CS 1671 / CS 2071 / ISSP 2071
Human Language Technologies

Session 28: Project presentations

May 1, 2026

School of Computing and Information

Instructions

- Plan for **7-min presentations max** not including Q&A
- Cover at least these key points
  - Project motivation (briefly)
  - Task description, including example input and output
  - Data
  - Methods: baseline system and LLM-based system (or your equivalents)
  - Results and implications

Put your slides in this presentation after your project name slide by **class session, 10am on Fri May 1**

Schedule

1. Raymond, Chris, Rose, Forest, Joshua
2. Jay, Saung, Lyndsey, Amanda, Jonah
3. Praz, Wyatt, Justin, Pier, Ryan
4. Amyia, Sarina, Kiana, Michelle, Grace
5. Nihal, Kevin, Owen, Nate, Enzo, Matthew
6. Irisin, Yifei, Sanjana, Ciara, Astalaxmi
7. Marcus, Aaron, Patrick, Hongyu, Jack
8. David, Jeana, Griffin, Cole, Daley
9. Ryder, Ifemi, Vivien, Heather, Fatimah
10. Kee, Aidan, Hannah, Brett, Raina

Translating Customer Service Chats

Joshua Frank, Forest Maguire, Rose Resnick, Christopher White, Raymond Zong

Motivation

- There are ~1,170 languages in use today (Ethnologue)
- Customer support representatives can only speak so many languages
- There is a need for high quality machine translation between languages

Due to our projects scope we chose to focus on one language
We chose French because it was the language we were most familiar with that was represented in the data we were provided

The Task

- Our task was to take an input in the English language and output an accurate French translation or vice-versa
- An input might be: “Please follow the instructions below.”
- And the output would be: “Veuillez suivre les instructions ci-dessous.”

Data

- Our data was provided by the WMT 2024 Chat Shared Task
- Our training data set was 9,497 translation pairs
- Our validation data set was 2,420 translation pairs
- For preprocessing we de-duplicated our data, lowercased it, removed punctuation marks and removed emojis

Methods

- Failed Seq2Seq Model
- Originally, we tried to train a sequence-to-sequence model from scratch
- Though we tried several approaches, this ended up failing
- MarianMT
- Our baseline model
- This model was specifically created to translate from English to French
- We also created a version of it that was finetuned on our data
- LLaMA & Gemma
- Decoder-only models
- We used the following prompt (translated into French if our prompt was in French)
- "You are a professional French-to-English translator. Translate the user's French text into natural English. Return only the translation — no explanations or extra text."

Results

- We evaluated our models using chrF and BERTScore

Baseline MarianMT Results:

chrf: 65.2

BERTScore F1: 0.8649, BERTScore Precision: 0.8651, BERTScore Recall: 0.8653

Finetuned MarianMT Results:

chrF: 76.1551

BERTScore F1: 0.9144, BERTScore Precision: 0.9153, BERTScore Recall: 0.9140

LLaMA Results:

chrF: 70.21,

BERTScore F1: 0.6893, BERTScore Precision: 0.6794, BERTScore Recall: 0.6998

Gemma Results:

chrF: 63.58,

BERTScore F1: 0.6671, BERTScore Precision: 0.685, BERTScore Recall: 0.6496

Finetuned MarianMT was the most accurate!

Thank you!

3. Praz, Wyatt, Justin, Pier, Ryan

Training LLMs from Scratch

Pretraining to Fine-Tuning: Building Code Generation Models

Ryan Bloch | Pierandrea Ferrara | Justin Liang | Wyatt McMullen | Praz Nagarajan

Project Overview

Objective

Build code generation LLMs end-to-end: from random weight initialization through pretraining to instruction fine-tuning.

Two models developed:

- GPT-2 Medium (355M params)
- Custom Transformer (253M params)

Data & Pipeline

Pretraining:

Stack De-duplicated (Python subset, ~30M files)

Fine-tuning:

Alpaca-20k + CodeInstructions-122k

Evaluation:

50 LeetCode problems scored by GPT-4.1 mini (0–5 scale)

Model Architecture & Training

GPT-2 Medium

355M Parameters

- 24 layers, 1024 hidden dim, 16 heads
- 1024-token context window
- 4× NVIDIA A100 GPUs
- 97K steps, ~22 hrs training
- Val loss: 1.007 (perplexity 2.74)
- Two rounds of fine-tuning

Scratch Transformer

253M Parameters

- 16 layers, 1024 hidden dim, 16 heads
- 512-token context window
- Single GPU training
- 63K steps to convergence
- Val loss: 1.617
- Built entirely in PyTorch

Pretraining

Training both models from random weight initialization on raw Python code

01 Data

500K Python files from Stack De-duplicated. Metadata stripped — only raw source code tokenized using GPT-2 BPE tokenizer (50,257 vocab).

02 Training Setup

Mixed precision (bf16), AdamW optimizer, gradient clipping at 1.0, cosine LR decay with linear warmup (2K steps). Effective batch: 128 (GPT) / 32 (Scratch).

03 Outcome

GPT-2: val loss 1.007 (perplexity 2.74) after 97K steps. Scratch: val loss 1.617 after 63K steps. Both learned valid Python syntax generation.

Instruction Fine-Tuning

Supervised fine-tuning to transform base models into instruction-following code generators

Round 1

Both Models

- Dataset: Alpaca-20k (instruction/input/response)
- SFTTrainer with bf16 precision
- LR: 2e-5 (10× lower than pretraining)
- Cosine schedule, 200 warmup steps
- Early stopping: patience 3, threshold 0.01
- Stopped at 2 epochs — eval loss: 1.069

Round 2

GPT-2 Only

- Expanded to ~150K examples (+CodeInstructions-122k)
- LR dropped to 1e-5 (already tuned once)
- Looser stopping: patience 5, threshold 0.005
- Eval loss improved: 1.069 → 1.014
- Qualitative leap: correct Fibonacci output vs. broken attempt from Round 1

Evaluation Method

Custom LLM-as-judge benchmark using GPT-4.1 mini

Why Custom Evaluation?

- Standard benchmarks (LiveBench) would yield near-zero scores for small models
- BERTscore doesn't capture syntax correctness or problem-solving ability
- LLM-as-judge captures multi-faceted code quality better than token-matching metrics

Scoring Rubric (0–5)

0 — Completely unrelated or empty output

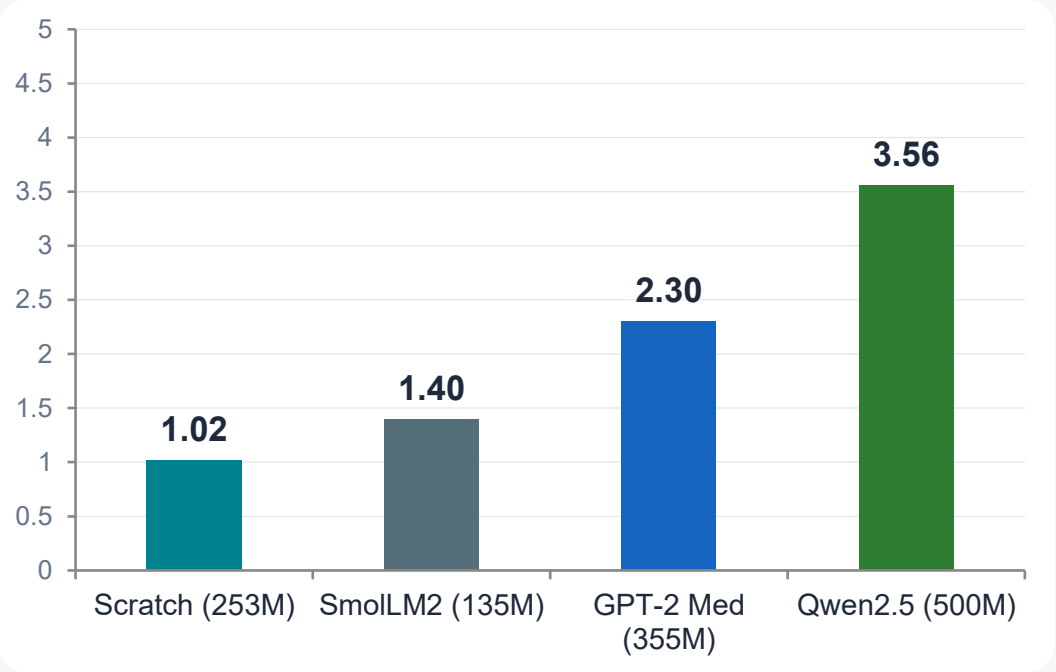
3 — Core idea present but incomplete / buggy

5 — Fully correct, runs, solves the problem

50 easy LeetCode problems across various topics, 4 models compared side-by-side

Evaluation Results

Average LLM evaluation score across 50 LeetCode problems (0–5 scale, graded by GPT-4.1 mini)



Key Findings

- GPT-2 Med outperformed SmolLM2 despite similar approaches
- Scratch model suffered from under-generation (early stop tokens)
- GPT-2 showed over-generation past valid responses
- Both models produced valid Python syntax

Takeaways & Future Work

Key Takeaways

- Fine-tuning quality matters as much as model size
- Transformer architecture complexity significantly impacts output quality
- Smaller models can approach larger-model quality with targeted fine-tuning

Future Work

- Second fine-tuning round for Scratch model
- Prompt engineering to improve crude model outputs
- Address over/under-generation patterns in both models

4. Amyia, Sarina, Kiana, Michelle, Grace

Who's Who in *Modern Family*:
Character Attribution in a TV Sitcom

CS 1671 Project Proposal Presentation

Michelle Hong, Amyia Singh, Grace Hines, Sarina Saran, Kiana Kazemi

Project Motivation

- Distinguish between character set of diverse personalities
- Test ability of model to identify characters based on script alone
- Provide insight on spoken language representation in American media

Task

Our task was to train a model that learns patterns in dialogue and predicts which character is speaking given a line from the sitcom Modern Family.

Example Input

A line of dialogue such as:

i collect antique fountain pens , i 'm adept at japanese flower arrangement ... ikebana . and i was a starting offensive lineman at the university of illinois . surprise !

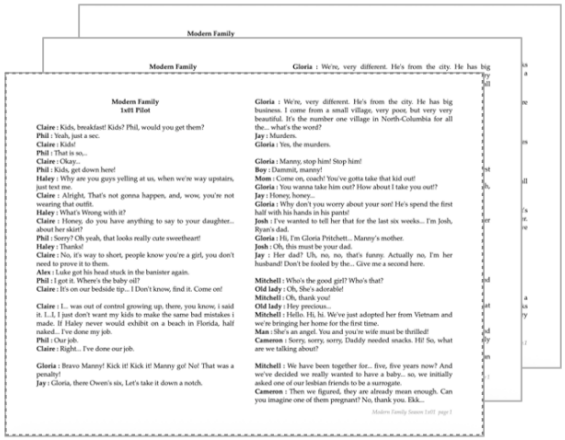
Example Output

A prediction of the character:

Cameron

Datasets

SCRIBD Script PDFs



Modern Family

Modern Family

Modern Family
1x01 Pilot

Claire : Kids, breakfast! Kids? Phil, would you get them?
Phil : Yeah, just a sec.
Claire : Kids!
Phil : That is us...
Claire : Okay...
Phil : Kids, get down here!
Haley : Why are you guys yelling at us, when we're way upstairs, just text me.
Claire : Alright. That's not gonna happen, and, wow, you're not wearing that outfit.
Haley : What's Wrong with it?
Claire : Honey, do you have anything to say to your daughter... about her skirt?
Phil : Sorry? Oh yeah, that looks really cute sweetheart!
Haley : Thanks!
Claire : No, it's way to short, people know you're a girl, you don't need to prove it to them.
Alex : Luke got his head stuck in the banister again.
Phil : I got it. Where's the baby oil?
Claire : It's on our bedside tip... I Don't know, find it. Come on!

Claire : I... was out of control growing up, there, you know, i said it. I, I, I just don't want my kids to make the same bad mistakes i made. If Haley never would exhibit on a beach in Florida, half naked... I've done my job.
Phil : Our job.
Claire : Right... I've done our job.

Gloria : Bravo Manny! Kick it! Kick it! Manny go! No! That was a penalty!
Jay : Gloria, there Owen's six, Let's take it down a notch.

Gloria : We're, very different. He's from the city. He has big business. I come from a small village, very poor, but very very beautiful. It's the number one village in North-Columbia for all the... what's the word?
Jay : Murders.
Gloria : Yes, the murders.

Gloria : Manny, stop him! Stop him!
Boy : Dammit, manny!
Mom : Come on, coach! You've gotta take that kid out!
Gloria : You wanna take him out? How about I take you out!?
Jay : Honey, honey...
Gloria : Why don't you worry about your son! He's spend the first half with his hands in his pants!
Josh : I've wanted to tell her that for the last six weeks... I'm Josh, Ryan's dad.
Gloria : Hi, I'm Gloria Pritchett... Manny's mother.
Josh : Oh, this must be your dad.
Jay : Her dad? Uh, no, no, that's funny. Actually no. I'm her husband! Don't be fooled by the... Give me a second here.

Mitchell : Who's the good girl? Who's that?
Old lady : Oh, She's adorable!
Mitchell : Oh, thank you!
Old lady : Hey precious...
Mitchell : Hello. Hi, hi. We've just adopted her from Vietnam and we're bringing her home for the first time.
Man : She's an angel. You and you're wife must be thrilled!
Cameron : Sorry, sorry, sorry. Daddy needed snacks. Hi! So, what are we talking about?

Mitchell : We have been together for... five, five years now? And we've decided we really wanted to have a baby... so, we initially asked one of our lesbian friends to be a surrogate.
Cameron : Then we figured, they are already mean enough. Can you imagine one of them pregnant? No, thank you. Ekk...

Modern Family Season 1x01 page 1

Seasons 1-3
10 characters → 6 characters
Removed the children (Haley, Alex, Luke, Manny)

CSV Dataset

Character	Line
Claire	Kids? Phil, would you get them?
Phil	Yeah, just a sec.

16,332 lines
241,214 preprocessed tokens
9,586 vocabulary size

Methods | Baseline & LLM

Baseline Model

Bag-of-Words Unigram

- Simplest possible text representation
- True baseline model

TF-IDF Unigram + Bigram

- Down-weighted common words
- Character-specific words have more weight

Large Language Model

- Input: line from show
- Output: character that spoke the line
- Compared output with gold label to determine accuracy

Zero-Shot Prompting

- Fed LLM a line from the show

In-Context Learning

- Fed LLM several examples of lines from the show with their associated characters

Results

Model	TF-IDF Unigram+Bigram	BoW Unigram	Zero-Shot Prompting	In-Context Learning
Accuracy	0.357	0.338	0.288	0.284

Our baseline models performed better than the large language models…

These findings suggest that language alone carries some information about character identity, but not enough for highly accurate classification. The traditional TF-IDF model outperformed the more advanced LLM, showing that task-specific lexical features can be more effective than general-purpose language knowledge when training data is limited and utterances are short.

5. Nihal, Kevin, Owen, Nate, Enzo, Matthew

6. Irisin, Yifei, Sanjana, Ciara, Astalaxmi

UNIVERSITY OF PITTSBURGH · NLP GROUP PROJECT

Predicting S&P 500 Intraday Direction from Financial News

FinBERT-Based Supervised Systems vs. Gemma3 Zero-Shot Reasoning

Irisin Yu · Astalaxmi Dhanaseelan · Ciara Dwyer · Yifei Tian · Sanjana Pejathaya

73.8%

Gemma3 Zero-Shot Accuracy

3,507 Trading Days · 2008–2024 · Binary Classification

Research Overview

01

Problem & Task

Binary classification: will S&P 500 close ≥ open?

02

Data

3,507 trading days, 2008–2024, news headlines + price data

03

Methods

6 systems: FinBERT + classifiers vs. Gemma3 zero-shot LLM

04

Results

73.8% vs 65.0% accuracy — LLM wins by 8.8 points

05

Insights & Future

Why LLMs dominate; next steps including few-shot & ensembles

Problem Statement & Task Definition

Core Research Question

Can same-day financial news headlines predict whether the S&P 500 closes above its opening price?

Task Definition

Binary Classification:
Predict whether S&P 500 close ≥ open (label 1) or close < open (label 0).

Realistic Intraday Constraint:
All features — headlines, opening price, and prior-day technical indicators — are observable before market close.

Evaluation Period:
2008–2024 (covers financial crisis, COVID crash, bull markets)

WORKED EXAMPLE

Input Headlines:

"Fed raises interest rates by 75 basis points amid persistent inflation concerns"

"Tech stocks fall sharply as bond yields surge"

"Consumer confidence index drops to 18-month low"

Correct Output: Label 0 — Market Closed LOWER

FinBERT: assigns high negative probability · Gemma3: reasons from economic context

Dataset & Data Processing

3,507	2008–2024	14,297	5.6 avg
Total Trading Days	Coverage Period	Individual Headlines	Headlines Per Day

Preprocessing Steps

1. **Binary Label:**
   y = 1 if close ≥ open, else y = 0
2. **Near-Flat Filter:**
   Remove days where |close − open| / open < 0.2%, reducing to 2,572 days
3. **Class Distribution:**
   After filter: 55.4% up days, 44.6% down days
4. **Train/Test Split:**
   Chronological 80/20 split to prevent data leakage

Feature Set Summary

Feature	Description
sent_pos / neg / neu	FinBERT softmax probabilities (3-dim)
gap_open	(today open − yesterday close) / yesterday close
prev_ma5_ratio	Yesterday close / 5-day moving average
prev_rsi14	14-day RSI (shifted 1 day)
prev_vol_ratio	Yesterday volume / 5-day avg volume
sent_pos/neg_mean/max	Per-day aggregated FinBERT statistics
sent_variance	Variance of positive prob across headlines
n_headlines	Number of headlines that day

Methodology: Two Paradigms

PARADIGM 1

Traditional NLP + Machine Learning

Step 1: FinBERT

Extract 3-dim sentiment probabilities from financial headlines

Step 2: Feature Engineering

Add 4 price-based indicators: gap_open, MA5 ratio, RSI14, vol_ratio

Step 3: Classifiers

LinearSVC, XGBoost, Feature Tokenizer Transformer (tabular deep learning)

Step 4: Multi-Headline Ext.

Per-headline FinBERT scoring → aggregate 7 stats (mean, max, variance)

VS

PARADIGM 2

LLM Zero-Shot Reasoning

Step 1: Gemma3-4B-IT

General-purpose instruction-tuned LLM (no task-specific training)

Step 2: Structured Prompt

Headlines → structured prompt → single-digit answer (0 or 1)

Step 3: Zero-Shot

No fine-tuning, no in-context examples, greedy decoding (temp=0)

Step 4: Text-Only Input

No price data — model reasons purely from news headline semantics

The Six Experimental Systems

S1 64.1%

FinBERT → LinearSVC

3-dim sentiment only. Baseline system.

S2a 64.2%

+ Price → LinearSVC

7-dim: sentiment + 4 price features. Best linear.

S2b 62.4%

+ Price → XGBoost

Non-linear boosted trees. n=300, max_depth=3.

S3 62.2%

+ Price → FT-Transformer

Tabular deep learning, 18K params, 2 attn heads.

S4 65.0%

Multi-Sent → FT-Transformer

11-dim: 7 per-headline stats + 4 price. Best supervised model. ★

S5 53.9%

Multi-Sent → LinearSVC

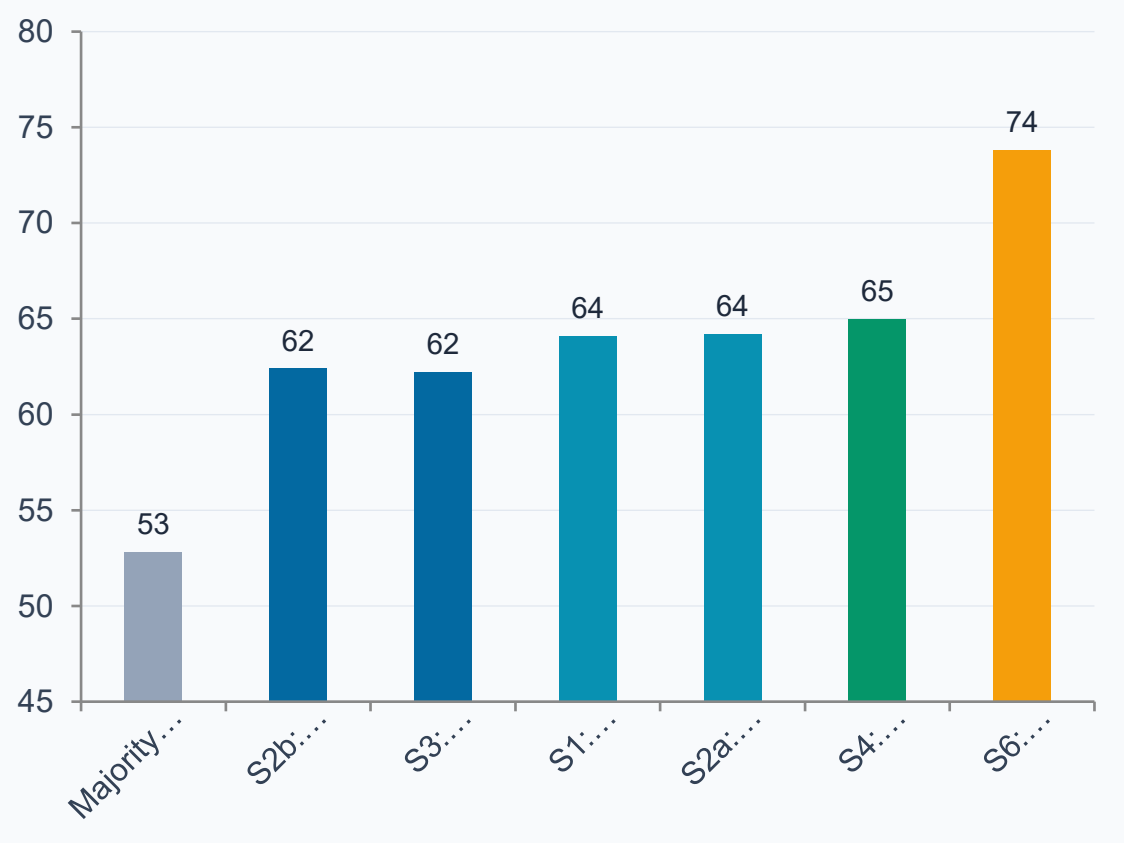
11-dim features on linear boundary. Class collapse — predicts UP 99.6% of the time. ⚠

S6 73.8%

Gemma3-4B Zero-Shot

Raw headlines only. No fine-tuning. University API. No price data. 🏆

Results: Accuracy Comparison



Key Takeaways

+20 pts above baseline

Gemma3 surpasses the majority-class baseline by over 20 percentage points

+8.8 pts over best supervised

Gemma3 (73.8%) beats the best traditional model (65.0%) by a substantial margin

Multi-headline matters

Per-headline aggregation (S4) improves over single-pass FinBERT (S3) by +2.8 pp

Linear beats deep

With ~2K training samples, LinearSVC outperforms the FT-Transformer

S5 class collapse

Multi-headline LinearSVC predicts 'up' 99.6% of the time — a cautionary failure mode

Gemma3-4B: Prompt Design (Zero-Shot)

Prompt Template (used verbatim for every test day)

```
You are a financial market analyst.
Below are S&P 500 news headlines
from one trading day.

Predict whether the S&P 500 will
close HIGHER or LOWER than its
opening price that same day.

Reply with a single digit only —
no spaces, no punctuation:
  0  =  market closes UP (close ≥ open)
  1  =  market closes DOWN (close < open)

Headlines: {headlines}

Answer:
```

Prompt Design Choices

Role framing

"Financial market analyst" primes domain expertise

Single-digit output

Eliminates parsing ambiguity; max 5 new tokens

Greedy decoding

Temperature = 0 for reproducibility

No price data

Model reasons from text semantics only

0 = UP, 1 = DOWN

Explicit label mapping to avoid confusion

100% parseable

All 515 test responses were valid (0 skipped)

Gemma3 Detailed Performance Analysis

Confusion Matrix — Gemma3-4B (515 test days)

ACTUAL ↓ \ PREDICTED →	UP (0)	DOWN (1)
UP (0)	175 True Negative	101 False Positive
DOWN (1)	34 False Negative	205 True Positive

⚠ Bearish Bias: FN (34) << FP (101) — Gemma3 is more cautious about predicting 'up'. This conservatism benefits accuracy given balanced classes.

Per-Class Metrics — Key Systems

System / Class	Precision	Recall	F1
S4: Multi-FT — Down	0.61	0.67	0.64
S4: Multi-FT — Up	0.69	0.64	0.66
S5: LinearSVC — Down	0.75	0.01	0.02
S5: LinearSVC — Up ⚠	0.54	1.00	0.70
S6: Gemma3 — Up	**0.84**	**0.63**	**0.72**
S6: Gemma3 — Down ★	**0.67**	**0.86**	**0.75**

Feature Importance (S2a LinearSVC coefficients):

gap_open (+0.238) → sent_pos (+0.096) → prev_rsi14 (+0.14) → sent_neg (−0.068)

Error Analysis

False Positives (FP = 101)

Predicted DOWN — Actual: UP

Common pattern: Mixed or mildly negative headlines that don't overcome underlying market momentum.

Example: Days with geopolitical tensions but no major domestic economic data — Gemma3 predicts conservative negative even when market absorbs the news and closes higher.

False Negatives (FN = 34)

Predicted UP — Actual: DOWN

Common pattern: Technically positive headline sentiment (e.g., strong earnings) on days where broader macro conditions caused a sell-off.

Key insight: A nuance that even Gemma3 cannot fully capture without access to real-time price context.

System 5: Class Collapse Case Study

What happened:

Multi-headline LinearSVC predicted 'up' for 508 of 512 test days (99.6%). Only TN = 3 and FN = 1.

ROC-AUC = 0.648 — reasonable score, but accuracy = 53.9% (near majority baseline).

Why it happened:

→ 11-dim multi-headline features don't provide linearly separable signal in the filtered dataset

→ class_weight='balanced' did not resolve the collapse

→ FT-Transformer (S4) overcomes this via attention-based feature interaction

→ Lesson: richer features ≠ better classifier when the boundary is non-linear

Discussion: Why Does Gemma3 Dominate?

01 Information Richness

FinBERT compresses each headline into a 3-dimensional sentiment vector, discarding named entities, event types, causal chains, and financial domain knowledge. Gemma3 processes the full text.

02 Semantic Nuance

"Fed raises rates amid inflation" (bearish) vs. "Fed signals pause in rate hikes" (bullish) — both contain the word "rate" but have opposite market implications. Gemma3 distinguishes them; FinBERT collapses them.

03 Pretrained World Knowledge

Gemma3's pretraining presumably includes financial commentary, historical event-market relationships, and economic theory — prior knowledge that 2,045 training examples cannot impart to a smaller supervised model.

04 Why Transformer < LinearSVC?

With only ~2,000 training samples, the FT-Transformer (18K parameters) cannot benefit from its capacity advantage. The Transformer's self-attention requires sufficient data diversity to learn meaningful cross-feature interactions.

Future Work & Extensions

Few-Shot Prompting

Add 3–6 in-context examples (both up and down days) to anchor Gemma3's output format and calibrate priors. Expected further improvement from Brown et al. (2020).

Next-Day Prediction

Shift target to predicting next-day open vs. today's close — a more practically useful trading signal. LLaMA-based next-day system was explored; extend Gemma3 for direct comparison.

Larger LLMs

Evaluate Gemma3-12B or equivalent models for zero-shot accuracy. Conduct systematic scaling study to determine whether performance gains justify inference cost overhead.

Ensemble Approaches

Late-fusion ensemble combining Gemma3's soft probability outputs with FinBERT-based supervised features — capturing both qualitative LLM reasoning and quantitative price signals.

Error-Driven Refinement

The 101 false positives share a pattern: mildly negative macro headlines on positive momentum days. Targeted data collection — headlines from momentum-reversal days — could address this cluster.

Cross-Market Generalization

Current system is S&P 500 index only. Extend to individual equities, sector ETFs, and international markets to test whether the LLM advantage generalizes across asset types.

Limitations & Ethical Considerations

⚠ Limitations

S&P 500 only

Does not generalize directly to individual equities or other markets

Single train/test split

Results may vary across time periods, especially extreme events (2008 crisis, COVID)

Data contamination risk

Gemma3-4B training cutoff may overlap with test period — potential leakage cannot be fully ruled out

No transaction costs

Accuracy ≠ trading profitability; slippage, market impact not modeled

Headline bias

Collection methodology not fully documented; possible survivorship bias in news sources

⚖ Ethical Considerations

Market Volatility Risk

Deployed at scale, automated prediction systems could amplify volatility by creating feedback loops between news sentiment and algorithmic trading.

Financial Loss Potential

Errors in prediction could, in a deployed context, lead to real financial losses for users who trust the system's outputs.

Academic Use Only

This project is purely academic. The systems described are NOT intended for deployment or live trading.

Media Bias in Data

The financial news corpus may reflect existing biases — over-representing certain sectors, geographies, or market conditions.

CONCLUSION

Key Findings

1. **Gemma3 dominates:**
   73.8% accuracy — 8.8 pp above the best supervised model, 20+ pp above baseline

2. **LLMs offer qualitative reasoning:**
   Large-scale pretraining enables nuanced financial understanding that fine-tuning on limited data cannot match

3. **Multi-headline aggregation helps:**
   Per-headline FinBERT scoring + aggregation adds +2.8 pp over single-pass approach

4. **gap_open is the king feature:**
   Overnight gap encodes near-term sentiment visible to market participants more powerfully than any other feature

5. **Data size constrains deep learning:**
   With ~2K samples, linear methods outperform FT-Transformers; attention needs diversity to shine

73.8%

Gemma3
Zero-Shot

vs.

65.0%

Best Supervised
(S4 Multi-FT)

github.com/Irisiiin/Stock-Prediction-Project

7. Marcus, Aaron, Patrick, Hongyu, Jack

Annotating Online Gaming Voice Chat

Motivation

- Toxic speech is unfortunately common in online gaming spaces
- Many videogames have some form of toxic speech monitoring
  - This often comes in the form of player reporting followed by some form of manual review
- Constructing a model that could automatically detect and categorize toxic speech would streamline this process
- To train a model, we need a large dataset of true instances of toxicity
- Our goal is to create a structured system to create a dataset of true instances of toxicity

Task Description

- Create a codebook for standardized application of labels over transcripts
- This involves...
  - Generating a set of labels covering common types of toxicity
  - Each annotator reads through a set of transcripts and applies labels as they see fit
  - The group meets to discuss label application criteria
  - Label application criteria is used to create codebook

Data

- 480 software-generated transcript files
- Transcripts generated from livestreams on the streaming platform, Kick
- Streams selected featured gameplay/commentary over the 13 most popular games in June 2024
- Transcripts capture up to 5 hours of a stream

Methods - Annotations

Doccano: Setup & Limitations

- Initial unfamiliarity with setup and functionality
- Required complex configuration:
  - Project setup, user accounts, permissions
  - Data loading
  - Remote access via Docker + URL tunneling
- No file naming or sorting system after upload
  - Files shuffled → difficult coordination
  - Could not reliably reference specific documents
- Annotation constraints:
  - Only one label per text span initially
  - Prevented overlapping/disagreement analysis
  - Later fixed, but slowed early progress
  - **Transition to Excel**
- Doccano's highlighting/labeling features proved to be better for span annotation

Transition to Excel

- Doccano's highlighting/labeling features proved limited
- Excel chosen as alternative despite not being designed for annotation
- Excel Advantages
  - Easy data import and structured formatting
  - Text split into rows → cleaner annotation workflow
  - Labels applied via binary columns (1 = present, 0 = absent)
  - More flexible and scalable for team coordination

Methods - Codebook

- Formally define the goals of the project/task
- What is offensive content?
- What are the labels?
  - What do we need to know about a label?
    - Content it covers
    - Examples of it in the transcripts
- Refine and iterate after annotation sessions

Methods – Inter-Annotator Agreement

- Multi-rater setting: >2 annotators → Cohen's Kappa not applicable
- Why chance-corrected measures? Raw agreement can be inflated by chance, especially with imbalanced labels, so we use metrics that adjust for expected agreement
- Metrics considered:
  - Fleiss' Kappa
  - Gwet's AC1
  - Krippendorff's Alpha
- Key consideration: Class imbalance → "kappa paradox" (high observed agreement, low kappa)
- Planned approach:
  - Gwet's AC1 (robust to prevalence imbalance)
  - Krippendorff's Alpha (handles missing data, flexible across settings)
- Final approach:
  - Used Fleiss' Kappa due to computational constraints (dataset size, runtime)
  - All metrics implemented in R (`irr` package)

Results - Codebook

- 10-page guide
- Covers
  - Forms of Hate Speech
    - Targeting Sex, Gender, Race, Ethnicity, Religion, etc.
  - Harassment
  - Sexual Vulgarity
  - Microphone Spamming
  - Fatphobia
  - Implicit Toxicity

2.3. Harassment

This is content intended to offend another player in some way. It may be to disrupt their ability to enjoy the game, to make them feel unsafe, or to influence them in some other manner (e.g., quitting the game). While similar to hate speech in some respects, it does not target a player based on their identity (i.e., the characteristics described above).

2.3.1. Examples

The following examples are taken directly from the transcripts:

a. **Line:** "Shut up, drug addict"
   Reasoning: This message is intended to offend another player by calling them a "drug addict".
b. **Line:** "Nice noise, nice noise pollution. You can't afford good property. You can't afford good land, bro. That's sad. That's so sad. Bro, why would I come there? I probably can't hear myself. Think it's so loud. You see you live on a highway. You're poor."
   Reasoning: This message attacks another player as being poor. It seems that the other player may have a lot of background noise ("Nice noise, nice noise pollution") that the speaker conflates with the player being poor. Note that this is all the same line: the speaker bombards another player with offensive messages to disrupt them.

Line: "Nice noise, nice noise pollution. You can't afford good property. You can't afford good land, bro. That's sad. That's so sad. Bro, why would I come there? I probably can't hear myself. Think it's so loud. You see you live on a highway. You're poor."

Line: "You got a lot of extra wind in your voice, so it sounds like it." After previously saying "You sound fat too, so I imagine you're telling the truth."

Results – Inter-Annotator Agreement

- Observed agreement: Fleiss' Kappa κ = 0.443
  - Indicates **moderate agreement** per Landis and Koch framework
  - Reflects **reasonable but imperfect consistency**
- Context:
  - Typical for **complex / subjective annotation tasks**
  - Likely impacted by **class imbalance**
- Key limitation:
  - Reliance on Fleiss' Kappa alone
  - Kappa may **underestimate agreement** under skewed label distributions
- Future directions:
  - Optimize computation (efficient pipelines / subsampling)
  - Evaluate multiple agreement metrics
- Takeaway:
  - κ = 0.443 provides a **useful baseline** for annotation reliability

κ	Interpretation
< 0	Poor agreement
$0.01 - 0.20$	Slight agreement
$0.21 - 0.40$	Fair agreement
$0.41 - 0.60$	Moderate agreement
$0.61 - 0.80$	Substantial agreement
$0.81 - 1.00$	Almost perfect agreement

8. David, Jeana, Griffin, Cole, Daley

Project Motivation

- AI usage increasing
- May harm human creators
- AI development outpaces AI regulation

Task Description

- Goal: predict whether a given piece of text was written by a human or generated by AI
- Binary classification — output 0 (human) or 1 (AI)
- Harder than it sounds — even humans struggle to tell the difference

Input	Gold	Predicted
"My friend bought tickets to this Friday's big game..."	AI (1)	Human (0) ❌
Wikipedia article about Call of Duty	Human (0)	AI (1) ❌

Data

- Large publicly available dataset from HuggingFace
  - Mix of AI and human-generated English Text
  - Labeled 0 for human and 1 for AI
  - Raw dataset contained 872,525 datapoints
  - After preprocessing the final dataset included 848,986
- Preprocessing
  - Removed 23,518 rows with identical text strings
  - Removed 21 rows with fewer than 3 words in the text string

Methods

- Baseline: Logistic Regression with three feature sets
  - Unigram — bag-of-words, 5,000 features
  - TF-IDF — unigrams with sublinear term weighting
  - Unigram + Bigram — adds word-pair features
- LLM: Gemma 3:27B (chosen for speed — 13.97s / 10 queries)
  - Zero-shot: prompt only, no examples
  - Few-shot: 2 human + 2 AI examples per query (<100 words each)

Results

Model	Accuracy	Precision	Recall	F1
Unigram LR	**0.8529**	**0.8703**	**0.9034**	**0.8866**
Unigram + Bigram LR	0.8024	0.8228	0.8786	0.8498
TF-IDF LR	0.7987	0.8237	0.8697	0.8461
LLM Few-shot (Gemma 3:27B)	0.6482	0.7240	0.7223	0.7232
LLM Zero-shot (Gemma 3:27B)	0.6332	0.7465	0.6411	0.6898

Test set: 84,899 samples | **Bold** = best model

Key Takeaways:

- Logistic regression significantly outperformed both LLM approaches, with the unigram model achieving the highest F1 of 0.8866
- The LLM struggled despite being a 27 billion parameter model; without task-specific training data, it had to rely on the prompt alone
- Adding few-shot examples helped the LLM improve from 0.6898 to 0.7232 F1, but made it more likely to flag human text as AI
- All models had the most trouble with formal human writing and casual AI writing, where the two classes look most similar on the surface

Discussion

- Unigram LR (F1: 0.89) outperformed few-shot LLM (F1: 0.72) by ~17 points
  - LR trained on 679K in-distribution examples; LLM had only a prompt
- Adding bigrams or TF-IDF did not help, simple word frequency is the strongest signal
- All models struggled at the same boundary: formal human text ↔ casual AI text
  - False positives: encyclopedic/academic human writing flagged as AI
  - False negatives: conversational AI text mistaken for human
- Few-shot improved zero-shot (0.69 → 0.72) but made the LLM more aggressive -> more false positives

9. Ryder, Ifemi, Vivien, Heather, Fatimah

Classifying Restaurants by Yelp Reviews

Multi-task NLP Classification using Traditional Models and LLMs

Fatimah Alfaraj, Heather Guzik, Vivien Lim,
Ifemi Olojo-Kosoko, Ryder Pham

Project Motivation & Task

Motivation

- Yelp has millions of user-written reviews
- Difficult to organize and search manually
- Goal: Automatically classify reviews

Task

- Predict cuisine type
- Predict restaurant formality

Example

Input: “"Best local pizza place around! The pizza is delicious and fresh. We've tried the Hot, Mild, Garlic Parmesan, and BBQ wings and all of them are fantastic. This is a carry-out place, so don't expect to bring the family in, but there are a couple of small tables if you're just running in to get a slice.”

Cuisine Output: American

Formality Output: Casual

Data

Dataset Source:

1. Yelp Review Data Set (has over 6 million reviews)
2. Focused on Restaurants only

Target Labels:

- Cuisine Types (derived from categories)
- Formality level (derived from business attributes):
  - Casual
  - Mid-range
  - Fine dining

Data Overview

Review dataset

Columns in dataset: ['review_id', 'user_id', 'business_id', 'stars', 'useful', 'funny', 'cool', 'text', 'date']
Loaded 25 rows

First 25 rows of the dataset:

	review_id	user_id	business_id	stars	useful	funny	cool	text	date
0	KU_O5udG6zpxOg-VcAEodg	mh_-eMZ6K5RLWhZyISBhwA	XQfwVwDr-vOZS3_CbbE5Xw	3	0	0	0	If you decide to eat here, just be aware it is...	2018-07-07 22:09:11
1	BiTunyQ73aT9WBnpR9DZGw	OyoGAe7OKpv6SyGZT5g77Q	7ATYjTIgM3jUlt4UM3IypQ	5	1	0	1	I've taken a lot of spin classes over the year...	2012-01-03 15:28:18
2	saUsX_uImxRlCVr67Z4Jig	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	3	0	0	0	Family diner. Had the buffet. Eclectic assortm...	2014-02-05 20:30:30
3	AqPFMleE6RsU23_auESxiA	_7bHUi9Uuf5__HHc_Q8guQ	kxX2SOes4o-D3ZQBkiMRfA	5	1	0	1	Wow! Yummy, different, delicious. Our favo...	2015-01-04 00:01:03
4	Sx8TMOWLNuJBWer-0pcmoA	bcjbaE6dDog4jkNY91ncLQ	e4Vwtrqf-wpJfwesgvdgxQ	4	1	0	1	Cute interior and owner (?) gave us tour of up...	2017-01-14 20:54:15
5	JrtxlS1TzJ-iCu79ul40cQ	eUta8W_HdHMXPzLBBZhL1A	04UD14gamNjLY0IDYVhHJg	1	1	2	1	I am a long term frequent customer of this est...	2015-09-23 23:10:31
6	6AxgBCNX_PNTOxmbRSwcKQ	r3zeYsv1XFBRA4dJpL78cw	gmjsEdUsKpj9Kxu6pdjH9g	5	0	2	0	Loved this tour! I grabbed a groupon and the p...	2015-01-03 23:21:18
7	_ZeMknuYdlQcUqng_Im3yg	yfFzsLmaWF2d4Sr0UNbBgg	LHSTtnW3YHCeUkRDGyJOyw	5	2	0	0	Amazingly amazing wings and homemade bleu chee...	2015-08-07 02:29:16
8	ZKvDG2sBvHVdF5oBNUOpAQ	wSTuiTk-sKNdcFyprzZAjg	B5XSoSG3SfvQGtKEQQT1SQ	3	1	1	0	This easter instead of going to Lopez Lake we ...	2016-03-30 22:46:33
9	pUycOfUwM8vqX7KjRRhUEA	59MxRhNVhU9MYndMkz0wtw	gebiRewfieSdtt17PTW6Zg	3	0	0	0	Had a party of 6 here for hibachi. Our waitres...	2016-07-25 07:31:06
10	rGQRf8UafX7OTlMNN19I8A	1WHRWwQmZOZDAhp2Qyny4g	uMvVYRgGNXF5boolA9HXTw	5	2	0	0	My experience with Shalimar was nothing but wo...	2015-06-21 14:48:06
11	I3Wk_mvAog6XANluGQ9C7Q	ZbqSHbgCjzVAqaa7NKWn5A	EQ-TZ2eeD_E0BHuvoaeG5Q	4	0	0	0	Locals recommended Milktooth, and it's an amaz...	2015-08-19 14:31:45
12	XW_LfMv0fV2I19c6xQd_Jw	9OAtfnWag-ajVxRbUTGlyg	lj-E32x9_FA7GmUrBGBEWg	4	0	0	0	Love going here for happy hour or dinner! Gre...	2014-06-27 22:44:01
13	8JFGBuHMoiNDyfcxuWNtrA	smOvOajNG0IS4Pq7d8g4JQ	RZtGWDLCAtuipwaZ-UfjmQ	4	0	0	0	Good food--loved the gnocchi with marinara|nth...	2009-10-14 19:57:14
14	UBp0zWyH60Hmw6Fsasei7w	4Uh27DgGzsp6PqrH913giQ	otQS34_MymijPTdNBoBdCw	4	0	2	0	The bun makes the Sonoran Dog. It's like a snu...	2011-10-27 17:12:05
15	OAhBYw8IQ6wlfw1owXWRWw	1C2lxzUo1Hyye4RFIXly3g	BVndHaLihEYbr76ZDCMEGw	5	0	0	0	Great place for breakfast! I had the waffle, w...	2014-10-11 16:22:06
16	oyaMhzBSwfGgemSGuZCdwQ	Dd1jQ7S-BFGqRbApFzCFw	YtSqYvIQ_pOItsVPSx54SA	5	0	0	0	Tremendous service (Big shout out to Douglas) ...	2013-06-24 11:21:25
17	LnGZB0fjfgeVDVz5IHuEVA	j2wlzrntrbKwyOcOiB3I3w	rBdG_23USc7DletfZ1txGA	4	1	0	0	The hubby and I have been here on multiple occ...	2014-08-10 19:41:43
18	u2vzZaOqJ2feRshaaF1doQ	NDZvyYHTUWWu-kqgQzzDGQ	CLEWowfkj-wKYJIQDqT1aw	5	2	0	1	I go to blow bar to get my brows done by natal...	2016-03-07 00:02:18
19	Xs8Z8ImKkosqW5mw_sVAoA	lQsF3Rc6IgCzjVV9DE8KXg	eFvzHawVJofxSnD7TgbZtg	5	0	0	0	My absolute favorite cafe in the city. Their b...	2014-11-12 15:30:27

Business dataset

Columns in dataset: ['business_id', 'name', 'address', 'city', 'state', 'postal_code', 'latitude', 'longitude', 'stars', 'review_count', 'is_open', 'attributes', 'categories', 'hours']
Loaded 25 rows

First 25 rows of the dataset:

	business_id	name	address	city	state	postal_code	latitude	longitude	stars	review_count	is_open	attributes	categories	hours
0	Pns2l4eNsfO8kk83dixA6A	Abby Rappoport, LAC, CMQ	1616 Chapala St, Ste 2	Santa Barbara	CA	93101	34.426679	-119.711197	5.0	7	0	{'ByAppointmentOnly': 'True'}	Doctors, Traditional Chinese Medicine, Naturop...	None
1	mpf3x-BjTdTEA3yCZrAYPw	The UPS Store	87 Grasso Plaza Shopping Center	Affton	MO	63123	38.551126	-90.335695	3.0	15	1	{'BusinessAcceptsCreditCards': 'True'}	Shipping Centers, Local Services, Notaries, Ma...	{'Monday': '0:0-0:0', 'Tuesday': '8:0-18:30', ...
2	tJRfWHKlK_TAnsiWlNQQ	Target	5255 E Broadway Blvd	Tucson	AZ	85711	32.223236	-110.880452	3.5	22	0	{'BikeParking': 'True', 'BusinessAcceptsCredit...	Department Stores, Shopping, Fashion, Home & G...	{'Monday': '8:0-22:0', 'Tuesday': '8:0-22:0', ...
3	MTSW4McQd7CbVtyjqoe9mw	St Honore Pastries	935 Race St	Philadelphia	PA	19107	39.955505	-75.155564	4.0	80	1	{'RestaurantsDelivery': 'False', 'OutdoorSeati...	Restaurants, Food, Bubble Tea, Coffee & Tea, B...	{'Monday': '7:0-20:0', 'Tuesday': '7:0-20:0', ...
4	mWMc6_wTdE0EUBKIGXDVfA	Perkiomen Valley Brewery	101 Walnut St	Green Lane	PA	18054	40.338183	-75.471659	4.5	13	1	{'BusinessAcceptsCreditCards': 'True', 'Wheelc...	Brewpubs, Breweries, Food	{'Wednesday': '14:0-22:0', 'Thursday': '16:0-2...
5	CF33F8-E6oudUQ46HnavjQ	Sonic Drive-In	615 S Main St	Ashland City	TN	37015	36.269593	-87.058943	2.0	6	1	{'BusinessParking': 'None', 'BusinessAcceptsCr...	Burgers, Fast Food, Sandwiches, Food, Ice Crea...	{'Monday': '0:0-0:0', 'Tuesday': '6:0-22:0', '...
6	n_0UpQx1hsNbnPUSlodU8w	Famous Footwear	8522 Eager Road, Dierbergs Brentwood Point	Brentwood	MO	63144	38.627695	-90.340465	2.5	13	1	{'BusinessAcceptsCreditCards': 'True', 'Restau...	Sporting Goods, Fashion, Shoe Stores, Shopping...	{'Monday': '0:0-0:0', 'Tuesday': '10:0-18:0', ...
7	qkRM_ZX51YqxK3btIwAQlg	Temple Beth-El	400 Pasadena Ave S	St. Petersburg	FL	33707	27.766590	-82.732983	3.5	5	1	None	Synagogues, Religious Organizations	{'Monday': '9:0-17:0', 'Tuesday': '9:0-17:0', ...

Approach & Method

Task:

- Multi-class text classification

Models:

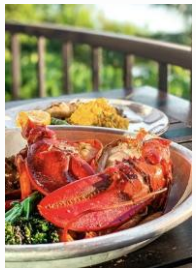
Baseline:

- CountVectorizer (unigrams)
- Logistic Regression

LLM:

- Gemma 3 (zero-shot prompting)

  No training required - uses prompt-based classification



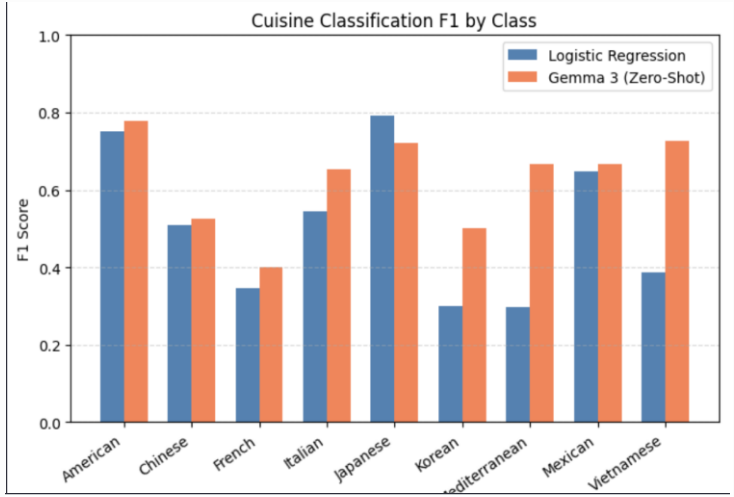
Results & Key Findings

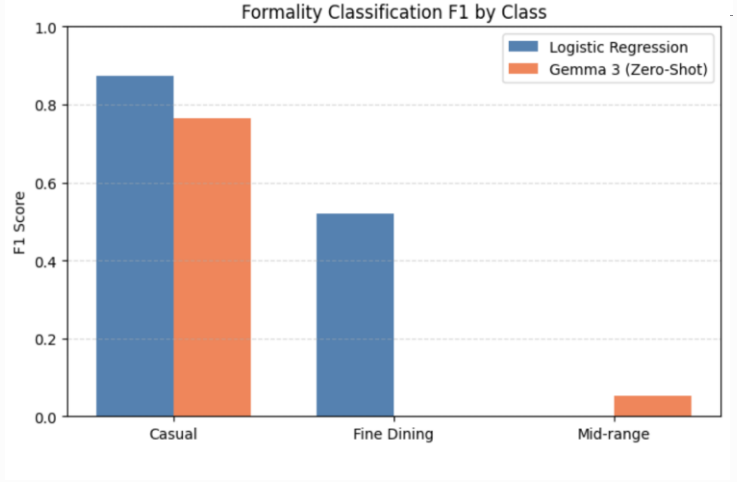
Results

System	Cuisine	Formality
Logistic Regression	66.1% (n=908)	78.9% (n=199)
Zero-Shot Gemma 3	71.3% (n=195)	53.8% (n=199)

Key Findings:

- LLM performs better on cuisine classification
- Baseline performs better on formality
- LLM never predicted Fine Dining (0/44)





Q&A

10. Kee, Aidan, Hannah, Brett, Raina

Movie Summarization

By: Raina, Brett, Hannah, Aidan, Kee

Motivation

1

Can check to see if summaries posted on sites like Wikipedia & IMDB are accurate

2

Generate summaries for movies that lack them

3

Remember key plot points of a movie you just watched

Data

- Custom dataset of movie subtitles and gold summaries
- Training set – 60% of data (832 rows)
- Dev set – 10% of data (93 rows)
- Test set – 30% of data (397 rows)
- Subtitles from Open Subtitles
- Summaries from Wikipedia
- 1322 unique data points
- Average 1255 lines of dialog per movie
- Average of 7942 words per movie

Methods

Baseline: Zero-shot prompting

Advanced: Parameter Efficient finetuning with LoRA

Results

- Finetuning led to no significant improvement
- Llama performed better than DeepSeek in both cases

Discussion and Limitations

- A key finding is that summary length heavily influences ROUGE scores, which may have over-rewarded Llama's naturally longer generations.
- LoRA ranks were kept low (8 for Llama, 16 for DeepSeek) to avoid memory errors and stay within time limits
- Dataset quality also impacted results - some low scores were traced back to mismatched or corrupted subtitle files rather than model failures.

2. Jay, Saung, Lyndsey, Amanda, Jonah

PitchPerfect

Predicting Song Artists from Lyrics

Using N-Gram and LLM Approaches

CS 1671 — Group 10

Amanda Cotumaccio · Jay Patel · Jonah Belback · Lyndsey Dippold · Saung Yati Oo

Project Motivation

Songs have unique styles — word choice, phrasing, and patterns differ between artists

Can a model learn to identify an artist just from their lyrics?

Useful for music recommendation, plagiarism detection, and authorship attribution

Task Description

Multi-class Text Classification

Input: Full song lyrics

Output: A single artist name

Classes: 13 artists

Example

Input:

"heartbreaker you got the best of me but i just keep on coming back..."

Output:

Mariah Carey

Data

13	1,300	100
Artists	Songs	Per Artist

- Source: Genius Song Lyrics dataset (Kaggle)
- Filtered to top 20 pop artists, then English-only songs
- Lyrics cleaned: lowercased, brackets removed, punctuation stripped
- Split: 70% train · 15% validation · 15% test
- Only lyrics and artist name used as input/label

Methods - nonLLM

All models use Logistic Regression as the classifier

ngram.py

Unigram + Bigram BOW

CountVectorizer with ngram_range=(1,2). Captures single words and pairs of words.

tfidf.py

TF-IDF Char N-Grams

TF-IDF with analyzer='char', ngram_range=(2,4). Captures spelling and stylistic patterns.

combined.py

Word + Char Combined

Stacks both word-level and character-level TF-IDF features together for best performance.

Methods - LLM

DistilBERT

- Pretrained DistilBERT encoder
  - Version of BERT that's supposed to be better at for the task
- Given first 512 tokens of lyrics
- Word Embeddings feed into a single layer Linear Regression for classification
  - Oversight that it wasn't MLP

BERT+MLP

- Pretrained BERT encoder
- Sliding window done over entire lyrics before averaged
  - Include encodings or all lyric's tokens
- Word Embeddings feed into a MLP for classification

Limitations:

- More required computation and memory

Results

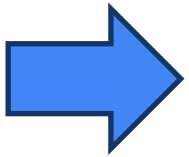
Model	Accuracy	Macro F1
Unigram + Bigram BOW	39.5%	0.38
TF-IDF Char N-Grams	46.2%	0.45
Combined Word + Char	**49.2%**	**0.48**
LLMv1 (DistilBERT)	20.0%	0.15
LLMv2 (BERT+MLP) ★	**57.0%**	**0.55**

Key Findings

- All models well above random chance (~8%)
- Char n-grams outperform word n-grams
- Acappella & New Model Army — easiest
- Kylie Minogue — hardest to classify

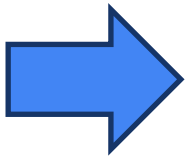
Limitations

- Dataset had to be reduced in size
  - Limited number of songs per artist
  - Subset of most popular artists
    - Doesn't include large diversity in genre
    - Doesn't include other languages



Restricts ability for models to model complexity

- Formatting of lyrics
  - Often short, repetitive
  - Structured on common themes within genres



Reduces amount of distinctive information