CS445 Final Report

Task 9: Detecting Multilingual, Multicultural, and Multievent Online Polarization

Group 25:

Mohamad Samer Bader - 30437

Eid Alhamali - 30482

Lyan Alhavasli - 32714

Ata Nuri Koçoğulları - 30799

Yasin Mirioğlu - 30623

1. Introduction

Online polarization detection aims to identify short user-generated messages that show a clear polarized attitude toward social or political issues. In this project, we follow the SemEval Task 9 about sentence-level polarization detection, where given a short post, we need to predict whether it is polarized or not. We treat polarization as language that reflects a strong "us vs. them" stance in a hostile way about people's beliefs, noting that it is related to, but different from, toxicity and subjectivity.

A core challenge is multilingual generalization, where the task includes many languages and writing systems, including lower-resource settings where purely supervised training tends to be weaker. To address this, our system uses a teacher-student knowledge distillation approach, consistent with our milestone plan of using a stronger multilingual teacher to guide a smaller student model. Essentially, we train Teacher A (XLM-R Large) on a carefully selected set of 7 high-resource languages (eng, spa, deu, rus, tur, pol, arb) using both the original labeled data and synthetic augmentation. We then distill its knowledge into a compact Student (XLM-R Base) that is used alone at test time (teachers are discarded). This preserves the inference efficiency while still transferring cross-lingual polarization hints learned by the teacher to the student. Compared with our milestone proposal of using two teachers (a language expert + a topic expert), we decided to drop the topic-based Teacher B after discovering a label leakage. The topic/type-vector features were informative of the label (topic presence almost implied polarization), yielding misleadingly inflated validation performance. Therefore, our final pipeline relies on Teacher A only with ground-truth supervision and implements a safe "partial distillation" rule, where we distill only when an English translation (text-en) exists, otherwise we train on hard labels only. On the official Codabench development evaluation of 22 languages, our final student model achieved an average Macro F1 score of 0.783 and an average Accuracy of 0.816.

2. Related Work

Our work builds on the POLAR shared-task 9 setting for online polarization, where polarization is context-dependent across languages and cultures. Earlier studies approach polarization from different angles, for example, He et al. (2021) studies topic-level polarization in partisan COVID-19 news with PaCTE, which uses partisanship-aware contextual topic embeddings to rank topics by ideological separation. This motivated our focus on robust signals that generalize across domains and languages, even though our task is sentence-level classification rather than topic ranking.

Polarization is related but not identical to toxicity, creating label ambiguity. Nugraha et al. introduces a multi-labeled Indonesian discourse dataset and shows that auxiliary labels help keep emotional content from being mistaken for true polarization. Later work finds that jointly modeling toxicity and polarization and using contextual metadata can improve performance, which supports our decision to remove any features that leak polarization labels (Section 3-4) rather than adding extra supervision blindly.

Our models are based on XLM-R (XLM-RoBERTa), a multilingual masked language model pretrained at scale. Conneau et al. shows that scaling multilingual pretraining improves cross-lingual transfer, even for lower-resource languages, motivating our choice of XLM-R Large as teacher and XLM-R Base as student. For compression and transfer, we used teacher-student knowledge distillation (Hinton et al., 2015), and were inspired by Zhang et al. (2023), who proposed a dual distillation framework for cross-lingual, cross-target stance detection, where we simplified this to a single reliable teacher after

detecting leakage in the second pathway. For contextual signals, we initially considered BERTopic (Grootendorst, 2022), which clusters transformer embeddings and derives topics using class-based TF-IDF, but we ultimately removed the topic-feature teacher due to leakage, using this line of work mainly to analyze "context features" and their failure modes.

3. Methodology

3.1 Task and Dataset

We participated in the POLAR SemEval-2026 Task 9 for Subtask 1 (Polarization Detection), a binary classification task, where given a short online text we need to predict whether it contains polarized opinions (Yes) or not (No). The shared task covers 22 languages covering multiple scripts and cultural contexts.

In our milestone, we planned to rely on the official labeled training set and an unlabeled development set for evaluation. This remained true in the final pipeline; however, the released dataset expanded beyond the initial version described in the milestone (which mentioned 13 languages "for now").

Our final training set was the official merged training file (master_dataset.csv), containing 77,368 rows, of which 73,681 belong to the train split used for model fitting/tuning (the rest are non-train rows in the official release). We did not use any external datasets beyond the official release.

3.2 Chosen Papers

In our methodology, we directly implemented ideas from some of the papers mentioned in Section 2. Other than the POLAR shared-task 9 dataset, we followed Conneau et al., by adopting XLM-R as our backbone for the teacher and XLM-R Base as the student due to their strong cross-lingual transfer. For training, we applied teacher-student knowledge distillation as introduced by Hinton et al. (2015) and our design is inspired by the multi-teacher framework of Zhang et al. (2023), which we simplified to a single-teacher setup to avoid label leakage. Finally, insights from Nugraha et al. guided our decision not to add toxicity or metadata features that might leak polarization labels, even though we do not re-implement their multi-label architecture.

3.3 Deterministic Row Identity (reproducible preprocessing)

To make preprocessing resume-safe and merge-safe across multiple runs and machines, each row was assigned a deterministic identifier: row_id = sha256(lang + "||" + text)[:16]. This row_id was used for deterministic sharding in translation, safe re-merging of translated shards, and stable de-duplication/overlap checks across phases.

3.4 Translation Pipeline (creating text_en)

Our milestone plan used translation to connect high-resource teacher languages with the rest of the multilingual data. In the final system, translation was implemented as follows. We added an English column text_en to the master dataset. If lang is one of the selected teacher languages, we set text_en = text (copy; avoids unnecessary translation noise/cost). Otherwise, we translate text to text_en. This design keeps the teacher's input distribution consistent and enables distillation for languages the teacher was not directly trained on.

Parallel sharding strategy (5 people × 2 sessions): To reduce the total runtime, translation was distributed across five team members, each running two independent sessions. Rows were deterministically split using row_id%2 between sessions to ensure no overlap and full coverage. The translation outputs were saved as shard CSVs and then merged back into a single dataset using row_id as a key (fill text_en only if missing; deduplicate with last-write-wins).

Translation failures: After all shard merges, 5,768 rows remained without text_en, primarily due to rate limits and marker/hashtag transformations. We accepted these failures as a small fraction of the total and handled them in training (Section 3.6) to avoid silent supervision errors.

3.5 Synthetic Data Generation for Teacher Training (Phase 1)

To strengthen the teacher, we generated synthetic training examples only from the train split, focusing on rows that were already labeled as polarized. The generation setup was as follows: the seed pool is set to 73,681 train rows and the polarized seeds are at 9,040. The planned outputs per seed are 4 (2 polarized paraphrases + 2 neutral rewrites), while the produced synthetic rows are 26,121 (after 2,508 failures and removing 7 duplicates). The final teacher training file is teacher_training_data_aug_v2.csv. Lastly, the total size after augmentation merge is 103,489 rows (77,368 original + 26,121 synthetic). Each synthetic row stored augmentation metadata (augmentation type, source_row_id, seed_text, seed_lang) to support leakage-safe splitting (next section).

3.6 Teacher A Training (XLM-R Large)

Following the teacher-student plan in our milestone, we trained a Language Expert Teacher (Teacher A) using XLM-RoBERTa Large, using the strength of large multilingual transformers for cross-lingual robustness. XLM-R is well-established for cross-lingual transfer due to multilingual pretraining at scale.For the teacher language selection, Teacher A was trained on a chosen set of 7 languages:[‘eng’,’spa’,’deu’,’rus’,’tur’,’pol’,’arb’] chosen to maximize the transfer through script and typological diversity (Latin/Cyrillic/Arabic scripts, Indo-European + agglutinative Turkish).

For the teacher training pool construction, we train split the rows from the official dataset and filtered them to the 7 teacher languages. From the synthetic data, we augmented and filtered the rows to the 7 teacher languages as well. The combined pool was deduplicated by the exact (lang, text) to remove repeats.Because synthetic samples are paraphrases/rewrites of real seeds, splitting naively can leak near-duplicates across train/validation. To ensure leakage-safe internal validation (group-aware split), we formed groups: group_id = source_row_id if the row is synthetic, else group_id = id. We then performed a group-aware, label-stratified split (per language) so that any seed and its synthetic versions remain in the same split. This implements the milestone goal of avoiding misleading evaluation signals when using translation and augmentation. Moreover, Teacher A was trained for 3 epochs and checkpoints were saved (Drive paths listed in the Appendix section later).

3.7 Teacher B Attempt and Removal (Data leakage audit)

In the milestone, we proposed a second teacher (Teacher B) as a Topic Expert built using BERTopic-style topic clusters to provide contextual signals. However, after implementing and testing this idea, we identified feature leakage, where the topic/type-vector features were present only for polarized rows (non-polarized rows often had all-zero vectors), making feature presence a direct proxy for the label,

inflating validation scores unrealistically. To preserve honest generalization, we dropped Teacher B entirely and kept Teacher A only.

3.8 Distillation Targets (Teacher logits)
To train the student with distillation, we ran Teacher A inference to produce soft targets on the real (non-synthetic) train rows. Where teacher input was valid, distillation was computed only for the rows with text_en available (translated or copied), while synthetic rows were excluded from the "real logits" distillation export to avoid a "teacher-trained-on-synth supervising synth" feedback loop. We exported teacher outputs (logits and probabilities) and merged them back into the master training table (e.g., teacherA_p1).

3.9 Student Training (XLM-R Base) with Partial Distillation
Our final submission model is a single XLM-RoBERTa Base student, consistent with the milestone inference plan of removing the teachers and using a single student at test time. The student input is always the original multilingual text (not text_en). Because the official dev set is unlabeled (per milestone), we created an internal train/validation split from labeled training data using a fixed seed and label stratification.

We implemented mixed-supervision loss (key final design), where for each training example, if the teacher probability teacherA_p1 exists (text_en exists), we combine hard-label cross entropy with distillation loss (KL divergence) to align student probabilities with the teacher. If teacherA_p1 is missing (translation missing), we train using hard labels only (no distillation). This "partial distillation" strategy ensures we do not drop the 5,768 missing-translation rows, while also preventing invalid teacher supervision.

3.10 Inference and Submission Formatting
At inference time, the pipeline is simple. We run the trained student on each language file and output pred_[lang].csv with columns (id, polarization) for all 22 evaluation languages, then zip the folder for the Codabench submission. This matches the competition-compliant teacher-student deployment plan described in the milestone.

4. Results
This section reports the internal validation results used for model selection and sanity-checking, and the official Codabench development evaluation results across all 22 languages.

4.1 Student Internal Validation Performance (Model selection)
We trained the student model (XLM-R Base) with mixed supervision (hard labels & distillation when teacher targets were available) and evaluated it on an internal validation split created from the labeled training data, since the official dev set is unlabeled. Training dynamics on the internal validation split are shown in Table 1 in the appendix. The best internal validation performance was achieved at Epoch 3, with Accuracy of 0.8104, Macro Precision of 0.8098, Macro Recall of 0.8109, and Macro F1 of 0.8100. For

the Precision-Recall (Internal Val), the Precision-Recall curve is provided in the Appendix (Figure 1), with Average Precision (AP / PR-AUC) of 0.8864.

4.2 Confusion matrix (Internal validation)

On the same internal validation set, the confusion matrix shows TN = 1407 and TP = 1563, with FP = 309 (non-polarized mislabeled as polarized) and FN = 386 (polarized missed). With the positive class defined as "polarized," this gives Positive precision of 0.8349 and Positive recall of 0.8019. The corresponding confusion matrix is shown in Figure 2 (Appendix).

4.3 Official Codabench Development Evaluation (22 languages)

After selecting the student configuration using internal validation, we trained the final student model and produced predictions for all evaluation languages following the official submission format. On the Codabench dev evaluation (22 languages), the system achieved an Average Macro F1 of 0.783 and an Average Accuracy of 0.816. The platform also provides per-language metrics in the following format: Language | Accuracy | Precision | Recall | F1 Binary | F1 Macro | F1 Micro. We included the full per-language table (Figure 3) in the Appendix.

5. Discussion

5.1 Dataset Selection and its Impact on Performance

We relied on the official POLAR multilingual dataset for Subtask 1, which spans 22 languages with diverse scripts and resource levels. This choice aligns the system directly with the evaluation distribution, but it also creates practical constraints, such as the official dev set is unlabeled, requiring internal validation for model selection. Also, performance is sensitive to cross-lingual transfer because many languages have limited labeled supervision. These properties motivated our teacher-student design and our emphasis on robust training practices (leakage checks, group-safe validation).

A key dataset-driven issue we encountered was that not all preprocessing steps could be completed perfectly at scale. In particular, our translation pipeline produced an English column (text_en) to enable distillation, but 5,768 rows remained untranslated due to rate limits and strict marker-protection constraints. Instead of dropping these examples (which would reduce multilingual coverage and bias the training distribution toward easier languages), we retained them using hard-label-only training. This improved data coverage and prevented systematic under-training on segments of the multilingual distribution.

5.2 Approach Selection: Advantages and Disadvantages

Our final approach uses a single-teacher distillation pipeline: Teacher A (XLM-R Large) model, learns strong multilingual decision boundaries on a carefully selected, high-resource subset of languages, boosted with synthetic augmentation, while the Student (XLM-R Base) is trained with both hard labels and teacher soft targets, but only when teacher targets are valid. This design introduces several advantages and disadvantages.

The advantages are firstly cross-lingual transfer without requiring multilingual teacher coverage for every script, where translating non-teacher languages into English (text_en) lets the teacher supervise them even

if the teacher was not trained to directly process those scripts at full strength. Furthermore, we had efficiency at inference, where only the student model is used at test time, which is practical for deployment and submission constraints. Also, we had robustness under incomplete preprocessing, where the "partial distillation" rule (distilled only when text_en exists) prevents invalid supervision and keeps training data coverage high. Lastly, leakage-aware evaluation and modeling, where our aware internal validation and auditing of feature leakage helped ensure that internal results reflected real generalization rather than artifacts.

On the other hand, there were some disadvantages such as, Translation introducing noise and dependence on external tooling. Even when translation succeeds, it can distort subtle pragmatic cues related to polarization (e.g., sarcasm or slang), and failures can occur from rate limits or formatting constraints. In addition, because distillation supervision is heterogeneous (only available when text_en exists), the student experiences mixed supervision regimes, which can complicate optimization and make performance sensitive to 'α' (the weight between hard labels and distillation). Lastly, teacher-language restriction may miss certain phenomena, where training Teacher A on only 7 languages was intentional for quality and coverage, but it may limit exposure to patterns specific to other languages/events in the dataset.

5.3 Comparison to Existing Systems and Literature

Our methodology follows established findings that large multilingual encoders (e.g., XLM-R) provide strong cross-lingual transfer, where knowledge distillation can compress this capability into smaller models for inference. Compared to multi-teacher distillation frameworks in the literature, our initial milestone direction was also multi-teacher; however, we found in practice that adding extra feature pathways can be risky when dataset artifacts exist.

A key outcome of our development process was that the topic/type-vector pathway created misleadingly high validation performance, because topic presence was correlated with the label. In other words, the second teacher did not measure polarization, it exploited a shortcut in the data. Removing this pathway likely reduced internal scores compared to the inflated "leaky" version, but it produced a more honest model and aligned with our goal of strong multilingual generalization. In terms of performance, our internal validation Macro F1 (0.810) was higher than the official Codabench Macro F1 (0.783), which is expected due to three factors being the internal validation is drawn from the labeled training distribution, the official evaluation covers 22 languages and likely includes more challenging distributions, and the final evaluation reflects cross-lingual generalization and real-world ambiguity.

5.4 Limitations

Our approach is constrained by a translation bottleneck and occasional missing text_en, although we handle this safely by falling back to hard-label-only training, missing translations reduces the share of examples that can benefit from distillation. Internal validation also remains a proxy, because the official dev set is unlabeled, we cannot tune thresholds or hyperparameters directly on the evaluation distribution, which may leave some generalization gains unrealized. In addition, binary framing loses nuance, where polarization is contextual and a single binary label can confuse mild partisanship with intense polarization, particularly across cultures and languages. Finally, due to compute and time constraints, hyperparameter exploration was limited beyond core settings and early stopping.

5.5 Potential Improvements with More Time/Resources

With more time and compute, dependence on translation would be reduced by boosting the translation reliability (retry policies, relaxed marker constraints, higher rate limits) or by training teacher model/s to cover more scripts/languages. Also, adding language-aware calibration, since a single global threshold may be suboptimal across 22 languages (e.g., per-language calibration or temperature scaling with labeled dev or a small calibration set). Moreover, improving synthetic augmentation beyond polarized seeds by generating harder non-polarized alternative scenarios or using adversarial generation to sharpen boundaries. Furthermore, refining distillation through confidence-based distillation, multi-temperature settings, or intermediate-layer distillation, and lastly integrating leakage audits (feature-label correlation scans, permutation tests) as a standard early-stage module, motivated by our Teacher B experience.

6. Conclusion

This project addressed multilingual online polarization detection (POLAR SemEval-2026 Task 9, Subtask 1) as a binary classification problem across 22 languages. To handle the challenges of cross-lingual generalization and limited labeled development data, we built a teacher-student distillation pipeline, where a strong Teacher A (XLM-R Large) is trained on a carefully selected set of 7 high-resource languages (supported by synthetic augmentation), providing soft supervision to a compact Student (XLM-R Base) that is used alone at inference time.

During development, we explored a second teacher pathway using topic/type features (Teacher B), but removed it after identifying label leakage, prioritizing reliable generalization over inflated internal metrics. We also introduced practical safeguards for real-world preprocessing constraints, where translation failures left 5,768 examples without English text (text_en), and we handled these safely using a partial distillation rule, where we distill only when the teacher input is valid and otherwise train on hard labels. On internal validation, our student reached Macro F1 of 0.8100 with Accuracy of 0.8104 and PR-AUC (Average Precision) of 0.8864, shown by the confusion matrix and PR curve. On the official Codabench development evaluation of over 22 languages, our final submission achieved an average Macro F1 of 0.783 and an average Accuracy of 0.816, demonstrating strong multilingual performance from an efficient and novel student-only deployment.

7. Individual Contributions

Our team collaborated throughout the project with proper communication. While each member worked on a primary part, we all contributed to design discussions, validation, and troubleshooting. In addition, we gathered our available resources to run the translation stage in parallel.

Mohamad: Led the overall system architecture design (with group consensus), implemented the student training pipeline (mixed supervision + partial distillation) and the submission/inference code (per-language prediction files and packaging).
Eid: Implemented the synthetic data generation pipeline (Phase 1 augmentation), including seed selection, generation setup, logging, and data export.
Lyan: Implemented the translation pipeline, including creation of text_en, deterministic sharding logic, shard merging, and translation integrity checks.
Ata: Implemented the Teacher A training pipeline (Teacher 1), including data filtering, deduplication, leakage-safe internal validation split, training, and checkpoint organization.

Yasin: Implemented the teacher logits generation pipeline, producing soft targets for distillation and exporting/merging logits into the master training file used by the student.

8. Appendix

Figures:

Table 1: Internal Validation Performance

Epoch	Train Loss	Val Loss	Accuracy	Macro F1
1	0.4643	0.4374	0.7918	0.7910
2	0.4129	0.4219	0.7970	0.7964
3	0.3839	0.4255	0.8104	0.8100
4	0.3558	0.4507	0.8090	0.8089 *(early stopping)*

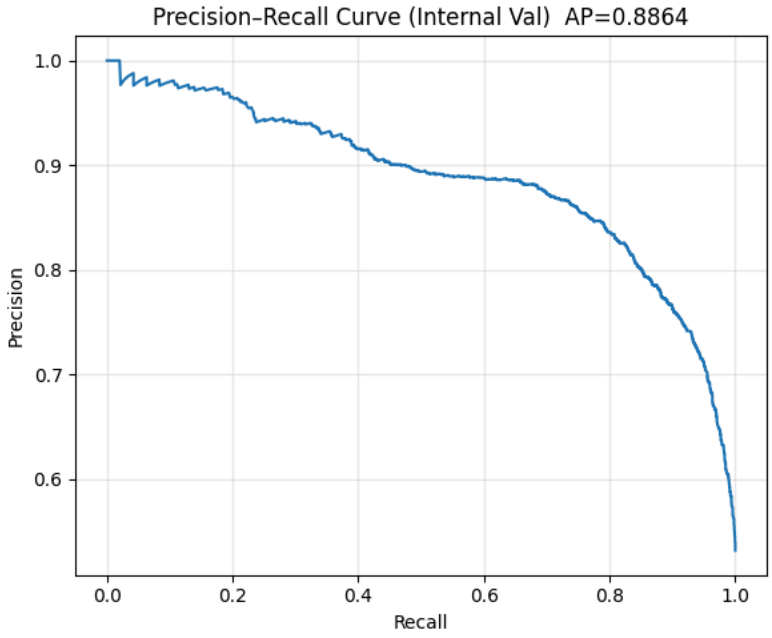

***Figure 1*: Precision-Recall Curve (Internal Val), AP = 0.8864**

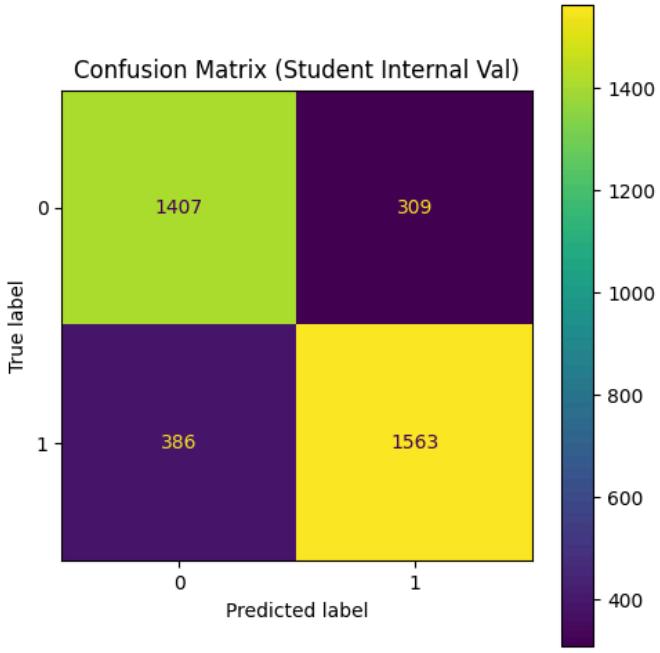

***Figure 2*: Confusion Matrix (Student Internal Val)**

Language	Accuracy	Precision	Recall	F1 Binary	F1 Macro	F1 Micro
AM Amharic	0.7952	0.8385	0.8934	0.8651	0.72	0.7952
AR Arabic	0.7929	0.7703	0.76	0.7651	0.79	0.7929
BE Bengali	0.8434	0.7895	0.8571	0.8219	0.8411	0.8434
DE German	0.717	0.6962	0.7237	0.7097	0.7168	0.717
EN English	0.825	0.8039	0.6949	0.7455	0.8061	0.825
FA Persian	0.8659	0.8926	0.9231	0.9076	0.8316	0.8659
HA Hausa	0.9396	0.7143	0.75	0.7317	0.8488	0.9396
HI Hindi	0.8759	0.906	0.9464	0.9258	0.774	0.8759
IT Italian	0.6446	0.5556	0.7246	0.6289	0.6439	0.6446

KH	Khmer	0.8976	0.9293	0.9601	0.9444	0.6453	0.8976
MY	Burmese	0.8333	0.9014	0.7901	0.8421	0.8328	0.8333
NE	Nepali	0.89	0.9348	0.8431	0.8866	0.8899	0.89
OR	Odia	0.8136	0.8095	0.4857	0.6071	0.7425	0.8136
PA	Punjabi	0.81	0.7917	0.8085	0.8	0.8095	0.81
PO	Polish	0.7731	0.717	0.76	0.7379	0.7689	0.7731
RU	Russian	0.8144	0.6909	0.7308	0.7103	0.7869	0.8144
SP	Spanish	0.7091	0.6957	0.7619	0.7273	0.7078	0.7091
SW	Swahili	0.7937	0.75	0.8793	0.8095	0.7923	0.7937
TE	Telugu	0.8898	0.8966	0.8814	0.8889	0.8898	0.8898
TU	Turkish	0.7565	0.7414	0.7679	0.7544	0.7565	0.7565
UR	Urdu	0.7797	0.8512	0.8306	0.8408	0.7415	0.7797
ZH	Chinese	0.8879	0.9126	0.8624	0.8868	0.8878	0.8879

***Figures 3-4*: Submission Results**

Pipeline Artifacts (notebooks) used in the Project:

- CS445_Data_Preprocessing&Generation_Pipline_.ipynb
- CS445_Translation_Pipeline_P{0-4}_S{0-1}.ipynb *(10 replicated notebooks for parallel sharded translation)*
- Teacher_Training_Pipeline.ipynb
- Teacher_logits_generation_Pipeline.ipynb
- Student_Training_Pipeline.ipynb
- CS445_Submission.ipynb

Final Training File Used by the Student:

- master_with_teacherA_logits.csv

9. References

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)* (pp. 8440–8451). Association for Computational Linguistics.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

He, Z., Mokhberian, N, Câmara, A., Abeliuk, A., & Lerman, K. (2021). Detecting polarized topics using partisanship-aware contextualized topic embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2021* (pp. 2102–2118). Association for Computational Linguistics.

Naseem, U., Ren, J., Anwar, S., Kohail, S., Garrido Veliz, R. A., Geislinger, R., Jabr, A., Abdulmumin, I., Qureshi, L., Borkar, A. A., Mukhtar, M. I., Ayele, A. A., Ahmad, I. S., Ali, A., Semmann, M., Muhammad, S. H., & Yimam, S. M. (2025). POLAR: A benchmark for multilingual, multicultural, and multi-event online polarization. *arXiv preprint arXiv:2505.20624*.

Nugraha, A. A., Saputri, M. E., Septiandri, A. A., & Adriani, M. (2022). A multi-labeled dataset for Indonesian discourse: Examining toxicity, polarization, and demographic information. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING 2022)* (pp. 6649–6661). International Committee on Computational Linguistics.

Zhang, R., Yang, H., & Mao, W. (2023). Cross-lingual cross-target stance detection with dual knowledge distillation framework. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)* (pp. 10804–10819). Association for Computational Linguistics.

POLAR @ SemEval-2026 Task 9. (n.d.). *Detecting multilingual, multicultural and multievent online polarization* [Task description]. Retrieved from https://polar-semeval.github.io/