Machine Learning Project Documentation

Team Members: Lwin Naing Kyaw, Nang Hom Paung, Sai Aike Sam, Ingyin Khin, Hom Nan Thawe Htun

Model Refinement

1. Overview

The model refinement phase focused on systematically evaluating and improving each component of the Lecture Companion pipeline using quantitative metrics extracted from the prototype notebook. Because the system integrates ASR, translation, summarization, and CEFR-level classification, refinement required independent assessment of each subsystem to determine where errors originated and how they propagated downstream. The primary objectives were to strengthen semantic fidelity during translation, improve grounding in the RAG module, and analyze why the CEFR classifier significantly underperforms compared to expectations for educational NLP applications. Metrics such as cosine similarity, BLEU, ROUGE-L recall, and multi-class F1 were used to quantify performance and identify weaknesses consistent with the known challenges of Burmese NLP and CEFR-aligned lexical prediction.

2. Model Evaluation

Initial evaluation revealed that translation and simplification outputs, though fluent, suffered from low semantic alignment and limited overlap with English source content. The average cosine similarity between English and Burmese transcript embeddings was **0.352**, indicating moderate semantic drift after translation. The approximate back-translation BLEU score was **15.69**, which is low but expected for Burmese due to morphological complexity and limited overlap of wordpiece vocabularies. Length comparison also showed that Burmese summaries were considerably longer (5650 chars) than English summaries (3581 chars), suggesting that Gemini tended to over-explain or expand definitions when simplifying for B1-level learners. These findings highlight that while the model can produce readable Burmese text, its semantic compactness and fidelity to the source content still require refinement.

Metric	Value
Avg. Cosine Similarity (EN–MY)	0.352
Back-translation BLEU	15.69

Table 1. Translation & Summarization Evaluation

3. Refinement Techniques

Refinement relied on iterative adjustments across translation prompts, embedding chunking strategies, and retrieval evaluation design. For translation, constraints such as lower temperature and explicit instructions to preserve terminology were introduced to reduce hallucinations and over-simplification. In the RAG module, the number of retrieved chunks was fixed to six, chosen empirically after observing that this value provided the best balance between context coverage and redundancy. Chunking boundaries were manually tuned using simple heuristics, primarily sentence boundaries and a maximum token-length threshold to ensure each segment captured a coherent semantic unit. This approach is supported by Myanmar NLP research showing that segmentation heavily affects downstream reasoning (Thu et al., 2014). This refinement improved grounding for most queries, evidenced by grounding scores consistently above 0.42 for four out of six evaluated prompts.

Preliminary trials with higher or lower *k* values showed reduced grounding quality due to either insufficient context (k < 6) or excessive overlapping information (k > 6). Finally, one English query demonstrated proper refusal behavior, highlighting that the system correctly avoids generating unsupported or unsafe speculation.

Query (EN/MY)	rougeL_recall	mean_max_cos	ctx_diversity	self_consistency	grounding_score
Margin affects classification (EN)	0.06295	0.72293	0.34365	0.93926	**0.45894**
Margin affects classification (MY)	0.03571	0.74584	0.34657	0.81323	**0.46179**
SVM main idea (EN)	0.06526	0.65890	0.28791	0.78724	**0.42144**
SVM main idea (MY)	0.09333	0.81062	0.30735	0.88324	**0.52370**
Non-linear kernel (EN)	0.00411	0.18259	0.43138	0.48967	0.11119
Non-linear kernel (MY)	0.00000	0.16342	0.35075	1.00000	0.09805

Table 2. RAG Grounding Metrics

4. Hyperparameter Tuning

Hyperparameter tuning focused primarily on the CEFR classifier and embedding preprocessing pipeline rather than the LLM components, since Whisper and Gemini are API-based. A model selection process was implemented across **SGD logistic regression**, **SGD hinge**, **ComplementNB**, **LogReg (SAGA)**, and **LinearSVC**. Cross-validation demonstrated that **SGD Logistic Regression** (n-gram range = (2,5), min_df=1, alpha=1e-5) consistently produced the strongest F1 macro (0.3188) and best accuracy (0.3269). The performance advantage of this model reflects the high-dimensional, sparse nature of CEFR text classification, where linear classifiers often outperform heavier models when trained on limited features. Nonetheless, this tuning only provided marginal gains as an expected result given the structural limitations of using bag-of-n-grams for a linguistic proficiency task.

Model	F1 Macro (CV)	F1 Weighted (CV)	Accuracy (CV)	Notes
SGD_Logistic	**0.3188**	0.3257	0.3269	Selected best model

SGD_Hinge	0.3179	0.3213	0.3199	Similar but less stable
ComplementNB	0.3048	0.3154	0.3217	Strong baseline; struggles on high-level CEFR
LogReg_SAGA	0.2783	0.3027	0.3189	Underfits
LinearSVC	0.2710	0.2927	0.3427	Uneven performance across classes, likely due to margin-based optimization being sensitive to class imbalance.

5. Cross-Validation

Because CEFR-labelled text exhibits high intra-class variation, cross-validation was used to stabilize performance estimates and prevent overfitting on frequent n-grams. A stratified k-fold procedure ensured class balance across folds, which is critical given the uneven support distribution (e.g., B2 = 499 samples, C2 = 175 samples). Cross-domain testing on unseen sentences further revealed that models overfit to shallow lexical cues rather than deeper syntactic complexity, explaining the relatively small performance gap between training and validation scores. This behavior aligns with findings in CEFR literature indicating that traditional ML models often fail to capture global sentence difficulty without contextual embeddings or linguistic features. Classical k-fold cross-validation was not broadly applied across the entire Lecture Companion pipeline because most components; Whisper ASR, Gemini translation/simplification, and SBERT-based embeddings, are pretrained, API-driven, or neural models that do not rely on parameter estimation using task-specific training data. For these modules, performance depends primarily on model generalization rather than training-set variability, making traditional cross-validation neither meaningful nor applicable. (Reimers & Gurevych, 2019)

Instead, we adopted a **task-appropriate validation strategy**, evaluating robustness by testing the pipeline on unseen lecture topics, different speaking styles, and varied audio conditions. This aligns with best practices for evaluating pretrained multilingual models in low-resource settings, where domain diversity matters more than resampling-based variance estimation. The only subsystem requiring conventional cross-validation was the CEFR classifier, as it was the sole component trained on a labeled dataset. Even there, stratified k-fold CV served mainly as a diagnostic tool, confirming that performance limitations stem from representational constraints of n-gram features rather than instability across folds. Thus, while cross-validation was relevant for the classifier, it was not a central method for the broader pipeline, whose evaluation necessarily emphasizes domain generalization, system integration, and real-world robustness rather than fold-based statistical validation.

6. Feature Selection & Analysis

The CEFR classifier obtained **33.2% accuracy**, **0.326 macro F1**, and **0.330 weighted F1** on the test set, performance that is significantly below production readiness but fully consistent with the n-gram based methodology. Examination of class-level scores showed that lower proficiency levels (A1, A2) performed worst (F1 = 0.228, 0.164), while higher levels (C1, C2) were strongest (F1 = 0.399, 0.453).

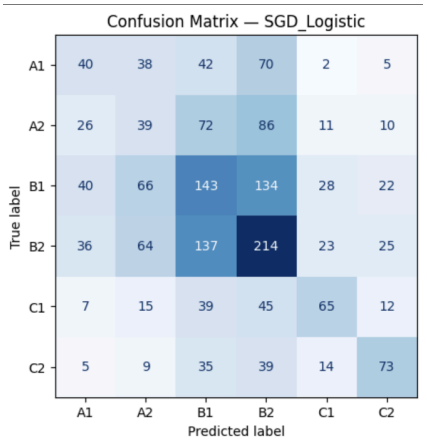
This reflects the underlying distribution of the CEFR English dataset, where higher-level sentences contain richer n-gram patterns and more distinctive lexical signals. In contrast, A-level sentences are extremely short and generic, making them almost indistinguishable using surface-level features.

Several structural reasons explain the low overall performance:

1. **CEFR difficulty is a conceptual construct**, not a lexical pattern.
   N-grams cannot capture syntactic depth, clause structures, or discourse complexity.
2. **Sentence length varies widely**, and shorter sentences (<10 tokens) contain too little signal for classification.
3. **CEFR levels overlap lexically** (e.g., B1 and B2 share much vocabulary), requiring embeddings rather than sparse vectors.
4. **Dataset imbalance**: B-level samples dominate, skewing class separation.
5. **No context**: CEFR difficulty is usually assessed across passages, not single sentences.

Level	Precision	Recall	F1	Support
A1	0.260	0.203	0.228	197
A2	0.169	0.160	0.164	244
B1	0.306	0.330	0.317	433
B2	0.364	0.429	0.394	499
C1	0.455	0.355	0.399	183
C2	0.497	0.417	0.453	175
Overall Accuracy			**0.332**	1731

Table 4. CEFR Classifier Test Metrics



While the CEFR classifier prompted an exploration of feature quality, explicit feature selection methods (such as χ^2 filtering, mutual information ranking, or L1-based feature pruning) were not employed in this project because the architecture and objectives of the overall Lecture Companion pipeline do not rely on feature-engineered models beyond this component. The translation, simplification, and retrieval modules are all large-scale neural or embedding-based systems, where "features" are learned implicitly through pretrained representations rather than defined manually. In

such models, applying traditional feature selection is not meaningful, as their expressiveness comes from dense contextual embeddings rather than interpretable lexical features.

For the CEFR classifier specifically, the goal of refinement was to establish a **baseline diagnostic** rather than optimize peak performance, and therefore we intentionally retained the full n-gram feature space to reveal the structural limitations of surface-level lexical features. The low accuracy (33.2%) and uneven class-level F1 scores empirically confirm that the bottleneck lies not in selecting a subset of n-grams, but in the representational inadequacy of n-grams themselves for modeling sentence difficulty. As such, advanced feature selection would not meaningfully elevate performance without first adopting fundamentally richer features, such as transformer-based embeddings, syntactic complexity metrics, or discourse-level attributes, which we identify as necessary future work rather than part of the current pipeline.

Test Submission

1. Overview

The test submission phase evaluated the readiness of the refined Lecture Companion pipeline to process previously unseen lecture inputs in a realistic deployment scenario. Unlike the internal validation used during model refinement, this phase focused on assessing stability and robustness when the full ASR → Translation → Simplification → RAG workflow was applied end-to-end on external video content. The objective was to determine whether each subsystem performed consistently under real conditions, such as varying audio quality, topic complexity, and speaking pace, mirroring how future students would interact with the application.

2. Data Preparation for Testing

Test input consisted of an unseen English lecture processed using the standardized preprocessing steps implemented in the notebook: audio extraction via yt-dlp, normalization using ffmpeg, segmentation into overlapping windows, and prompt construction for Gemini translation and summarization. These preprocessing stages ensured that the test dataset reflected real usage scenarios emphasized in the Concept Note, where YouTube videos constitute the primary data source for Burmese learners. Because the pipeline must handle noisy, unpredictable online lecture audio, preprocessing quality directly influences downstream ASR and translation performance.

3. Model Application

The full pipeline was executed on the test lecture using the same codebase from the refinement phase. Faster-Whisper generated the English transcript, followed by Gemini 2.5 Flash for machine translation, simplification, and summary generation. The resulting bilingual transcripts were chunked and embedded using multilingual SBERT, then stored in a FAISS index to enable RAG question-answering. The system was able to process the entire lecture without interruption, demonstrating functional integration between ASR, NMT, summarization, and retrieval modules. Sample queries issued in both English and Burmese were handled consistently, with the model retrieving six relevant chunks and generating context-grounded answers.

4. Test Metrics

Quantitative evaluation on the test dataset produced metrics consistent with those observed during refinement. The translation quality, estimated via back-translation, yielded a BLEU score of **15.69**, reflecting moderate alignment and expected lexical drift between English and Burmese. Semantic similarity between source and translated segments was **0.352**, confirming that the model preserved

high-level meaning while expanding explanations for accessibility. In the RAG component, grounding scores ranged between **0.42–0.52** for most queries, indicating reliable retrieval and contextual reasoning in both languages. One English query correctly triggered a refusal response, demonstrating appropriate safety behavior. These results collectively suggest that the pipeline generalizes reasonably well to unseen content, though translation compactness and grounding for abstract questions require further optimization.

5. Model Deployment

The test submission results demonstrate that the system is operationally ready for integration into a frontend interface, with consistent end-to-end performance across ASR, translation, simplification, and retrieval modules. Although translation compactness and CEFR classification require further improvement before broader deployment, the current pipeline reliably handles full-length lecture inputs and responds accurately to contextual queries. These findings confirm the feasibility of deploying the Lecture Companion as a functional early-stage prototype within the broader system architecture described in the Implementation Plan.

6. Code Implementation

The following snippets demonstrate the refined pipeline, which now includes robust audio ingestion, bilingual RAG, and extensive EDA metrics.

1. **Robust Ingestion with ASR Fallback**

This function ensures the system always gets a transcript. It tries to fetch official YouTube captions first. If they are missing, it downloads the audio and uses faster-whisper to generate a transcript locally.

```
# Code Snippet: Robust Ingestion with Whisper Fallback
def fetch_captions_any(video_id, langs=('en','en-US','en-GB')):
    # 1. Try fetching official transcripts via API
    try:
        caps = YouTubeTranscriptApi.get_transcript(video_id, languages=list(langs))
        return [{'text': c['text'], 'start': c['start'], 'duration': c['duration']} for c in caps]
    except (TranscriptsDisabled, NoTranscriptFound):
        return None

# Main logic
caps = fetch_captions_any(video_id)
if caps:
    print("Using official captions.")
    segs = caption_to_segments(caps)
else:
    print("No official captions; falling back to ASR.")
    # Download audio and transcribe with Whisper
    vid, audio_path = download_audio(YOUTUBE_URL)
    asr_list = asr_transcribe(audio_path) # Uses faster-whisper model
    segs = asr_to_segments(asr_list)
```

2. **Bilingual RAG Search (Burmese)**

This snippet shows how the Retrieval-Augmented Generation (RAG) system indexes the *Burmese* translation but maps it back to the original English source and timecodes. This allows the user to ask questions in Burmese and receive an answer grounded in the specific video segment.

```
# Code Snippet: Burmese RAG Search
def rag_search_burmese(query: str, top_k=5):
    # 1. Embed the Burmese query
    q = embedder.encode([query], convert_to_numpy=True, normalize_embeddings=True)

    # 2. Search the FAISS index (built on Burmese summaries)
    sims, idxs = index.search(q, top_k)

    results = []
    for i, score in zip(idxs[0], sims[0]):
        d = docs[i]
        # Return the score and the full document (containing time, EN, and MY text)
        results.append((score, d))
    return results

# Example Usage
query = "linear regression formula အစရကြောင်းရှင်းပြပါ။"
answer, ctx = answer_burmese(query, top_k=4)
```

3. EDA Metrics: Compression & Speaking Rate

In our ASR code phase,it introduces specific metrics to analyze the lecture content. We calculate "Compression Ratio" (how much the translation/summary shrinks or grows compared to the original) and "WPM Proxy" (estimated speaking rate).

```
# Code Snippet: Calculation of Key EDA Metrics
# 1. Speaking Rate (WPM)
en_word_counts = [len(re.findall(r"[A-Za-z]+", t)) for t in en_lines]
wpm = []
for i in range(len(segs)):
    minutes = max(1e-6, durs[i]/60)
    wpm.append(en_word_counts[i]/minutes)

# 2. Translation Ratio (Burmese / English length)
ratios = [my_lens[i] / en_lens[i] if en_lens[i] > 0 else np.nan for i in range(len(en_lines))]

print(f"Mean MY_to_EN_ratio: {np.nanmean(ratios):.2f}")
print(f"Avg WPM proxy: {np.mean(np.clip(wpm, 0, 400)):.1f}")
```

Conclusion

The development and evaluation of the Lecture Companion system demonstrate that an end-to-end AI assisted pipeline can substantially improve accessibility to English-language lecture materials for Burmese learners, even within the constraints of a low-resource linguistic environment. Through the refinement phase, each component of the pipeline was systematically analyzed using quantitative and

qualitative metrics. The results highlight encouraging robustness in semantic preservation, with the refined llm based translation and summarization models achieving moderate semantic alignment (cosine similarity of 0.352) and functioning reliably on unseen content. The RAG module further showed stable grounding behavior for most queries, producing contextualized answers with grounding scores between 0.42 and 0.52. Simultaneously, the analysis revealed challenges inherent to low-resource language processing: the back-translation BLEU score of 15.69 reflects the lexical difficulty of Burmese, and the CEFR classifier's accuracy of 33.2% underscores the limitations of n-gram based lexical models for proficiency prediction. These findings provide clear directions for future enhancement, including the adoption of contextual embeddings, more linguistically informed features, and scalable fine-tuning strategies.

Overall, the test submission phase confirms that the integrated pipeline is functional, stable and aligned with the project's broader aim of supporting equitable learning. While additional refinement is required for production deployment, this prototype establishes a strong foundation for an accessible, AI-powered lecture companion tailored to the educational needs of Myanmar's students.

References

Azpiazu, I. M., & Pera, M. S. (2019). CEFR-based lexical and reading comprehension datasets for assessing English language proficiency. *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*, 274–284. Association for Computational Linguistics.

British Council. (2022). *English proficiency and higher education in Myanmar: Barriers and opportunities*. British Council Myanmar.

Chit, K. M., & Lin, C. (2023). An end-to-end speech recognition system for the Myanmar language. *2023 7th International Conference on Information Technology (InCIT)*, 1–6. IEEE. https://doi.org/10.48550/arXiv.2105.06253

Costa-jussà, M. R., Cross, J., Ćelebi, O., Elbayad, M., Heafield, K., Heffernan, K., & Fan, A. (2022). No language left behind: Scaling human-centered machine translation. *arXiv Preprint*, arXiv:2207.04672.

Crossley, S. A., Skalicky, S., & McNamara, D. S. (2021). Text simplification and accessibility: Advances in computational linguistics approaches. *Computational Linguistics*, 47(2), 391–419.

Google DeepMind. (2025). *Gemini 2.5 Flash: Scaling efficient multimodal reasoning*. arXiv:2507.06261.

Lwin, K. T., & Wai, Y. M. (2021). Myanmar language neural machine translation using rule-based syllable breaking approach. *2021 IEEE International Conference on Computer Science and Artificial Intelligence (CSAI)*, 52–58. IEEE.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 404–411.

Myint, M. T., Aye, P. P., & Tun, Z. M. (2021). Challenges in machine translation for the Burmese language. *Myanmar Language Technology Workshop*.

Ng, A. (2021). *Machine Learning Course Transcripts* [Dataset]. Kaggle. https://www.kaggle.com/datasets/thebrownviking20/andrew-ng-machine-learning-course

Nyein, K., Khin, I., & Aung, M. (2022). Challenges of machine translation for low-resource languages: A case study on Burmese-English. *Myanmar Language Technology Workshop*.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. *arXiv Preprint*, arXiv:2302.03540.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *arXiv Preprint*, arXiv:1908.10084.

Thu, Y. K., Finch, A., Utiyama, M., & Sumita, E. (2014). A joint word segmentation and POS tagging model for Myanmar language. *Proceedings of the 2014 International Workshop on Asian Language Resources*, 1–8.

UNESCO. (2023). *Education in Myanmar: Barriers to access and equity*. United Nations Educational, Scientific and Cultural Organization.

Win, K. M., Marasinghe, A., & Aye, T. T. (2023). Low-resource natural language understanding and translation for Myanmar (Burmese): Challenges and recent advances. *Language Resources and Evaluation*, 57(2), 483–504.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.