Can YouTube comment text predict political channel ideology?

CBS NLP Final Project, 2026

Research question: Can the lexical content of a YouTube comment alone predict the political ideology of the channel it was posted on?

Natural Language Processing and Text Analytics (CDSCO1002E)

Group Number: Fri-174161-2

Characters: 34,105

Number of pages: 15

Mikolaj Sapek: S185880

Julia Nowak : S160787

Yasemin Pagano: S185872

Peter Emil Larse Have: S160614

Table of Contents

1. Introduction

YouTube's advertising ecosystem operates on a staggering scale, generating $36.1 billion in revenue in 2024, with projections pointing to $42.8 billion by 2026 (Curry, 2026). Although the platform uses advanced algorithms to precisely categorize video content, this vast reach masks a significant oversight: the classification gap. Beyond the numbers and advanced categorization technologies, the consequences for advertisers can be profound. This stems from the fact that while videos are strictly moderated, the comments directly beneath them remain largely unclassified. A completely neutral, brand-safe video can easily appear above a thread that has devolved into a toxic political fight. This discrepancy poses a significant risk to brands due to the "spillover effect". This phenomenon, rooted in economics and sociology, suggests that the mood or tone in one context can bleed into another meaning that ads placed alongside highly polarized or toxic content can unknowingly inherit that negative association. Ads appearing in the context of such volatile political debates expose brands to unaccepted environments, slowly eroding their long-term image and equity.

Since manually checking billions of YouTube comments is nearly impossible, this puts advertising agencies and existing content positioning models in a difficult position. Evolution in this sector requires a shift from general, rigid "brand safety" which blocks entire political circles to conditional "brand relevance." A right-wing comment thread is unsuitable for EV advertising but highly targeted for tactical equipment brands. The purpose of this paper is therefore twofold: we aim to answer the research question and contribute to online discourse by examining whether the lexical content of a YouTube comment alone can predict the political ideology of the channel on which it was posted. Second, we aim to discover whether it is possible to build and implement an automated machine learning-based classifier that can accurately detect, at the channel level, whether a given thread has a distinct left- or right-wing bias. Such a solution would enable the unattended labeling of high-risk assets, directly optimizing CPMs and protecting brand reputations from damaging associations.

We declare that GenAI was used for idea creation and conceptualisation of the project structure. GenAI was also used to assist with developing the modelling pipeline code, which the group reviewed, tested, and understood.

2. Literature review

Prior work has examined whether YouTube comments carry a political signal, but the commercial implications are unexplored. Three papers anchor our framing. Chae and Lee (2024) study cross-partisan political comments on YouTube around the 2019 Mueller report. Their corpus is 17 videos: 10 from political vloggers (5 liberal, 5 conservative) and 7 from mainstream news outlets (CNN, MSNBC, Fox News, LiveNOW from Fox, and C-SPAN). They manually code 1,000 vlogger comments (Gwet's AC1 =

0.835 on a 200-comment double-coded subset) and classify 4,230 mainstream-news comments using logistic regression, SVM and random forest, all on SBERT embeddings. They report that the rate of cross-cutting discussion varies with both the channel's political leaning and its media type, and that neutral outlets host more cross-partisan exchange. Their goal is to measure cross-partisan participation, not to ship a deployable classifier. Our project is the deployment-focused counterpart, applied to 301,536 comments across seven channels.

Yaman (2024) introduces LAMINEX, a hybrid framework that combines language-model output with network-based features to estimate political ideology on X. The dataset links 3,938 survey respondents to their X handles and a self-reported seven-point ideology score. Content-only models perform poorly: fine-tuned DistilBERT reaches only 0.12 correlation with the survey score. Combining ChatGPT classification, network correspondence analysis and VADER sentiment lifts that to 0.65. We take from Yaman the result that LLMs can extract some ideological signal from text, while noting that our task is binary channel-level classification on a much larger corpus, not continuous user-level scoring on a smaller one. Work on partisan news consumption on social media shows that audiences tend to mirror the political pattern of the outlets they engage with. Shin (2020), studying Twitter users with linked self-report and digital-trace data, finds clear selective-exposure patterns in the news that partisans follow. We carry the same intuition over to YouTube comment classification, with seven channels and five predictive models trained for advertising-placement decisions.

3. Theory

The five models below cover a spectrum from raw word counts to large pre-trained language models. The point of running all five is not to find the best one, but to see whether a political leaning is detected at each level of linguistic sophistication.

3.1 Bag of Words and Complement Naïve Bayes

Bag of Words is the simplest text representation in this study. Each document is encoded as a vector of word counts, with no information about word order or syntax. The classifier paired with it is Complement Naïve Bayes, which selects the class with the highest posterior probability, assuming that words are conditionally independent given the label (Jurafsky & Martin, 2014). The Complement variant estimates word probabilities from the complement of each class, which is more stable on imbalanced data. Even with the strong independence assumption, a simple word-count classifier is a useful baseline: if it can already separate the two classes, the data is lexically distinguishable.

3.2 LinearSVC with TF-IDF and n-grams

As a stronger linear model, we use a Linear Support Vector Classifier on TF-IDF features extended to bigrams and trigrams. TF-IDF down-weights terms that appear uniformly across documents and amplifies class-distinctive ones. Shin (2020) used the same feature family to identify divergent keyword usage between CNN and Fox News audiences. Adding bigrams and trigrams lets the model capture short politically loaded phrases such as “gun control” or “open borders” that unigrams alone would miss. The

model remains linear and cannot capture context-dependent meaning, but it scales well and is interpretable via coefficient inspection.

3.3 Word2Vec embeddings with Logistic Regression

Word2Vec represents each word as a dense vector trained such that words appearing in similar contexts end up near each other in the vector space (Jurafsky & Martin, 2014). The training objective here is skip-gram, which predicts surrounding words from a target word (Mikolov, 2013). Each comment is represented as a TF-IDF-weighted mean of its word vectors and passed to a Logistic Regression classifier. The representation is more semantic than a bag of words, but it has two known limitations. Embeddings are static, so a word has the same vector regardless of context and averaging discards word order, so the classifier never sees how words are arranged.

3.4 DistilBERT fine-tuning

DistilBERT is a distilled version of BERT, the bidirectional transformer of Devlin et al. (2019). Each word's representation is computed by attending to every other word in the sequence at once, weighted by relevance. This is what static embeddings cannot do, and it is the main reason for trying a transformer at all. DistilBERT keeps roughly 97% of BERT's performance with 40% fewer parameters (Sanh et al., 2019). We train on a balanced subsample of 100,000 comments (50,000 per class) drawn from the 241,228-comment training partition. Fine-tuning adds a classification head on top of the pre-trained model and updates all weights using our labelled data. Yaman (2024) reported only 0.12 correlation for DistilBERT on user-level ideology estimation on Twitter, but our task is different: binary classification at the channel level on a much larger training set, which we expect to give a stronger and more consistent signal.

3.5 GPT-4o-mini few-shot prompting

Our final approach is a prompted general-purpose LLM. Unlike the other four models, GPT-4o-mini does not see our training data. It relies entirely on whatever it learned during pre-training. We test two configurations: zero-shot, with just an instruction, and few-shot, with ten labelled examples (five per class) included in the prompt. The aim is to check whether existing LLM knowledge of political language is enough on its own, given that Yaman (2024) found GPT-4 useful only when combined with other signals.

3.6 Latent Dirichlet Allocation (LDA) topic modelling

As a supplementary analysis, we run LDA separately on left-leaning and right-leaning comments. LDA represents each document as a mix of topics, and each topic as a probability distribution over words (Blei, 2003). Running LDA on each side separately tells us what each one talks about, which is a qualitative complement to the classification numbers (Blei, 2012).

4. Data Acquisition

Our dataset consists of 375,435 comments scraped from seven US political YouTube channels in April 2026. We used yt-dlp rather than the official YouTube Data API v3, which caps daily requests at 10,000 units. Channels were selected to span the all sides of the media bias spectrum from centre-left to right while keeping the set small enough for binary classification. The seven channels are: CNN (centre-left, news), David Pakman and The Young Turks (left, commentary), Fox News (right, news), and PragerU, The Daily Wire and Tim Pool (right, commentary). Per-channel video coverage ranges from 86 to 834 distinct videos, depending on how concentrated each channel's audience activity was on individual videos. Each row in the raw dataset contains eight features: channel_name, channel_bias, channel_format (news vs commentary), video_title, is_reply (boolean), like_count (integer), language (auto-detected) and the comment text. Bias labels follow AllSides (2023) Media Bias Ratings and are assigned at the channel level, then propagated to every comment beneath it.

5. Exploratory Analysis

Before any modelling, we audited the raw dataset. Several patterns emerged that shaped the preprocessing decisions described below.

5.1 Duplicates

The raw collection contained substantial duplication: 40,422 exact row duplicates (10.8% of the dataset), and a further 49,714 comments with identical text appearing across different channels, suggesting bot-driven cross-posting or copy-paste behaviour by users active in multiple comment sections. We removed duplicates on the text column alone, which is stricter than deduplication and handles cross-channel copies in a single pass. After this step, 73,889 rows were gone, leaving 301,546 unique-text comments going into language and length filtering.

5.2 Language

Langdetect tagged 342,275 comments as English, marked 5,064 with no language label, and spread the rest across 31 other languages. The detector performs poorly on short text: “Are you serious?” came back as French, “Thank you Ana.” as Tagalog, and “fake” as Norwegian. We kept only comments tagged English or with no label, ending up with 301,536 comments after deduplication. The tradeoff is losing roughly 26,000 short English comments that the detector misclassified.

5.3 Comment length and like counts

Comment length is heavily skewed. The median is 15 words, but 10,232 comments (2.7%) contain a single word or nothing useful at all (“BS”, “👍👍👍”, “No”). Like counts follow the same pattern: 57.9% of comments received zero likes, while a handful reached the thousands, with one CNN comment peaking

at 11,000. We did not filter on like count for modelling, though a minimum-length filter would be a reasonable next step.

5.4 Class balance after deduplication

The cleaned dataset contains 137,143 left-leaning comments (45.5%) and 164,403 right-leaning comments (54.5%). The imbalance is moderate but worth noting. On the left side, David Pakman alone contributes 46,559 comments, which means any model trained here will have a particularly strong signal about what left-leaning language looks like according to his audience specifically.

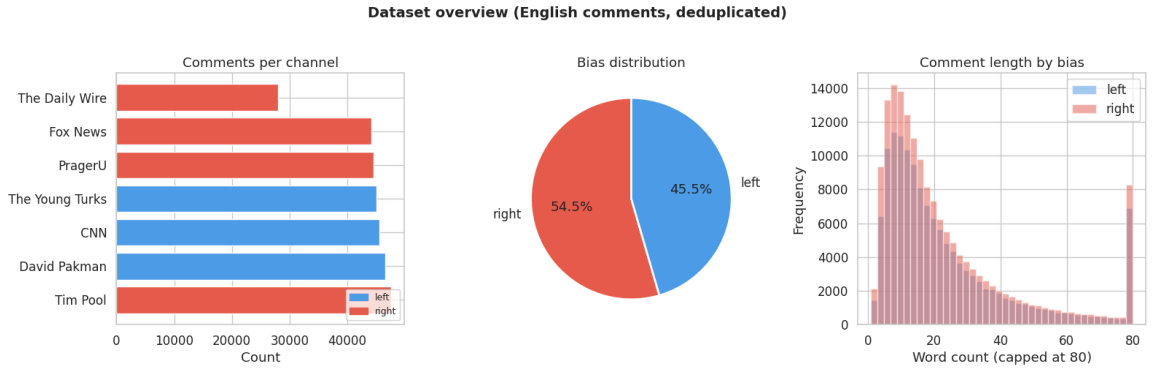


Figure 1. Dataset overview: comments per channel, bias distribution, and comment-length distribution by ideological side. Channel counts include CNN merged into the left class.

5.5.Vocabulary fingerprint by ideology

We computed the top 25 TF-IDF(Figure 2) terms per class as a first look at whether the two sides are lexically distinguishable without any classification yet. Shared vocabulary exists (“trump”, “biden”, “iran”, “israel”, “war”), but beyond that the two sides diverge noticeably. Right-leaning comments are marked by a cluster of religious and patriotic words (“god”, “bless”, “president”, “thank”, “great”) that simply do not appear on the left with comparable frequency. Left-leaning comments instead emphasise policy and economic terms (“money”, “pay”, “oil”, “workers”). This early separation suggests classification task is feasible even with simple features.

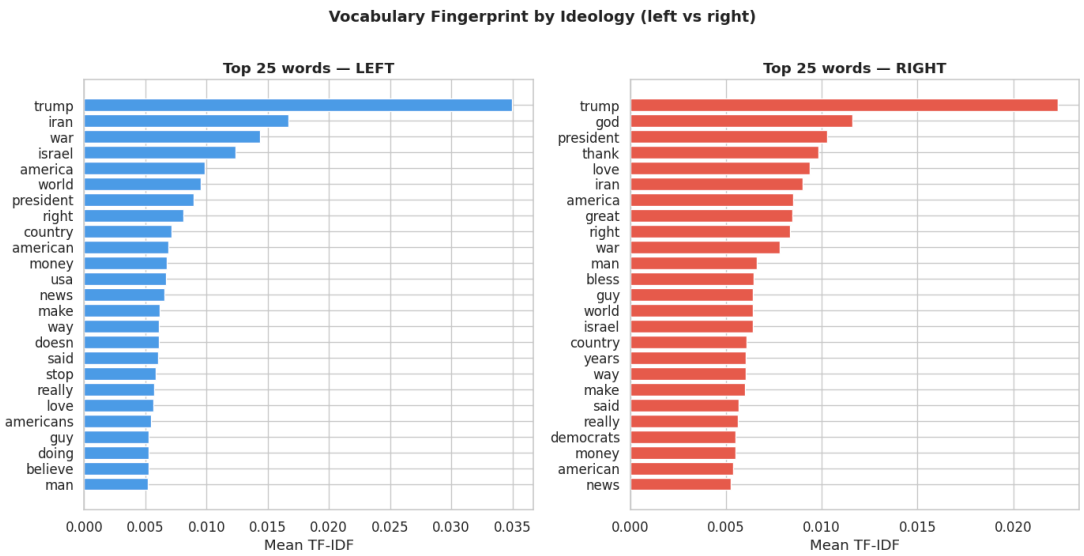


Figure 2. Top 25 TF-IDF terms per ideological class.

6. Preprocessing and final dataset

Cleaning happened in two steps. First we filtered and relabelled the rows, then we cleaned the comment text itself. Using random_state = 42 everywhere so the results can be reproduced exactly.

Stage 1: dataset-level cleaning

Five operations were applied to the 375,435 raw rows in order. First, the is_reply column was dropped since none of the five models use reply structure. Second, CNN was moved from centre-left to left, which collapses the label space into a clean binary. Third, duplicates were removed based solely on the text column, so a comment appearing under two different channels counts as one instance and is deduplicated in a single pass, addressing both bot spam and cross-channel copy-paste at once. Fourth, comments under three tokens were discarded, since one-word reactions carry no usable lexical signal. Fifth, we kept only comments tagged en or with a missing language label. The filter is tightened to strict English for the modelling subset.

Stage 2: text-level cleaning

A single clean_text() function processes every comment. Each comment goes through lowercasing, URL removal, replacement of punctuation, emoji and symbols with whitespace, whitespace collapsing, whitespace tokenisation, and finally stop-word removal. The pipeline outputs two cleaned versions of each comment. The difference between them is the stop-word list. text_clean_baseline strips only a short set of 14 common generic words (people, just, like, don, know, think, going, good, want, did, say, does, time, need), leaving partisan vocabulary like “trump”, “biden”, “maga” and “democrats” in place since those are exactly the terms that should carry predictive weight. text_clean goes further and also removes a 47-token

CHANNEL_STOP list covering host names and brand tokens (cenk, pakman, prageru, shapiro, tucker, carlson, cnn, fox, hannity, ingraham, kasparian, uygur, owens, walsh, knowles, among others). This debiased version is used in the cross-channel ablation. The performance gap between the two versions is how we measure the channel-identity effect reported later.

Final dataset

After both cleaning stages the modelling dataset contains 301,536 English comments. Each row carries the raw text, both cleaned-text variants, channel and bias labels, the format label (news vs commentary), like_count, language, video_title, and the binary target (0 for left, 1 for right). Class imbalance is handled at the model level rather than by resampling: LinearSVC and Logistic Regression use class_weight= "balanced", Complement Naïve Bayes handles imbalance by construction, and DistilBERT is trained on a balanced subsample of 50,000 comments per class. The full dataset is split 80/20 with stratification (train_test_split, stratify=y, random_state=42), giving 241,228 training comments and 60,308 test comments with class proportions preserved across all four supervised models.

7. Experimental setup

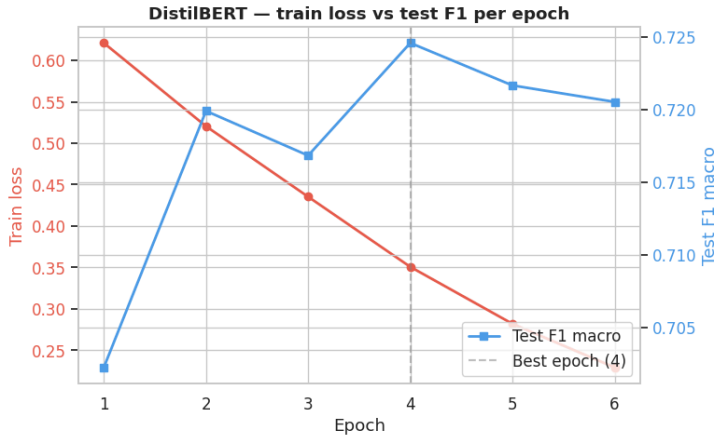


Figure 7. DistilBERT training loss per epoch.

Despite training a 67-million-parameter transformer with full self-attention on 100,000 labelled comments, DistilBERT delivers only a marginal improvement over a sparse linear classifier. We report two primary metrics: overall accuracy and macro-averaged F1. Macro F1 weights both classes equally and is robust to the moderate class imbalance in the test set.

All five models share the same train/test split. All classical models were implemented using scikit-learn (Pedregosa et al., 2012).We additionally designed a cross-channel split to test whether the models' predictions hold up on unseen sources. Experiments are reproducible from the cleaned dataset and the notebook (nlp_final.ipynb). Random seeds are fixed at random_state=42 for the train/test split, the stratified samples, Word2Vec training, DistilBERT subsampling and the GPT evaluation sample.

End-to-end runtime on the workstation is roughly 145 minutes: about 3 minutes for the classical models, around 100 minutes for DistilBERT, and the rest for Word2Vec and the GPT calls.

8. Results

Model 1: Complement Naïve Bayes with Bag of Words

The baseline uses a CountVectorizer with unigrams and bigrams, vocabulary narrowed to 50,000 features, and a minimum document frequency of 2. Bigrams catch short politically loaded expressions like “fake news”, “border crisis”, and “open borders” that unigrams independently miss. We use Complement Naïve Bayes with a smoothing parameter alpha = 0.1 since the complement formula handles class imbalance better than the standard multinomial formula. Even with a strong independence assumption, the baseline achieves 0.707 accuracy and 0.706 macro F1 on the held-out test set (Appendix 1). The model is more specific in right-leaning predictions, achieving 0.75 precision, while it is less certain in left-leaning predictions, achieving 0.66 precision score. This asymmetry occurs across all classical models and mirrors the underlying class imbalance.

Model 2: LinearSVC with TF-IDF word and character n-grams

The second model is a Linear Support Vector Classifier trained on two concatenated TF-IDF feature domains: word n-grams of length 1 to 3 (80,000 features, sublinear term-frequency scaling) and character n-grams of length 2 to 4 in word-boundary mode (40,000 features). The combined feature matrix has roughly 90,000 dimensions per comment. Character-level features are highly beneficial in the YouTube-comment domain because they treat slang, typos, and morphological variants such as “biden”, “bidens”, and “obiden” as related tokens rather than separated dictionary entries. LinearSVC with class_weight= “balanced” and regularisation C = 0.5 reaches 0.719 accuracy and 0.718 macro F1, an improvement of about 1.3 percentage points over the Naïve Bayes baseline (Appendix 2). There is the same precision asymmetry between left and right.

Features pushing predictions toward the right class include “president trump”, “leadership crisis”, “civil war”, “phil” and “never faced”. The strongest left-pushing features include “larry johnson”, “king george”, “project 2025”, and “right to exist”. As those phrases are not subtle stylistic markers, they explicitly express a political stance.

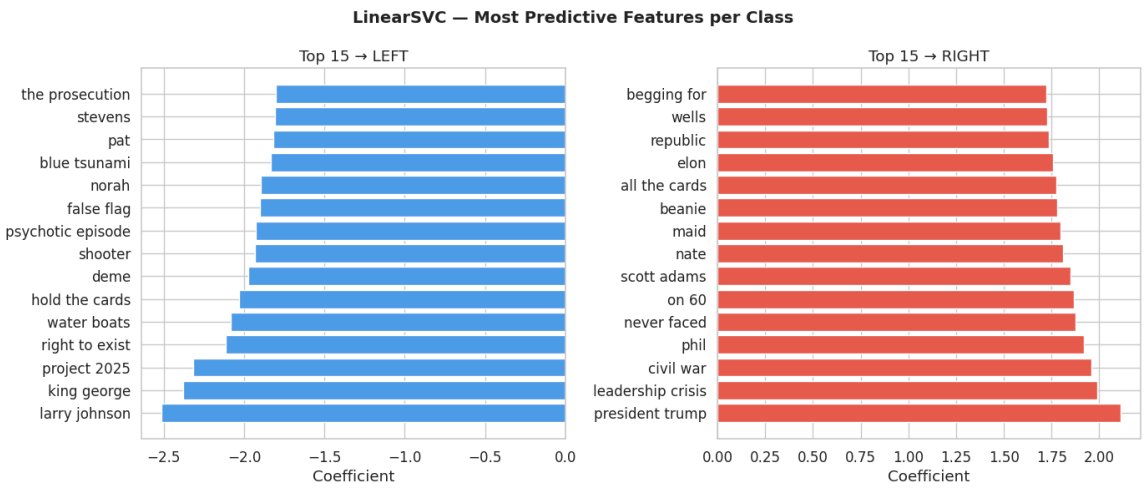


Figure 5. Most predictive features for each class, ranked by LinearSVC coefficient magnitude.

Model 3: Word2Vec embeddings with Logistic Regression

We trained 300-dimensional Word2Vec embeddings using the skip-gram objective (window = 7, min_count = 3, 10 epochs) on the training portion of the corpus, yielding a vocabulary of 31,031 words. Each comment is represented as a TF-IDF-weighted mean of its word vectors, where rare but discriminative words ("prageru", "cenk") get higher weight than standard ones. The pooled vectors are classified with Logistic Regression (class_weight= "balanced", C = 2.0).

This pipeline achieves 0.664 accuracy and 0.664 macro F1, the lowest scores among all supervised models (Appendix 3). The gap can be explained by two factors: firstly, mean-pooling discards word-order information that the n-gram representations maintain; secondly, the learned vector space is noticeably noisy: nearest neighbors of "trump" in our corpus include the misspelled "pedifile" and "because", alongside slang and conspiracy-theory tokens like "krasnov". A min_count of 3 admits a long tail of typos into the vocabulary, which cuts embedding quality. Larger pretrained embeddings (e.g. GloVe Twitter 200d) would likely close part of the gap.

Model 4: Fine-tuned DistilBERT

We fine-tuned distilbert-base-uncased on a balanced subsample of 100,000 training comments (50,000 per class). Hyperparameters: max sequence length 128, batch size 64, learning rate 2e-5, weight decay 0.01, linear warmup over 10% of training steps, gradient limiting at norm 1.0. Total training time on a single RTX 4070 was about 100 minutes; training was capped at seven epochs, but early stopping (patience = 2) terminated it after epoch 6 (Appendix 4). Training loss decreased across the six completed epochs, from 0.621 at epoch 1 to 0.229 at epoch 6. Once we counted per-epoch evaluation on the held-out test set. We could see it more clearly: test F1 plateaus around epochs 3-4, while the training loss keeps dropping, indicating overfitting. Early stopping (patience = 2) terminated training, and we reported the best-epoch result.

Model 5: GPT-4o-mini with zero-shot and few-shot prompting

We tested OpenAI's gpt-4o-mini in two prompting setups on a balanced random sample of 102 test comments (51 per class). Zero-shot performance cannot be distinguished from random guessing: 0.500 accuracy and 0.500 macro F1 on the binary task. Few-shot prompting has improved the accuracy to 0.520 and macro F1 to 0.519. It cannot be relied upon to contrast between them on raw, short, partisan-coded comments. Furthermore, a state-of-the-art general-purpose LLM, applied without fine-tuning, does not match a ten-year-old Naïve Bayes baseline. The 20-percentage-point gap between Naïve Bayes (0.706) and GPT-4o-mini zero-shot (0.500) is not a question of model capacity. The bottleneck is the lack of a supervised training signal. The model is unable to learn how to apply "left" versus "right" beyond what it can deduce from the prompt itself. Though reaching parity with supervised methods would require either fine-tuning or a much larger few-shot context.

8.1 Cross-channel generalisation

The standard 80/20 split allocates training and testing comments across all seven channels. The model can learn channel-specific stylistic fingerprints rather than transferable ideological language. We designed a more rigorous cross-channel evaluation in which the model never sees the test channels during training to measure how much of our reported performance is genuinely ideological.

Training set: The Young Turks (left), Fox News (right), PragerU (right), and 80% of CNN (left). Test set: David Pakman (left), The Daily Wire (right), Tim Pool (right), and 20% of CNN (left).

The CNN split is stratified to keep that one channel balanced across both partitions; every other channel appears only in train or only in test. We ran two conditions: a baseline using text_clean_baseline (channel names preserved) and a debiased condition using text_clean with the CHANNEL_STOP list applied to remove host names and brand tokens. Three observations stand out. First, generalizing to unseen channels costs 12.6 macro-F1 points. That fraction of the within-channel performance was channel-specific, not ideology-specific. Second, removing host and brand names did not close this gap. It actually led to a small additional drop of 0.7 points. Simply scrubbing channel names is not enough to remove channel identity from text, because the surrounding language still carries channel-specific topical and stylistic patterns. Moreover, the debiased model still performs well above the random-baseline F1 of 0.5, confirming that a genuine, transferable ideological signal exists; however, it is just smaller than the within-channel result alone would suggest.

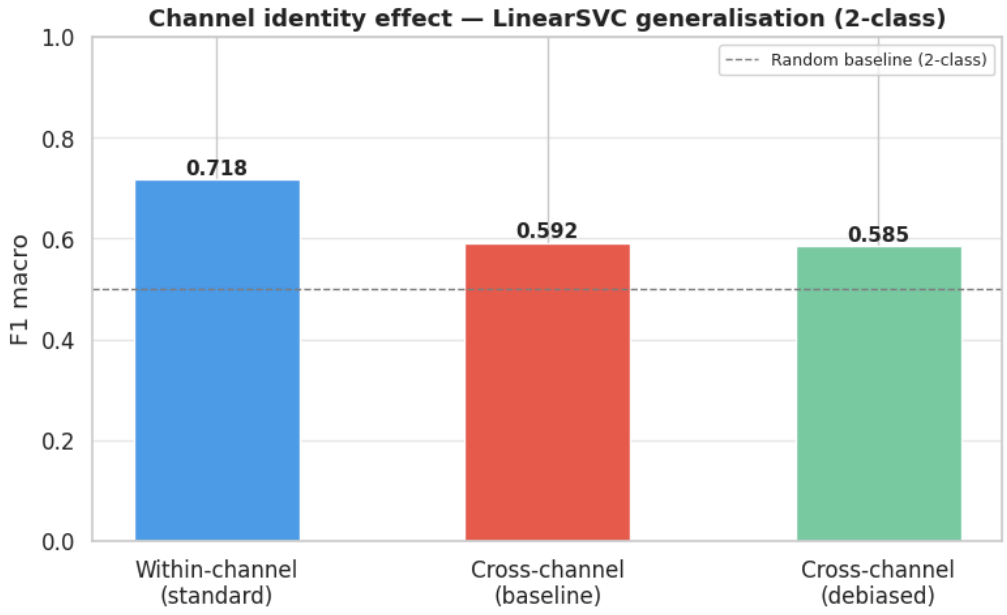


Figure 9. F1 macro under three evaluation regimes for the LinearSVC pipeline.

8.2 Model comparison

Bringing all five models onto a single chart makes the structure of the results visible at a glance. Performance falls into three tiers: the two strong supervised models (LinearSVC at 0.718 macro F1 and DistilBERT at 0.725, only 0.7 points apart), the weaker supervised models (Naïve Bayes at 0.706 and Word2Vec at 0.664), and the prompted GPT models (0.500 and 0.519).

Model	**Accuracy**	**F1 macro**
ComplementNB + BoW	0.707	0.707
LinearSVC + TF-IDF word+char	0.719	0.718
Word2Vec skip-gram + TF-IDF LR	0.664	0.664
DistilBERT fine-tuned	0.726	0.725
GPT-4o zero-shot	0.500	0.500
GPT-4o few-shot	0.520	0.519

Figure 10 Table comparison of accuracy and macro F1 across all five models.

8.3 Error analysis

Of the 59,511 test comments, the three supervised classifiers classifications 7,634 (12.2%) Inspecting these hard examples reveal three recurring: (i) zero political content for example “what show is this”

posed no position for the ideological signal; (ii) comments that rely on external knowledge no model has, such as references to specific controversies or off-platform events; (iii) generic emotional reactions with no clear ideological vocabulary, e.g. “I love her”, “savage”, “fake news maga fake news”. DistilBERT slightly outperforms LinearSVC overall (macro F1 +0.005). The two models disagree on 23.1% of the test set, since they learned different decision boundaries: LinearSVC relies on explicit partisan keywords, while DistilBERT is better at capturing implicit contextual and stylistic cues.

8.4 LDA topic modelling

We ran LDA separately on a 30,000-comment sample from each ideological class, with five topics per class, eight displayed words per topic, and the standard NLTK English stopword list applied. The intent is qualitative: we want to see what each side discusses, not to predict. The dominant token across almost every topic on both sides is “Trump”. Beyond that, the topical breakdown diverges in informative ways. Left-leaning topics cluster around conflict, foreign policy and economics: “Iran”, “Israel”, “oil”, “money”. Right-leaning topics include a comparable foreign-policy cluster but also feature one topic that has no equivalent on the left, dominated by religious-patriotic vocabulary: “god”, “bless”, “president”, “thank”, “love”, “great”, “man”. Interestingly, the method is distinct from EDA in Figure 2, yet they arrive at the same conclusion, indicating that the findings are more trustworthy. Both sides talk about Trump consistently; only the right surrounds him with religious-patriotic language.

9. Discussion

9.1 Lexical features perform comparably to contextual features here

LinearSVC and DistilBERT achieved closely comparable macro F1 scores (0.718 vs 0.725), so BERT's contextual representations offer only a marginal advantage on this task. That is notable because DistilBERT's self-attention is specifically built to capture word meaning in context, yet it produced only a small gain over a linear keyword-based model. Two explanations are plausible. The first is corpus-driven: YouTube comments are short, heavily partisan and lexically explicit, so the ideological signal is carried mostly by surface vocabulary rather than context. Word and character n-grams are enough to capture it. The second is methodological: our DistilBERT configuration was not extensively tuned, and changes to the warmup schedule, batch size, or learning-rate decay could plausibly improve performance. We favour the corpus-driven explanation, because both models show the same precision-recall asymmetry across classes. They struggle on the same subset of comments for the same reasons, not for model-specific ones.

9.2 Generalist LLMs do not solve this task without supervision

GPT-4o-mini under zero-shot prompting performs at chance (0.500) on a balanced two-class sample. Both numbers fall well short of the Naïve Bayes baseline trained on the same kind of data. GPT has World knowledge and large-scale pretraining but it is not a substitute for in-domain training when the signal is short, noisy and partisan-coded. This is consistent with Yaman (2024), who found GPT-4 useful only in combination with network-based features.

9.3 Channel identity is real but does not vanish on token removal

Generalising to unseen channels costs 12.6 macro-F1 points (about 17.5% of within-channel performance). Removing host and brand names (Tucker, Shapiro, Cenk), the most obvious channel-identity tokens, recovers none of that loss. Each channel leaves a deeper footprint in the text than just host names. The way audiences write, the topics they focus on, and how they combine those topics all carry channel identity. Simply removing brand names does not make a model fair because the channel signal is still fully present in everything else.

9.4 Right-class precision is systematically higher than left

Across all four supervised models, right-class precision exceeds left-class precision by 7 to 9 percentage points. This follows from the moderate class imbalance (54.5% vs 45.5%) and is consistent with how probabilistic and margin-based classifiers behave on imbalanced data. Threshold tuning or cost-sensitive loss could close the gap, though it would trade overall accuracy for balanced precision.

9.5 The religious-patriotic cluster is a strong right-only topic

Three independent analyses converge on the same observation: top-25 TF-IDF terms, LinearSVC coefficients, and unsupervised LDA topics. The right-leaning class contains a religious-patriotic vocabulary cluster (“god”, “bless”, “president”, “thank”, “love”, “man”, “great”) with no symmetric counterpart on the left. Both sides discuss Trump almost equally, but the affective register surrounding him diverges sharply. This is the qualitatively most informative finding for understanding why classification works at all. The convergence across one supervised lexical analysis (TF-IDF), one supervised model-based analysis (LinearSVC coefficients), and one unsupervised analysis (LDA) is methodologically robust: the cluster is not an artefact of any single feature representation or learning algorithm.

10. Limitations

This research highlights 3 weaknesses: channel labels are not a reliable substitute for the commenter's ideology, models do not generalise to unseen channels, and GPT evaluation is too small to be conclusive. First, the labelling scheme assigns every comment the ideology of its host channel, assuming all commenters share that channel's political stance. An example of this is the error analysis identified cases where cross-ideological comments, such as anti-Trump slang, were labeled right cause it was on Fox News. The classification task is more accurately described as predicting the channel's editorial bias from a comment rather than predicting the commenter's personal ideology. Any model trained on these labels is partly learning channel identity rather than genuine ideological stance. The true classification error rate is higher than the reported figures suggest, and all results should be interpreted with this ceiling in mind.

Secondly, the model struggled to generalise to unseen channels, as within-channel the F1 Macro score of 0.718 dropped to 0.592 when tested on a channel not seen during training, a loss of 12.6 points representing 17.5% of within-channel performance. Removing host and brand names does not recover this gap. The debiased condition scores 0.585, marginally worse than baseline, confirming the model learned

channel-specific writing style rather than transferable ideology. This could be that each Channel audience has a distinct digital footprint in terms of tone, possibly similar to the channel's ideology, following stylistic patterns and vocabulary choices that survive name removal and cannot be neutralised by a stop-word list. All reported results are therefore limited to within-distribution settings. A model trained on these seven channels would likely perform near random on any unseen political channel, limiting real-world applicability.

Per-epoch evaluation during training selects the best checkpoint based on test-set F1, which technically leaks the test set into model selection. The best epoch was selected at epoch 4 with a reported F1 of 0.727. After every epoch, the model checked its performance on the test set and kept whichever version scored highest. This means the test set was used to make decisions during training, not just to evaluate the final result. These two things should be kept separate. A clean three-way split of train, validation and test would have produced a slightly lower but more honest estimate.

11. Conclusion

Returning to the original research question: yes, the lexical content of a YouTube comment alone is sufficient to predict the political ideology of the channel it was posted on. Macro F1 is around 0.72 to 0.73 in the standard within-channel setting, and around 0.59 when the model must generalise to channels it has never seen. LinearSVC + TF-IDF (0.718) and a fine-tuned DistilBERT (0.725), perform comparably. The transformer offers only a marginal gain over a sparse linear classifier. A general-purpose LLM applied without supervision performs at chance.

For the business application that motivated the project, automated ideological labelling of comment sections to inform brand-safety and brand-relevance decisions, these results suggest that a lightweight, linear, easily interpretable classifier is sufficient. The signal is lexical, and the marginal value of heavier architectures is small.

Three extensions would meaningfully strengthen the analysis. First, an ensemble of LinearSVC and DistilBERT, weighted by complementarity in their error sets. Second, a three-class formulation that retains CNN as a centre-left class rather than collapsing it into the left, which would let the model express ideological gradation rather than a hard binary. Third, a deployment-style cost analysis: at YouTube's billion-comments-per-day scale, the difference between a sparse linear model (milliseconds per comment) and a fine-tuned transformer (tens of milliseconds per comment, GPU required) is the difference between tractable and intractable.

References

AllSides. (2023). *AllSides Media Bias Ratings*. AllSides. https://www.allsides.com/media-bias

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022.

Blei, David M. (2012). "Probabilistic Topic Models." *Communications of the ACM*, vol. 55, no. 4, 1 Apr. 2012, pp. 77–84, www.eecis.udel.edu/~shatkay/Course/papers/UIntrotoTopicModelsBlei2011-5.pdf, https://doi.org/10.1145/2133806.2133826.

Chae, S. W., & Lee, S. H. (2024). Where do cross-cutting discussions happen? Identifying cross-cutting comments on YouTube videos of political vloggers and mainstream news outlets. PLOS ONE, 19(5), e0302030. https://doi.org/10.1371/journal.pone.0302030

Curry, D. (2026, January 7). *YouTube revenue and usage statistics (2022)*. Business of Apps; Business of Apps. https://www.businessofapps.com/data/youtube-statistics/

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Jurafsky, Daniel, and James Martin (2026). Speech and Language Processing an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models Third Edition Draft Summary of Contents.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems 26.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. *ArXiv:1201.0490 [Cs]*. https://arxiv.org/abs/1201.0490

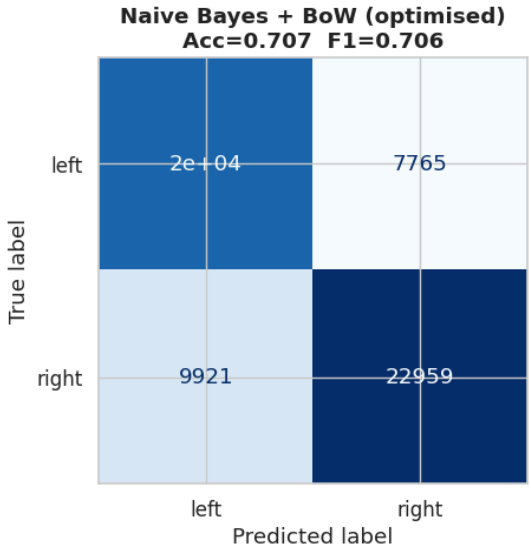
Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108.

Shin, J. (2020). How do partisans consume news on social media? A comparison of self-reports with digital trace measures among Twitter users. Social Media + Society, 6(4). https://doi.org/10.1177/2056305120981039

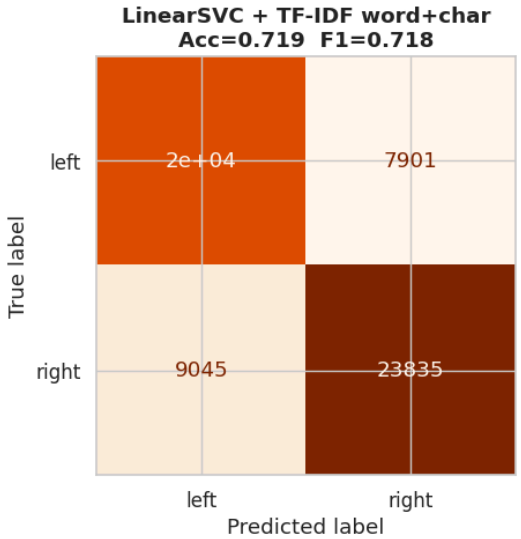
Yaman, S. (2024). Beyond the tweets: Leveraging language models for estimating political ideology on X. Working paper, January 2024. https://selimyaman.com/uploads/Using_Language_Models_to_Predict_User_Ideology_on_Twitter.pdf

Appendices:

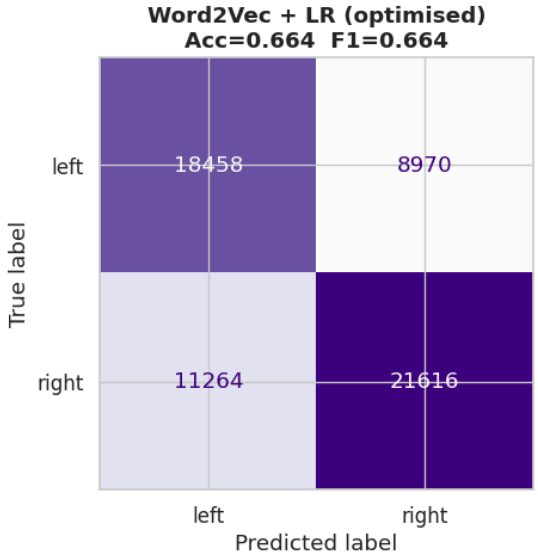
Appendix 1: *Confusion matrix for ComplementNB + BoW.*



Appendix 2: *Confusion matrix for LinearSVC with TF-IDF word and character n-grams.*



Appendix 3: Confusion matrix for Word2Vec + Logistic Regression.



Appendix 4: *Confusion matrix for fine-tuned DistilBERT.*