When the gradient LLM must reconcile feedback from multiple criteria in a single call, its suggestions lose task-specificity (59% drop; Appendix Table 4). The optimizer propagates the low-specificity signal through to the final prompt; (ii) Instruction interference operates at inference time. Independently optimized per-task instructions degrade when combined into one prompt (Table 3) because instruction-length asymmetry causes verbose rubrics to receive disproportionate attention relative to brief ones.

For practitioners customizing judges to domain-specific criteria (Kim et al., 2024), these results indicate architectural changes are required before the multi-objective setting can work reliably. Addressing either failure mode alone is insufficient. Separate judge calls per criterion eliminates interference but multiplies inference cost. Conflict-aware gradient resolution adapted from numerical multi-task learning (Yu et al., 2020; Liu et al., 2021) could address dilution if textual gradients can be mean-

ingfully embedded and projected. Our proposed diagnostics (gradient specificity and feedback adherence) provide the measurement tools to evaluate either approach.

8 Future Work

Our findings open several concrete directions to broader research on customized LLM evaluation.

Statistically reliable LLM diagnostics with PPI.

Our diagnostics are LLM-judged, introducing an evaluator-bias. Prediction-Powered Inference (Angelopoulos et al., 2023; Boyeau et al., 2025) combines a small human-judged set with a large LLM-judged set to produce provably unbiased estimates; the hierarchical PPI extension of Divekar and Majumder (2026) is directly applicable here, since annotations are per-gradient but the quantity of interest is the per-mode mean. This would allow us to report the specificity gap as a confidence interval and scale diagnostics to hundreds of runs without scaling human annotation linearly.

Synthetic task generation for aligned criteria.

A new direction is to *synthesize the criteria themselves* rather than treat them as fixed. Persona-driven synthesis at billion-person scale (Chan et al., 2024; Yu et al., 2023) and instruction-data generation with controlled diversity (Divekar and Durrett, 2024; Kowshik et al., 2024) can produce synthetic objectives that are complementary for optimization. If the synthesized tasks are mutually aligned by construction, the combined gradient should suffer less semantic drift, raising gradient focus and potentially mitigating both failure modes at their source.

Multi-objective critics for agentic workflows.

Multi-objective judge prompts could serve as critics in agentic LLM systems, where a critic must track several quality dimensions of tool-use trajectories at once (Ding, 2026; Chuang et al., 2026; Rudman et al., 2026). Applying our optimization pipeline to such critics raises an open question: do gradient dilution and instruction interference still emerge when criteria are tool-grounded and partially verifiable, or does verifiability yield more robust gradients? Our diagnostics provide a principled way to measure this.

Mitigations. Our diagnostics motivate concrete mitigations. For *gradient dilution*, a specificity-aware router could fallback to per-task gradient LLM when a multi-task LLM specificity drops

below threshold, capturing CCC’s hypervolume without losing task-focus. For *instruction interference*, we propose two avenues: (i) next-token attention masking for per-criterion output generation (eliminating interference as no cost), and (ii) length-aware instruction synthesis that normalizes rubric length during optimization so no single criterion dominates the attention budget.

Limitations

Our experiments are scoped to SUMMEVAL as it provides expert human annotations on four clearly separable criteria. Other benchmarks would validate our identified failure modes: BRIGHTER (Muhammad et al., 2025) tests whether dilution scales with task count and crosses language boundaries; ASAP++ (Mathias and Bhat-tacharyya, 2018) (per-trait essay-grading rubrics) tests whether the SCC to CCC gradient specificity cliff transfers beyond summarization; EM-SCAD (Vidros et al., 2017), GitBugs (Patil et al., 2026), test heterogeneous classification criteria rather than ordinal quality scales.

Other prompt optimization paradigms may exhibit different multi-task dynamics, though our diagnostics are algorithm-agnostic and can be applied to any textual gradient approach.

Our sample size ($N = 3$ runs) limits statistical power, and we restrict our claims to effect sizes that are robust at this sample size (e.g., the 59% specificity drop and -0.085 cherry-pick degradation). Finally, gradient specificity and feedback adherence are scored by an LLM evaluator, which introduces a potential confound.

Ethics Statement

Optimized judge prompts inherit biases present in the underlying LLMs and in the human annotations used for evaluation; practitioners should audit optimized prompts before deploying them in sensitive evaluation contexts (e.g., content moderation or hiring). Our code and diagnostics will be released under an open-source license to support reproducibility.

References

Lakshya A. Agrawal, Shangyin Tan, Dilara Soyulu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Daniel Klein, Matei Zaharia,

and Omar Khattab. 2026. [GEPA: Reflective prompt evolution can outperform reinforcement learning](#). In *The Fourteenth International Conference on Learning Representations*.

Anastasios N. Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnica. 2023. [Prediction-powered inference](#). *Science*, 382(6671):669–674.

Anthropic. 2026. [Introducing claude sonnet 4.6](#). Anthropic Blog.

Jill Baumann and Oliver Kramer. 2024. [Evolutionary multi-objective optimization of large language model prompts for balancing sentiments](#). In *Applications of Evolutionary Computation (EvoApplications)*, pages 212–224. Springer.

Pierre Boyeau, Anastasios Nikolas Angelopoulos, Tianle Li, Nir Yosef, Jitendra Malik, and Michael I. Jordan. 2025. [Autoeval done right: Using synthetic data for model evaluation](#). In *Forty-second International Conference on Machine Learning*.

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. [Scaling synthetic data creation with 1,000,000,000 personas](#). *arXiv preprint arXiv:2406.20094*.

Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, Joseph Krajcik, Namsoo Shin, and Jiliang Tang. 2026. [Confusion-aware rubric optimization for LLM-based automated grading](#). *arXiv preprint arXiv:2603.00451*.

Yun-Shiuan Chuang, Chaitanya Kulkarni, Alec Chiu, Avinash Thangali, Zijie Pan, Shivani Shekhar, Yirou Ge, Yixi Li, Uma Kona, Linsey Pang, and Prakhar Mehrotra. 2026. [Toward scalable verifiable reward: Proxy state-based evaluation for multi-turn tool-calling llm agents](#). *Preprint*, arXiv:2602.16246.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197.

DeepSeek-AI. 2026. [Deepseek-v4: Towards highly efficient million-token context intelligence](#).

Liang Ding. 2026. [Adarubric: Task-adaptive rubrics for reliable llm agent evaluation and reward learning](#). *Preprint*, arXiv:2603.21362.

Abhishek Divekar and Greg Durrett. 2024. [Synthesizr: Generating diverse datasets with retrieval augmentation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12–16, 2024*, pages 19200–19227. Association for Computational Linguistics.

Abhishek Divekar and Anirban Majumder. 2026. [PRECISE: Reducing the bias of LLM evaluations using prediction-powered ranking estimation](#). In *Proceedings of the AAAI Conference on Artificial Intelligence (IAAI Track)*, volume 40, pages 39929–39938.

- Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. [SummEval: Re-evaluating summarization evaluation](#). *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. [Connecting large language models with evolutionary algorithms yields powerful prompt optimizers](#). In *The Twelfth International Conference on Learning Representations*.
- Yichen Han, Bojun Liu, Zhengpeng Zhou, Guanyu Liu, Zeng Zhang, Yang Yang, Wenli Wang, Isaac Shi, Yunyan, Lewei He, and Tianyu Shi. 2025. [MAPGD: Multi-agent prompt gradient descent for collaborative prompt optimization](#). In *NeurIPS 2025 Workshop on Scaling Environments for Agents*.
- Yasaman Jafari, Dheeraj Mekala, Rose Yu, and Taylor Berg-Kirkpatrick. 2024. [MORL-prompt: An empirical analysis of multi-objective reinforcement learning for discrete prompt optimization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9878–9889, Miami, Florida, USA. Association for Computational Linguistics.
- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. [Prometheus 2: An open source language model specialized in evaluating other language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.
- Suhas S. Kowshik, Abhishek Divekar, and Vijit Malik. 2024. [CorrSynth: A correlated sampling method for diverse dataset generation from LLMs](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 16076–16095. Association for Computational Linguistics.
- Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021. [Conflict-averse gradient descent for multi-task learning](#). In *Advances in Neural Information Processing Systems*, pages 18878–18890.
- Yang Liu, Dan Iter, Yichong Xu, Shuo Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-eval: Nlg evaluation using gpt-4 with better human alignment](#). *ArXiv*, abs/2303.16634.
- Sandeep Mathias and Pushpak Bhattacharyya. 2018. [ASAP++: enriching the ASAP automated essay grading dataset with essay attribute scores](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).
- Daniel Melcer, Qi Chen, Wen-Hao Chiang, Shweta Garg, Pranav Garg, and Christian Bock. 2025. [Textual gradients are a flawed metaphor for automatic prompt optimization](#). *arXiv preprint arXiv:2512.13598*.
- Yarik Menchaca Resendiz and Roman Klinger. 2025. [MOPO: Multi-objective prompt optimization for affective text generation](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5588–5606, Abu Dhabi, UAE. Association for Computational Linguistics.
- Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine de Kock, Nirmal Surange, Daniela Teodorescu, Ibrahim Said Ahmad, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino D. M. A. Ali, Ilseyar Alimova, Vladimir Araujo, Nikolay Babakov, Naomi Baes, Ana-Maria Bucur, Andiswa Bukula, and 29 others. 2025. [BRIGHTER: BRIDging the gap in human-annotated textual emotion recognition datasets for 28 languages](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8895–8916, Vienna, Austria. Association for Computational Linguistics.
- Avinash Patil, Siru Tao, and Aryan Jadon. 2026. [Gibugs: Bug reports for duplicate detection, retrieval augmented generation, triage, and more](#). *Preprint*, arXiv:2504.09651.
- Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. [Automatic prompt optimization with “gradient descent” and beam search](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.
- William Rudman, Abhishek Divekar, Kanishk Jain, Sebastian Joseph, Stella S. R. Offner, Matthew Lease, Kyle Mahowald, Greg Durrett, and Junyi Jessy Li. 2026. [Vesta: Visual exploration with statistical tool agents](#). *Preprint*, arXiv:2606.00384.
- Ozan Sener and Vladlen Koltun. 2018. [Multi-task learning as multi-objective optimization](#). In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Prith Sharma and Austin Z. Henley. 2026. [Modular prompt optimization: Optimizing structured prompts with section-local textual gradients](#). *arXiv preprint arXiv:2601.04055*.
- William F. Shen, Xinchu Qiu, Chenxi Whitehouse, Lisa Alazraki, Shashwat Goel, Francesco Barbieri, Timon Willi, Akhil Mathur, and Ilias Leontiadis. 2026. [Re-thinking rubric generation for improving LLM judge and reward modeling for open-ended tasks](#). *arXiv preprint arXiv:2602.05125*.
- Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, Siyuan Lu, Yaliang Li, and Ji-Rong Wen. 2025. [Unleashing the potential of large language models as prompt optimizers: Analogical analysis with gradient-based](#)

- model optimizers. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*.
- Sokratis Vidros, Constantinos Koliass, Georgios Kambourakis, and Leman Akoglu. 2017. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Future Internet*, 9(1):6.
- Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2025. Direct judgement preference optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1979–2009.
- Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Iliia Kulikov, and Swarnadeep Saha. 2026. J1: Incentivizing thinking in LLM-as-a-judge via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.
- Bo Yang, Lanfei Feng, Yunkui Chen, Yu Zhang, Xiao Xu, and Shijian Li. 2026. FairJudge: An adaptive, debiased, and consistent LLM-as-a-judge. *arXiv preprint arXiv:2602.06625*.
- Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pages 12028–12068.
- Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc.
- Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J. Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. *arXiv preprint arXiv:2306.15895*.
- Mert Yükekgönül, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639(8055):609–616.
- Guang Zhao, Byung-Jun Yoon, Gilchan Park, Shantenu Jha, Shinjae Yoo, and Xiaoning Qian. 2025. Pareto prompt optimization. In *The Thirteenth International Conference on Learning Representations*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Yongchao Zhou, Andrei Ioan Muresanu, Ziwon Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

A Per-Task Gradient Specificity

Table 4 shows gradient specificity broken down by task for the combined-gradient modes (SCC, CCC). Averaged across SCC and CCC, consistency is the most diluted dimension (specificity 2.5), while coherence retains moderate focus (5.0). This suggests that the gradient LLM’s attention is not uniformly distributed across tasks when processing them simultaneously.

Mode	Fl.	Rel.	Coh.	Con.	Avg
PER-TASK GRADIENT MODES					
Single	8.9	8.9	9.1	9.0	9.0
SSS	9.0	9.0	9.1	9.0	9.0
SSC	9.0	9.1	9.1	9.0	9.0
CROSS-TASK GRADIENT MODES					
SCC	3.0	4.3	4.8	2.6	3.7
CCC	3.2	4.3	5.1	2.4	3.8

Table 4: Gradient specificity by task (mean over both val settings, all runs and steps). Per-task modes maintain uniformly high specificity. In all-task modes, consistency is most diluted and coherence retains moderate focus.

B Cherry-Pick Experiment: All Variants

Table 5 shows all six cherry-pick variants (three selection metrics and two validation settings). All variants degrade below the initial generic baseline, confirming that inference-time instruction interference is robust across metric choices.

C Gradient Specificity Under Gradient-Model Swap

We swap the gradient LLM backbone to DeepSeek-V4-Pro (DeepSeek-AI, 2026) while keeping the task, loss, and optimizer LLMs in the Qwen3 family and holding the rest of the pipeline fixed. As shown in Figure 4, SSC remains high (8.82 ± 0.10) while SCC drops sharply (4.22 ± 0.26 ,

Validation	Selection metric	Avg ρ
Initial prompt (generic)		0.284
MAE	Spearman	0.220
MAE	Off-by-one	0.232
MAE	MAE	0.231
None	Spearman	0.120
None	Off-by-one	0.200
None	MAE	0.172

Table 5: All cherry-pick variants. Every oracle combination degrades below the initial generic baseline. The worst case (val=none, Spearman selection) additionally shows zero fluency correlation.

a 52% reduction), indicating that gradient dilution persists under a cross-family gradient model.

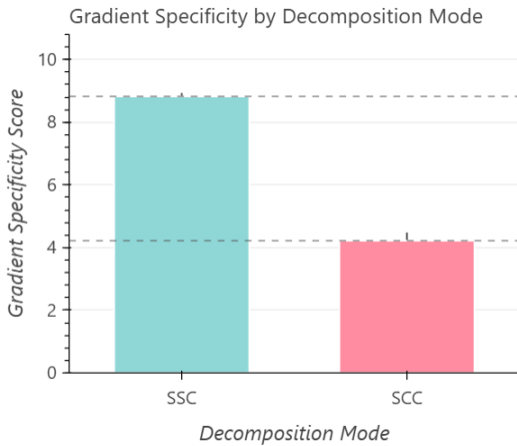


Figure 4: Gradient specificity for SSC vs. SCC after swapping the gradient LLM to DeepSeek-V4-Pro while keeping the other stages on Qwen3. SSC remains high (8.82 ± 0.10) while SCC drops to 4.22 ± 0.26 , a 52% reduction.

Mode	Initial	Best (step)	Δ	HVI
SINGLE-TASK	0.287	0.403 (9)	+0.117	—
SSS	0.384	0.421 (9)	+0.037	4.264
SSC	0.390	0.429 (7)	+0.039	4.436
SCC	0.379	0.418 (11)	+0.039	4.331
CCC	0.383	0.409 (7)	+0.026	4.282

Table 6: Main results under the MAE validation gate for the DeepSeek backbone. We measure the task-averaged Spearman correlation (mean over $N = 3$ runs). Notation same as Table 1.

D DeepSeek v4 Results

We repeat our analysis on DeepSeek v4 models (DeepSeek-v4-Flash as the task LLM; DeepSeek-V4-Pro for loss, gradient, and optimizer

LLMs). These provide a stronger starting point for the task-following and optimization steps. We use the MAE validation gate and run $N = 3$ seeds.

D.1 Overall Performance

Table 6 summarizes performance at step 12. We observe a stronger backbone for the task and optimization LLMs result in a superior increase from the initial prompt. Multi-objective modes start from a stronger initialization ($\rho \approx 0.38$) and reach best avg ρ between 0.409 and 0.429. SSC attains the best avg ρ and the highest HVI. However, we observe that the trend sustains as before, with the *improvement* through prompt optimization being highest for single-objective optimizations.

Figure 5 plots the per-task optimization trajectories. Unlike the Qwen3 setting, where multi-objective modes stagnated at or below the initial prompt (Figure 2), the DeepSeek backbone yields consistent improvement from initialization.

The pattern of improvement, however, mirrors the main results: Single-Task achieves the largest gain across steps, while the multi-objective modes show flatter trajectories. The stronger optimizer LLM lifts all modes substantially, but it does not close the gap between single-objective and multi-objective optimization.

Per-task traces remain bounded within narrower bands than their Qwen3 counterparts, consistent with the observation that a stronger task model produces more stable evaluation dynamics.

D.2 Gradient Specificity

With DeepSeek, the gradient dilution cliff persists: per-task gradient modes (Single/SSS/SSC) remain highly task-focused (mean 8.78 specificity), while joint-gradient modes (SCC/CCC) drop to mean 4.25 (a 52% reduction), indicating that combining criteria in a single gradient call yields substantially more generic feedback. As shown in Figure 6, gradient specificity remains high for per-task gradient modes, while joint-gradient modes exhibit a sharp drop under the DeepSeek configuration.

E Multi-Objective Judge Prompt Template

Our multi-criteria prompt has two parts: a **frozen skeleton** (evaluation directive, output format) and **mutable per-task instructions** (the “Instructions” section).

Only the per-task instructions are updated during optimization; the skeleton remains fixed through-

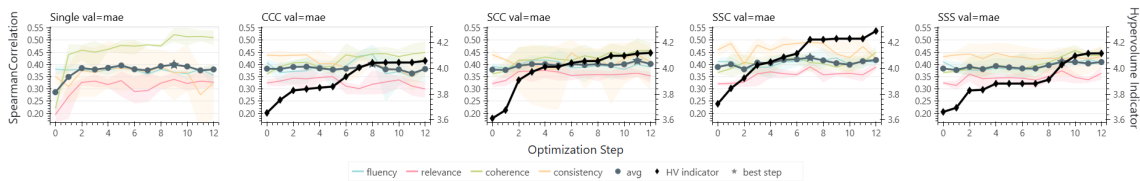


Figure 5: Per-task Spearman ρ for each optimization steps on SUMMEVAL with DeepSeek v4. Notation same as Figure 2.

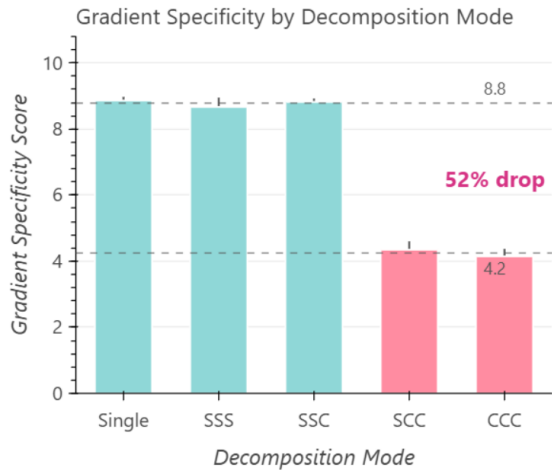


Figure 6: Gradient specificity by decomposition mode under DeepSeek.

out all 12 steps. Our skeleton is specific to SUMMEVAL but may be easily adapted to other problems.

Initial prompt template

You are a careful, calibrated evaluator. Your goal is to produce an accurate evaluation by following the Instructions below.

```
## Task
Evaluate the Summary given the Source Text using the Instructions below.
1. Consider every strength and flaw you find when making your evaluation.
2. Based on the number and severity of the strengths and flaws, assign a value.
Use the Instructions below to perform your evaluation.
Output a JSON with the requested scores. Do NOT include reasoning or explanations.
```

```
## Output format (follow this EXACTLY):
{
  "fluency": 1|2|3|4|5,
  "relevance": 1|2|3|4|5,
  "coherence": 1|2|3|4|5,
  "consistency": 1|2|3|4|5
}
```

```
## Instructions:
- fluency: Rate from 1 to 5.
- relevance: Rate from 1 to 5.
- coherence: Rate from 1 to 5.
- consistency: Rate from 1 to 5.
```

```
## Sample:
Summary: <summary text>
Source Text: <source text>
```

After optimization, the per-task instructions expand substantially. Below is an example from a SSS run (MAE validation, step 1), showing how the optimizer rewrites the generic one-line instructions into detailed, criterion-specific rubrics.

Example optimized instructions (step 1)

```
## Instructions:
- fluency: Fluency: Rate 1-5 based on grammatical correctness, sentence flow, and ease of parsing. A score of 5 reflects natural, effortless readability even with minor syntactic informality. Do not penalize for stylistic choices that do not impede comprehension.
- relevance: Relevance: Rate 1-5 based on how completely the summary addresses the core informational needs implied by the source text. A score of 5 means all key facts, outcomes, and implications are included with no deviation. Relevance penalizes both omission (missing key facts) and commission (adding unsupported claims).
- coherence: Coherence: Rate 1-5 based on logical flow, temporal sequence, and referential stability. A score of 5 means events unfold in a cause-effect or chronological order, with clear antecedents for all pronouns and noun phrases.
- consistency: Consistency: Rate 1-5 based on whether the summary contradicts any claim in the source text. A score of 5 requires no factual or inferential contradictions. A single major inconsistency reduces the score to 1.
```

F Diagnostic Evaluation Prompts

Below we include the prompts used for task-level diagnostics. Both diagnostics are evaluated post-hoc by Claude Sonnet 4.6.

F.1 Gradient Specificity Evaluator

The following prompt rates each textual gradient on a 1 to 10 scale for task-specificity:

Gradient specificity evaluator prompt

You are measuring how focused a piece of textual feedback (called a "gradient") is on a specific evaluation task, versus being diluted with generic advice or advice that belongs to other tasks.

The target task is "{task}". The possible tasks are: fluency, relevance, coherence, consistency.

Rate from 1 to 10 how well this gradient focuses on the "{task}" task.

1 = completely generic or mostly addresses other tasks.

10 = laser-focused on "{task}" with concrete, task-specific fixes.

The Gradient
{gradient_text}

Respond with ONLY a single integer from 1 to 10. No explanation.

F.2 Feedback Adherence Evaluator

The following prompt measures how well the optimizer incorporated each gradient into its instruction edit:

Feedback adherence evaluator prompt

You are evaluating whether revisions to task-specific instructions correctly addressed the gradient (suggested changes).

The instructions are for an LLM judge that evaluates the "{task}" task. The Gradient may contain suggestions about multiple tasks; consider only suggestions pertaining to "{task}".

Rate from 1 to 10 how well the New Instructions address the Gradient for "{task}".

1 = completely ignores/contradicts.

10 = precisely addresses every point while preserving what worked.

Old Instructions
{old_instruction}

New Instructions
{new_instruction}

Gradient (Suggested Changes)
{gradient_text}

Respond with ONLY a single integer from 1 to 10. No explanation.

Author Index

- Alikhani, Malihe, 167
Atwell, Katherine, 167
- Basu, Priyam, 1
Bernacchia, Alberto, 84
Bian, Kaiqi, 68
Buffelli, Davide, 84
- Cachola, Isabel, 97
Chan, David M., 191
Chelliah, Shobhana Lakshmi, 149
Chen, Bin, 202
Cioba, Alexandru, 84
Col, Naz, 191
- Darshan, Parth, 216
Divekar, Abhishek, 216
Dredze, Mark, 97
- Fort, Alexandra, 149
- Girdhar, Anubhav, 24
Gray, Morgan A., 167
Gusain, Jai Ranjan Singh, 68
- Jiang, Jiaojiao, 44
Joshi, Aditya, 44
- Karaca, H. Suhan, 44
Karakos, Damianos, 160
Keith, Francis, 160
Kosireddy, Tagore Rao, 11
- Lasko, Jonathan, 160
Li, Aiyu, 202
Li, Qiang, 24
Linardi, Sera, 167
Liu, Catherine, 131
Lopez, Kimberly, 131
Lucas, Evan, 11
- Maragoud, Raj, 68
Meisenbacher, Stephen, 117
Mishra, Suyash, 24
- Norlander, Peter, 117
- Øvrelid, Lilja, 55
- Paik, Hye-young, 44
Patil, Srikanth, 24
Peng, Qian, 202
- Raheja, Vipul, 1
Ramanathan, Manikandarajan, 68
Rial, Len, 167
- Sasse, Kuleen, 97
Savelka, Jaromir, 167
Schulz, Alexia E., 131
Singh, Amrita, 44
Singh, Avyav Kumar, 84
Steskal, Lubos, 55
Subramanian, Shreya, 131
- Tang, Adam, 131
Touileb, Samia, 55
- Uchendu, Adaku, 131
- Wall, Jeffrey David, 11
Wan, Jiajing, 55
Wu, Yen-Chen, 84
- Xie, Biancen, 68
- Yang, Zhengzhe, 182
- Zhang, Yunfeng, 1
Zinn-Brooks, Leif, 131