1 CENG442 NLP Assignment Part-2 Report

Azerbaijani YouTube Comments + Word2Vec/FastText + GRU (Domain Analysis)

1.1 Short Summary

This study builds an end-to-end sentiment analysis pipeline by combining (i) the Part-1 labeled dataset and (ii) an unlabeled Azerbaijani comment corpus collected from YouTube. The main components of the pipeline are: YouTube video discovery and domain assignment, comment fetching with an Azerbaijani-only filtering stage, per-video Excel delivery format, Word2Vec/FastText embedding training, GRU-based sentiment classifier training, and domain-based evaluation/analysis.

Best run and performance.

- `best_run_name = w2v_tuned`
- `best_model_path = runs/w2v_tuned/model.keras`
- `test_macro_f1 = 0.5212503468102407`

Evidence: `data/analytics/final_metrics_summary.json`, `data/analytics/best_model_path.txt`.

Domain-wise saved comment counts on YouTube (after Azerbaijani-only filtering). Source: `data/analytics/comments_progress_current.csv`.

Domain	videos_with_data	total_saved_comments
Technology & Digital Services	9	10423
Finance & Business	220	12773
Social Life & Entertainment	8	12031
Retail & Lifestyle	28	10036
Public Services	11	10258

1.2 Data Sources

1.2.1 Part-1 Labeled Data

The merged Part-1 labeled master file is:

- `data/part1/processed/part1_master_labeled_final_v2.csv`

Columns: `text`, `label_id`, `source_tag`, `source_id`, `domain_5`.

Train/Validation/Test splits (with `domain_5`).

- `data/part1/processed/splits_with_domain/train.csv`
- `data/part1/processed/splits_with_domain/val.csv`
- `data/part1/processed/splits_with_domain/test.csv`

Example columns: `text`, `label_value`, `label_id`, `source_id`, `domain_5`, `__match_flag`.

1.2.2 YouTube Unlabeled Data

YouTube comments are stored in a domain-segmented structure:

- Per-video Excel: `deliverables/<Domain>/<video_id>.xlsx`
- Per-video filtered JSONL: `data/youtube/comments_filtered/<Domain>/<video_id>.jsonl`
- Raw cache (for re-filtering): `data/youtube/comments_raw/<video_id>.jsonl`

Example Excel exports.

- `deliverables/Finance & Business/0EegQ0mpm3Y.xlsx`
- `deliverables/Finance & Business/0NOWNmtd-gc.xlsx`
- `deliverables/Finance & Business/0gC1cIRlCAs.xlsx`

Evidence: `report_tables/EVIDENCE_APPENDIX.md`.

1.3 YouTube Data Collection Pipeline (Summary)

At a high level, the pipeline proceeds as follows:

1. **Video discovery (seed queries):** Fetch candidate videos using domain-specific seed queries.
2. **Expansion (expand):** Grow and enrich the candidate pool (additional queries / metadata enrichment).
3. **Merge (merge):** Consolidate candidates into a domain-wise pool.
4. **Metadata-based domain assignment:** Assign domains using title/description/tags/channel signals and other metadata.
5. **Yield scan:** Sample a subset of comments to estimate the expected Azerbaijani comment yield per video/domain.
6. **Full fetch + filter + export:** Fetch comments until the target count is reached; apply the Azerbaijani-only filter; export per-video Excel deliverables.

Relevant scripts.

- `code/02_youtube_discover_videos.py`
- `code/02b_expand_video_candidates.py`
- `code/03_domain_assign_metadata.py`
- `code/03b_select_videos_for_comments.py`
- `code/04c_full_fetch_filter_export_until_target.py`

Key selection/output artifacts.

- `data/youtube/video_candidates/videos_raw.csv`
- `data/youtube/video_candidates/videos_raw_expanded_merged.csv`
- `data/youtube/videos_labeled/videos_with_domain.csv`
- `data/youtube/selection/selected_videos_for_full_fetch.csv`
- `data/analytics/comments_progress_by_video.csv` (pipeline progress tracking)

1.3.1 Quota / Persistent Error Handling and Resume/Skip Log (Updated)

Some videos have comments disabled; these are treated as persistent errors. They are logged and skipped to prevent the pipeline from stalling.

Evidence file.

- `data/analytics/youtube_selection_failures.csv`

Schema: `domain, video_id, reason, error_message, status`.

Example records (comments disabled).

- `Social Life & Entertainment, 9v4_HkLBwwg, commentsDisabled, ..., 403`
- `Retail & Lifestyle, rHS4SN4g5Yk, commentsDisabled, ..., 403`

1.4 Azerbaijani Language Filter (AZ-only)

The filter has two layers:

1. **Character/marker scoring** based on Azerbaijani-specific characters/markers.
2. **Strict rejection rules:**
   - Cyrillic detection (Unicode ranges `U+0400--U+04FF` and `U+0500--U+052F`)
   - Turkish-marker rejection
   - Non-Latin character rejection

Default acceptance threshold: `2.5`. Source: `code/utils/az_filter.py`, `code/config.py`.

1.4.1 Language Leakage Audit

After filtering, the domain-wise audit is stored at `data/analytics/lang_audit_by_domain_after_nonlatin.csv`.

1.5 Excel Delivery Format

Each video produces one Excel file:

- Cell A1 contains the video URL.
- Rows 2+: column A = domain, column B = comment.
- No personal identifiers are stored.

Source: `code/04c_full_fetch_filter_export_until_target.py`, `deliverables/`.

Domain	total_comments	cyrillic_comments	turkish_flagged_comments
Technology & Digital Services	10423	0	0
Finance & Business	12773	0	0
Social Life & Entertainment	12031	0	0
Retail & Lifestyle	10036	0	0
Public Services	10259	0	0

Reproducibility storage (JSONL).

- `data/youtube/comments_filtered/<Domain>/<video_id>.jsonl`
- `data/youtube/comments_raw/<video_id>.jsonl`

1.6 Embedding Training (Word2Vec & FastText)

Combined corpus statistics: `part1_lines=124046`, `youtube_lines=55522`, `combined_lines=179568` (`data/youtube/corpora/combined_corpus_stats.json`). Models: `models/embeddings/w2v_combined.model`, `models/embeddings/ft_combined.model`.

Out-of-vocabulary (OOV) rates. Source: `report_tables/oov_rates.csv`.

embedding	oov_rate
word2vec	0.07292145842916858
fasttext	0.0

FastText reduces OOV via subword modeling. The `vector_size` of both models is `300`; therefore, `embedding_dim=300` is used (`models/embeddings/w2v_combined.model`; `models/embeddings/ft_combined`

1.7 GRU Modeling (4 Conditions + Best)

Architecture.

Tokenizer + Padding → Embedding (Word2Vec/FastText) → GRU → Dropout → Dense + Softmax.

Source: `code/07_train_gru_4runs.py`.

Default hyperparameters.

- `GRU_UNITS = 64`
- `DROPOUT_RATE = 0.3`
- `MAX_LEN = 80`
- `EPOCHS = 5`
- `BATCH_SIZE = 256`
- `optimizer = adam`

Source: `code/config.py`, `code/07_train_gru_4runs.py`.

Mermaid diagram.

- `mermaid_gru_diagram.md`

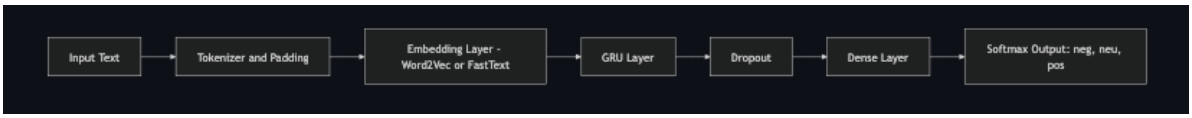

Figure 1: GRU pipeline diagram (Mermaid export).

1.8 Evaluation Results

1.8.1 Overall Test Macro-F1

The best model achieves an overall test Macro-F1 of:

`0.5212503468102407`.

Evidence: `data/analytics/final_metrics_summary.json`.

1.8.2 Experiment A: Domain-wise Macro-F1 (Best Model)

Domain-wise Macro-F1 scores for the best model are reported in: `report_tables/expA_domainwise_best.csv`.

scope	domain	macro_f1	n_samples
overall_all_rows	ALL	0.5212474067544739	4464
domain_only_official_5	Technology & Digital Services	–	0
domain_only_official_5	Finance & Business	–	0
domain_only_official_5	Social Life & Entertainment	0.5310514698548575	194
domain_only_official_5	Retail & Lifestyle	0.5801125734218889	119
domain_only_official_5	Public Services	0.5145550058247341	3798

Table 1: Experiment A: Domain-wise Macro-F1 for the best model.

Constraint note. For *Technology & Digital Services* and *Finance & Business*, `n_samples=0`; therefore, domain-wise Macro-F1 cannot be computed for these domains. This is verified by the Part-1 `domain_5` distribution and the test `domain_5` distribution: `data/analytics/part1_domain5_distributio` `data/analytics/test_domain5_distribution_v2.csv`.

1.8.3 Experiment B: LODO / Domain Shift (Best Model)

Leave-One-Domain-Out (LODO) domain shift results for the best model are reported in: `report_tables/expB_lod`

held_out_domain	macro_f1	n_test
Technology & Digital Services	–	0
Finance & Business	–	0
Social Life & Entertainment	0.19468084631263663	1993
Retail & Lifestyle	0.14934511845546838	1306
Public Services	0.1713100939487299	38173

Table 2: Experiment B: LODO domain shift (Macro-F1) for the best model.

1.9 Descriptive Analysis on YouTube (Not Ground Truth)

The YouTube corpus is unlabeled; therefore, the following results are **model predictions** and should be interpreted as descriptive summaries rather than ground-truth evaluation. Source: `report_tables/youtube_distribution_best.csv`.

domain	total	neg_count	neu_count	pos_count	neg_pct	neu_pct	po
Finance & Business	12773	4208	4922	3643	0.3294	0.3853	
Public Services	10259	3139	4387	2733	0.3060	0.4276	
Retail & Lifestyle	10036	2650	4376	3010	0.2640	0.4360	
Social Life & Entertainment	12031	1937	4648	5446	0.1610	0.3863	
Technology & Digital Services	10423	3302	3783	3338	0.3168	0.3629	

Table 3: Predicted sentiment distribution on unlabeled YouTube comments (best model).

1.10 Error Analysis

Top-error outputs.

- `report_tables/top_errors_sample_20.csv`
- `report_tables/top_errors_confident_wrong.csv`

`domain_5` **coverage summary.**

- `report_tables/errors_domain5_coverage_summary.md`

Summary (`confident_wrong`).

- `total_rows = 20`
- `domain_5_missing_or_NA_count = 5`
- `official_5_count = 15`

Domain distribution (`confident_wrong`).

- Public Services: 10
- Social Life & Entertainment: 5
- N/A: 5

Top error directions (`true_label` → `pred_label`).

- $1 \rightarrow 0$: 12
- $2 \rightarrow 0$: 7
- $0 \rightarrow 1$: 1

Likely causes of errors. Potential contributing factors include: blurred class boundaries for long, context-rich comments; domain-specific vocabulary and jargon; irony/sarcasm; and distributional effects caused by limited `domain_5` coverage in the Part-1 labeled data.
Evidence: `report_tables/top_errors_confident_wrong.csv`, `data/analytics/diagnostics/diag_test_repo`

1.11 Calibration / Additional Experiments (Optional)

Temperature scaling results. Results are summarized in:

- `data/analytics/calibration/calibration_summary.csv`

best_run_name	test_macro_f1_before	test_macro_f1_after_calibrated	test_macro_f1_label_smoothing	top_error_before	top_error_after	test_pos_pct_before
w2v_tuned	0.5212503468102407	0.4793089045998964	0.18095790199775078	1->0 (12)	2->0 (11)	0.276457399103139

Figure 2: Calibration experiment visualization (Figure 1).

top_error_after	test_pos_pct_before	test_pos_pct_after	youtube_pos_pct_avg_before	youtube_pos_pct_avg_after
2->0 (11)	0.276457399103139	0.2768817204301075	0.3249000000000001	0.28778

Figure 3: Calibration experiment visualization (Figure 2).

Interpretation. Although calibration reduces Macro-F1, it decreases the "positive spike" effect in the YouTube prediction distribution. The label-smoothing condition results in a substantial performance drop and is therefore not recommended.
Evidence: `data/analytics/calibration/temperature.json`, `runs/w2v_tuned_ls/metrics.json`.

1.12 Conclusions and Limitations

Conclusions. This project delivers an end-to-end pipeline for Azerbaijani YouTube comments, including:

- domain discovery and metadata-based domain assignment,
- an Azerbaijani-only (AZ-only) language filter,

- a per-video Excel delivery format,
- Word2Vec/FastText embedding training,
- a GRU-based sentiment classifier,
- domain-wise evaluation and domain-shift (LODO) evaluation,
- error analysis, and optional calibration experiments.

An automatically generated evidence inventory is provided in: `report_tables/EVIDENCE_APPENDIX.md`.

Primary limitation. In the Part-1 labeled dataset, there are no samples mapped to the *Technology & Digital Services* and *Finance & Business* categories under the `domain_5` mapping. Therefore, test-time domain-wise metrics cannot be computed for these two domains.
Evidence: `data/analytics/part1_domain5_distribution_v2.csv`, `data/analytics/test_domain5_distribut`

domain_5	count	pct
Public Services	3798	0.8508064516129032
N/A (unmapped)	353	0.07907706093189965
Social Life & Entertainment	194	0.04345878136200717
Retail & Lifestyle	119	0.026657706093189962

Table 4: Distribution of `domain_5` in the test split, showing zero coverage for Technology & Digital Services and Finance & Business.

Test split `domain_5` distribution (evidence snapshot).

1.13 Requirement Checklist (Q&A)

1. **Which Part-1 labeled datasets did you use and how did you load the merged version?**
   The following raw Part-1 files were used:
   - `data/part1/raw/labeled-sentiment.xlsx`
   - `data/part1/raw/train-00000-of-00001.xlsx`

   They were merged using `code/01_part1_prepare_master.py` to produce: `data/part1/processed/part1_ma`
   Train/validation/test splits are stored under: `data/part1/processed/splits_with_domain/`.

2. **How did you map Part-1 sources into the 5 domains?**
   The mapping was performed using a `source_tag` → `domain_5` table: `config/source_to_domain5_mapping.c`
   Coverage and distribution evidence: `data/analytics/part1_domain5_distribution_v2.csv`, `data/analytics/test_domain5_distribution_v2.csv`.

3. **What keywords/channels did you use per domain for YouTube discovery?**
   Domain-specific keyword sets were defined in `code/config.py`. Examples include:
   - **Technology & Digital Services:** *telefon*, *internet*, *təbiq*, *oyun*, *texnologiya*, *rəqəmsal*
   - **Finance & Business:** *bank*, *kredit*, *investisiya*, *valyuta*, *faiz*, *ipoteka*, ...

- **Social Life & Entertainment:** *musiqi*, *film*, *serial*, *şou*, *vlog*, *komediya*
- **Retail & Lifestyle:** *alış*, *qiymət*, *endirim*, *market*, *geyim*, *review*
- **Public Services:** *dövlət*, *xidmət*, *səhiyyə*, *təhsil*, *kommunal*, *vergi*

Boost terms were also used, such as *azərbaycan*, *azərbaycanca*, and *bakı/baku*.

4. **Which metadata fields did you store and where?**
   Stored metadata fields for candidate videos can be found in: `data/youtube/video_candidates/videos_raw.c`
   Columns include: `seed_domain`, `query`, `video_id`, `title`, `description`, `tags`, `channelTitle`, `publishedAt`, `categoryId`, `defaultLanguage`, `defaultAudioLanguage`, `viewCount`, `likeCount`, `commentCount`.
   Domain assignment uses these fields via: `code/03_domain_assign_metadata.py`.

5. **What Azerbaijani filter rules did you implement? What threshold and why?**
   The filter is implemented in `code/utils/az_filter.py` and includes:

   - Cyrillic rejection (Unicode ranges `--` / `--`)
   - Turkish-marker rejection
   - Non-Latin character rejection

   The scoring threshold is `2.5` (configured in `code/config.py`). The intent is to prevent Turkish leakage while enforcing sufficient density of Azerbaijani-specific characters/markers.
   Filter quality evidence: `data/analytics/lang_audit_by_domain_after_nonlatin.csv`.

6. **Confirm Excel format: A1 link + (domain, comment) rows.**
   Per-video Excel exports follow the format: cell **A1** contains the video URL; from row **2 onward**, each row is `(domain, comment)`.
   Source: `code/04c_full_fetch_filter_export_until_target.py`, `deliverables/`.
   Example files include: `deliverables/Finance \& Business/0EegQ0mpm3Y.xlsx`.

7. **How did you build the embedding matrix? What is your embedding dimension?**
   The embedding matrix is constructed by iterating over the Tokenizer `word_index` vocabulary and retrieving vectors from the trained Word2Vec/FastText models. Tokens missing from the embedding models are handled via an OOV/UNK strategy (see `code/07_train_gru_4runs.py`; `models/embeddings/w2v_combined.model`; `models/embeddings/ft_combined.model`).
   Both Word2Vec and FastText have `vector_size = 300`; therefore, `embedding_dim = 300` is used (`models/embeddings/w2v_combined.model`; `models/embeddings/ft_combined.model`).

8. **Difference between frozen vs. fine-tuned embeddings? What happened in results?**
   Frozen: embedding weights are fixed (not updated during training).
   Fine-tuned (tuned): the embedding layer is trainable and updated during classifier training.
   Test Macro-F1 results:

   - `w2v_frozen: 0.19745`
   - `w2v_tuned: 0.52125`
   - `ft_frozen: 0.27312`
   - `ft_tuned: 0.17821`

   Evidence: `runs/*/metrics.json`.

9. **Model settings: GRU units, max sequence length, dropout, optimizer, epochs.**
   Default settings: `GRU_UNITS=64`, `MAX_LEN=80`, `DROPOUT_RATE=0.3`, `optimizer=adam`, `EPOCHS=5`, `BATCH_SIZE=256`.
   Evidence: `code/config.py`, `code/07_train_gru_4runs.py`.

10. **Evaluation: overall Macro-F1, per-domain Macro-F1, domain shift results.**
    Overall Macro-F1: `0.52125` (`data/analytics/final_metrics_summary.json`).
    Per-domain Macro-F1: `report_tables/expA_domainwise_best.csv`.
    Domain shift (LODO): `report_tables/expB_lodo_best.csv`.

11. **Word2Vec vs. FastText: compare and relate to OOV evidence.**
    OOV rates: Word2Vec ≈ 0.073 and FastText $= 0.0$ (`report_tables/oov_rates.csv`).
    FastText reduces OOV via subword modeling, which provides coverage advantages for informal and noisy text (e.g., YouTube comments).

12. **Error analysis: examples and likely causes.**
    Top errors and representative misclassifications are provided in: `report_tables/top_errors_confident_wro`
    Domain_5 coverage summary is provided in: `report_tables/errors_domain5_coverage_summary.md`.
    Likely causes include irony/sarcasm, long context-heavy narratives, domain jargon, distribution imbalance, and the Part-1 `domain_5` coverage limitations.

1.14 Evidence Appendix (Short Guidance)

This project generates an appendix that automatically inventories evidence artifacts:

- Full inventory: `report_tables/EVIDENCE_APPENDIX.md`

This file lists repository paths that are confirmed to exist, including: deliverable Excel exports, filtered/raw JSONL files, pipeline scripts, embedding models, run artifacts, report tables, and diagnostics outputs.

Key additional evidence artifacts.

- **commentsDisabled / 403 log:** `data/analytics/youtube_selection_failures.csv`
- **Language leakage audit:** `data/analytics/lang_audit_by_domain_after_nonlatin.csv`
- **Final report tables:** `report_tables/expA_domainwise_best.csv`, `report_tables/expB_lodo_best.csv`, `report_tables/oov_rates.csv`, `report_tables/youtube_distribution_best.csv`
- **Top error files:** `report_tables/top_errors_confident_wrong.csv`, `report_tables/errors_domain5_co`
- **Mermaid diagram:** `mermaid_gru_diagram.md`