Language Understanding $\lambda(\mu)$ Laboratory

Kickoff: R&D Project Selection & Team Formation

Meeting #2

Ye Kyaw Thu
Lab Leader, LU Lab., Myanmar

15 June 2025 (SUN)

Overview

1. Survey Results: Top R&D Topics
2. Team Assignments
3. Project Details & Guidance
4. Next Steps & Action Items
5. Open Discussion & Q&A

Survey Results: Top R&D Topics (1st Priority)

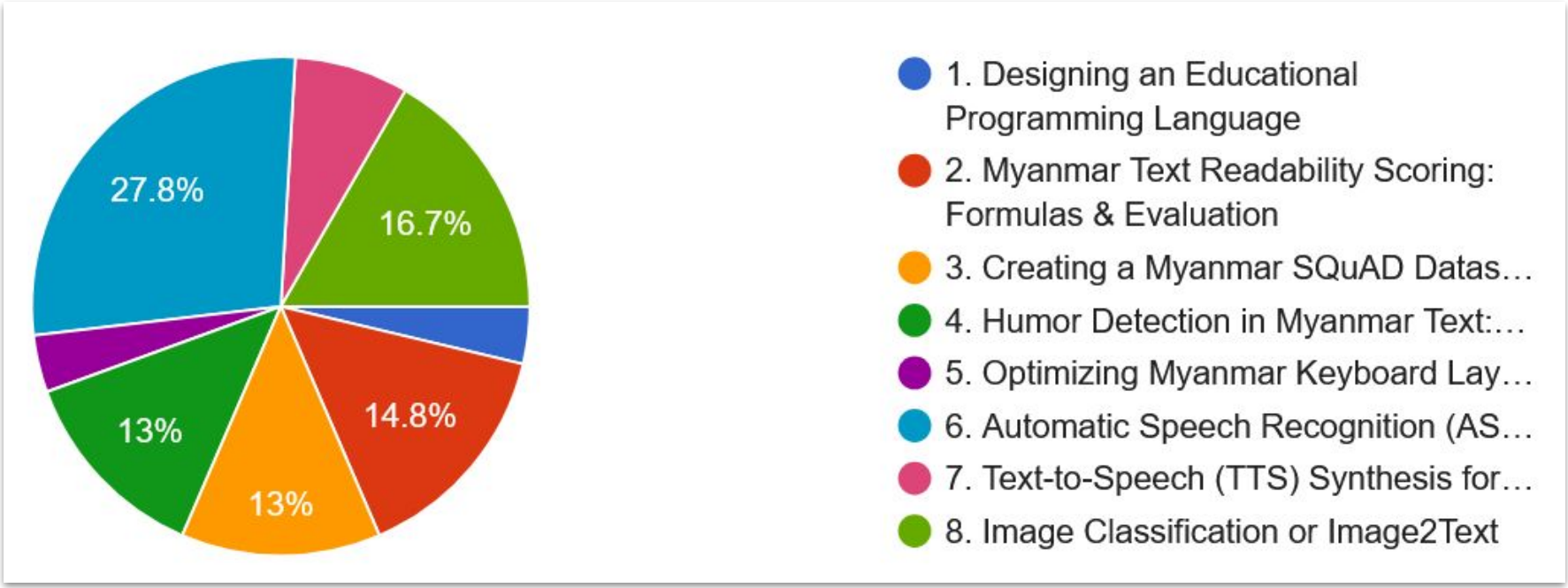


Fig. 1st priority project topics (54 respondents)

Survey Results: Top R&D Topics (1st Priority)

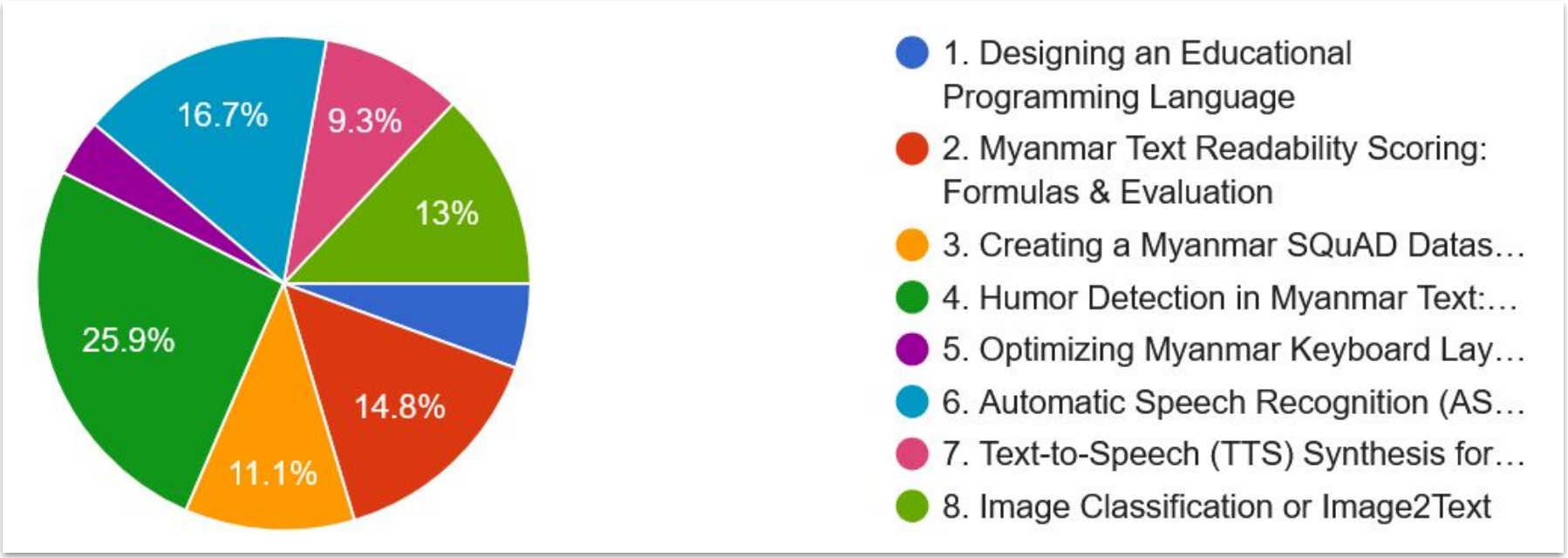


Fig. 2nd priority project topics (54 respondents)

Survey Results: Top R&D Topics (1st Priority)

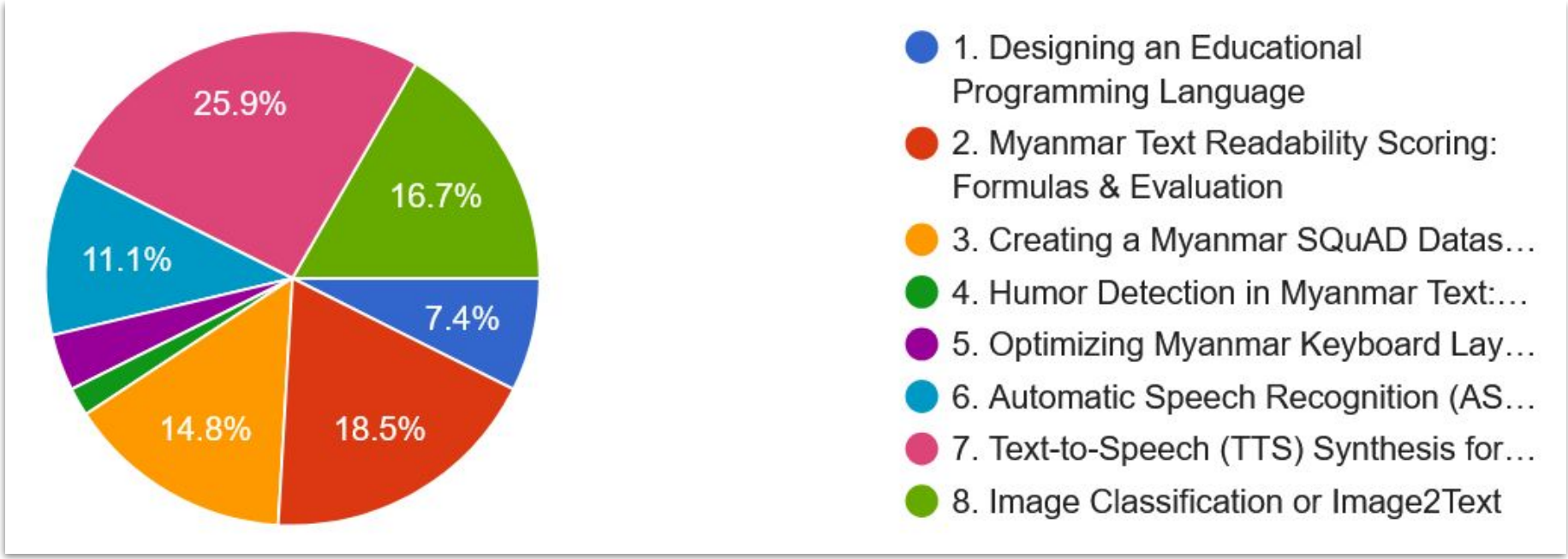


Fig. 3rd priority project topics (54 respondents)

Survey Results: Top R&D Topics

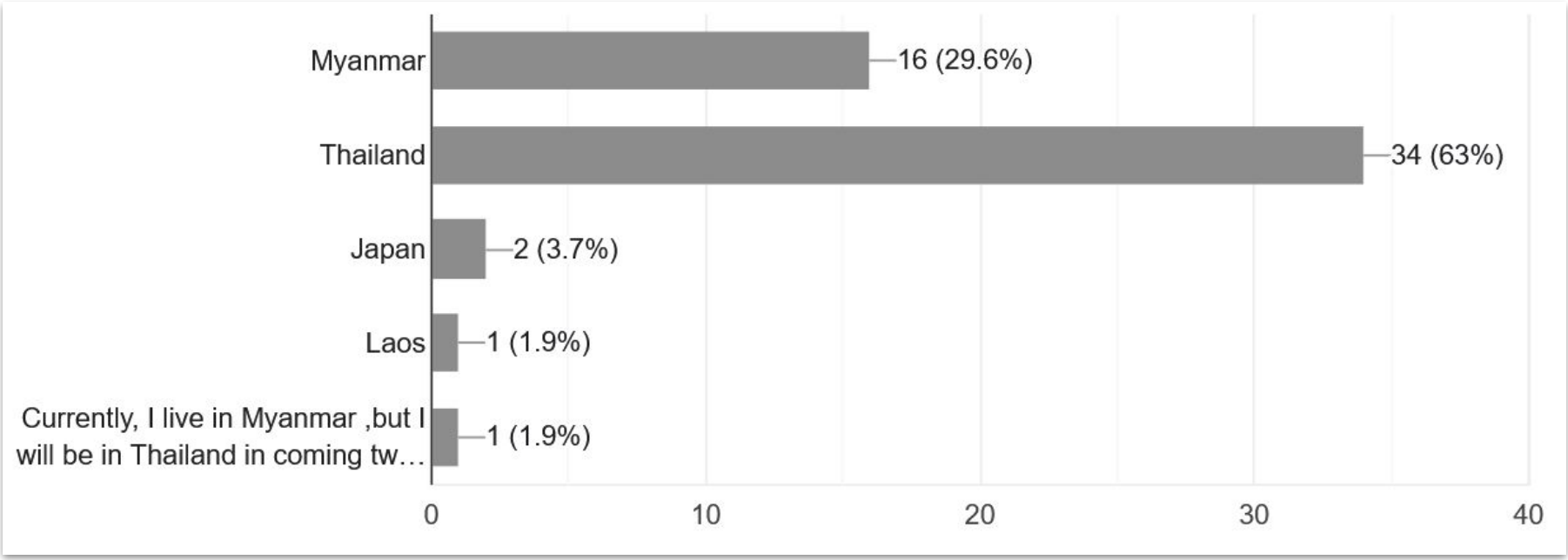


Fig.Current Country of Residence, 54 responses

Team Assignments (Humor Detection in Myanmar Text)

1. **Thura Aung** (Thailand)
2. **Ni Htwe Aung (Myanmar)**
3. Htet Htet Mon (Myanmar)
4. Phyo Thi Khaing (Thailand)
5. Zin Mar Myint (Thailand)
6. Sithu Kyaw Zinn Linn (Thailand)
7. Thant Sin Tun (Myanmar)
8. Kaung Hset Hein (Thailand)

Team Assignments (ASR)

1. **Thura Aung** (Thailand)
2. Khaing Hsu Wai (Japan)
3. Sai Wai Yan Phyo (Thailand)
4. Thiri Thaw (Thailand)
5. Moe Chan Myae Maung (Thailand)
6. Thet Htet San (Thailand)
7. Myat Oo Swe (Thailand)
8. Su Sandi Linn (Thailand)
9. Htet Arkar (Thailand)
10. Min Thiha Tun (Myanmar)
11. Saw Zi Dunn (Thailand)
12. Kyi Thant Sin (Thailand)
13. Eaint Lay Hmone (Thailand)
14. Htwe Myat Cho (Thailand)
15. Yadana Myint Hein (Myanmar)

Team Assignments (TTS)

1. Ye Bhone Lin (Myanmar)
2. Cham Myae Phyo (Thailand)
3. Thant Htut Aung (Myanmar/Thailand)
4. Nann Oak(Caesar) (Thailand)
5. Kyawt Eaindray Win (Thailand)
6. Thet Su Sann (Thailand)
7. Phoo Pwint Cho Thar (Thailand)
8. Htut Ko Ko (Thailand)

Team Assignments (SQuAD)

1. **Hay Man Htun (Lao)**
2. **Shwe Sin Moe (Myanmar)**
3. Khin Mo Mo (Thailand)
4. Chit Ko Ko Lwin (Thailand)
5. Kaung Myat Htet (Thailand)
6. Khaing Zin Theint (Thailand)
7. Thadoe Hein (Thailand)

Team Assignments (Myanmar Text Readability Scoring)

1. **Khaing Hsu Wai** (Japan)
2. **Hlaing Myat Nwe** (Japan)
3. **Thura Aung (Thailand)**
4. Seng Pan (Thailand)
5. Thiha Nyein (Thailand)
6. Kaung Khant Si Thu (Thailand)
7. Yu Myat Moe (Thailand)
8. Hsu Yee Mon (Thailand)
9. Khin Nyein Chan Thu (Thailand)

Team Assignments (Image2Text)

1. **Eaint Kay Khaing Kyaw** (Thailand)
2. Than Zaw Toe (Thailand)
3. Pyae Linn (Myanmar)
4. Zayar Htet (Myanmar)
5. Myat Thin Thin Kyi (Myanmar)
6. Thet Hmue Khin (Myanmar)
7. Lynn Myat Bhone (Myanmar)
8. Shin Thant Phyo (Myanmar)
9. Ma Hayman Soe (Myanmar)
10. Khaing Hsu Yee (Myanmar)

Project Details & Guidance (SQuAD Team)

- SQuAD dataset ကို ရုံးက ဆာဗာပေါ်ကို download လုပ်ခဲ့
- Preprocessing တချို့ လုပ်ခဲ့
- Meta ရဲ့ NLLB (No Language Left Behind) NMT နဲ့ အင်္ဂလိပ်ကနေ မြန်မာဘာသာကို စပြန်ထား
- ရှိပြီးသား NMT ကို သုံးပြီး machine translation လုပ်တာကို အတွေ့အကြုံမရှိတဲ့ သူတွေအနေနဲ့ မြင်သွားစေချင်လို့ practical work တစ်ခုအနေနဲ့ ဆရာလုပ်ပြတာ

Project Details & Guidance (SQuAD Team)

- HuggingFace ကနေ download လုပ်ခဲ့တယ်
- https://huggingface.co/datasets/rajpurkar/squad/tree/main/plain_text
- Parquet format file နှစ်ဖိုင်

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ file *.parquet
train-00000-of-00001.parquet:      Apache Parquet
validation-00000-of-00001.parquet: Apache Parquet
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

- Parquet ဖိုင်က ဒီအတိုင်း ဖွင့်ဖတ်လို့မရဘူး။ အောက်ပါလို့မျိုး သိမ်းထားတာမို့

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ head -n 3 train-00000-of-00001.parquet
PAR1□□□□□□□□mL□□□□□l□5733be284776f41900661182f□□7ff□□80j□1f□□7e□□Df84d058e614000b61bn□□B8c□B□□bdfpc□□d2:□□
8□8F□9j□aj□bj□c□a640f5□F□454J□54F□554J□U□□a70c.□□□0f64□MLJ□n83j85□□□bd:p54aF□94F□9□F□5J□UO□ac31.5ff□F□6□□
94F□□F□7□T□d320fe□F□□F□9□F□YOB□fYO□e92101□F□94F□□F□9PF□9fd3.604□TF□n8uJ8□FTOfb.104uJ□5□J□5PJ□□J□u□b1da66□
J□□F8u,J□uJ□5l□b2fe68UhF□994B89□F□9PF□U□b3d66a□8F□9F□9F□UO□b5346pduJ□uJ□94BTe□F□95df2□10□F□□B89F□YhF□Y699
2□□11□8F□n8YF8YF□@□b7f76□□29F□9F□9B□39F□86525c8UJ□n8u,J8UJ□U8722      5c9□F□a□F85PJ□5PF□a54□87ac.05cb□p□
F□□B89□F□9□8c20cc□8F□n8uHJ8□FT8a5cJ□94F894F□□F□9266     □d85J□ULF□□F□8ULF93160  dJ□|J□|J□5J93e6□□dc5lJ□9P
F8U□J□U□J□94c:□□□J□9B8tJ□UhF97460□e&    J□9F89F□9F□ □9816.□Oe2□F□9F□9lF□9F□8eb2□□e69F□YF□94F□94F□94□a5:}□
9□F□9□F□9□F□9b366□□&    J□□TF8nT9lc:D□e□8F□n89F8a3c6l□□f3uF□4□J□UJ□5J□5□a4c52□□f&□      J□J□□BT&□      □
552Of3UJ□uJ□"   J□XJ□□□db2(100□F□9lF□9lF□b49260c□J□5lF□dUOJ□□J□□□82a12□□cY□J□□J□□F□Y□J□□□83496YJ□□F85J□5□
83e:5UJ□UJ□UJ□UJ□9l□496D□□c5J□J□J□<J□□□53:□□&P□F□&P□J□9BT*□F□cOO6□□19□J□□J□□J□c0e6□□1d5lJ□□B8□B□□B□□□1a2□
11□J□□F8□F□□F□□□292P11b□TF□□TF□nT□F8udc:$□□11&□J□YLF8(J□(J□u□c46□□11=F□□F□□F□DJ□c742□□62J□n8F8J□u□ca05. □
62&H□J□9F89F□9F□y□a:(□□249lF□9lF□94F□94B□&□□cbda6□□&□□J□□pF8□pB□&P□J□9□cb2□127"J□n8J8yFpd50.□12□F□□B□□F□□
F□      06be85543aeaaa□□□8c90MV□IV□I□6bf6b□
□T6□
TR8nTiV8i□□□e46□ □V□-4□B81□e86cf60□7n□-PR88e822T□62V□n8-P□bfJT9da7:□□94F□!□F□iR□□□V□i!□□8473.,□O□d
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

1. Apache Parquet ဖိုင်ကို normal text file အဖြစ် ဖြေပေးတဲ့ code က ဆရာ့ GitHub: Tool အောက်မှာ ရှိတယ်။
2. https://github.com/ye-kyaw-thu/tools/blob/master/python/parquet_extractor.py
3. $ python ./parquet_extractor.py --parquet_file ./train-00000-of-00001.parquet
4. $ python ./parquet_extractor.py --parquet_file ./validation-00000-of-00001.parquet
5. အထက်ပါ command နှစ်ခုနဲ့ ဖြေပြီးသွားရင် original ဖိုင်က CSV ဖိုင်မို့လို့ .csv ဖိုင် နှစ်ဖိုင်ကို output အဖြစ် ရလာလိမ့်မယ်

Project Details & Guidance (SQuAD Team)

ye@lst-gpu-server-197: ~/ye/exp/gpt-mt/nllb/data/squad

```
id,title,context,question,answers
5733be284776f41900661182,University_of_Notre_Dame,"Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend ""Venite Ad Me Omnes"". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.",To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?,"{'text': array(['Saint Bernadette Soubirous'], dtype=object), 'answer_start': array([515], dtype=int32)}"
5733be284776f4190066117f,University_of_Notre_Dame,"Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend ""Venite Ad Me Omnes"". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.",What is in front of the Notre Dame Main Building?,"{'text': array(['a copper statue of Christ'], dtype=object), 'answer_start': array([188], dtype=int32)}"
5733be284776f41900661180,University_of_Notre_Dame,"Architecturally, the school has a Catholic character.
```

Project Details & Guidance (SQuAD Team)

- ဘာသာပြန်ဖို့အတွက်က CSV ဖိုင်ထဲက field တစ်ခုချင်းစီကို ဆွဲထုတ်ယူဖို့ လိုအပ်တယ်
- Python code ရေးခဲ့တယ်
- Internship ကျောင်းသားတွေလေ့လာလို့ ရအောင် GitHub မှာ repository အသစ်တစ်ခု ဆောက်ပြီး သိမ်းပေးထားမယ်

Project Details & Guidance (SQuAD Team)

```
#!/usr/bin/env python3
import argparse
import csv
import sys
import os

def clean_text(text):
    """Remove newlines and extra spaces from text"""
    if not text:
        return text
    # Replace newlines with spaces
    text = text.replace('\r', ' ').replace('\n', ' ')
    # Collapse multiple spaces into one
    return ' '.join(text.split())
```

Project Details & Guidance (SQuAD Team)

```
def extract_field(input_file, column, output_file=None):
    try:
        with open(input_file, 'r', encoding='utf-8') as csvfile:
            # Use csv reader to properly handle quoted fields with commas
            reader = csv.DictReader(csvfile)

            if column not in reader.fieldnames:
                print(f"Error: Column '{column}' not found in CSV file. Available columns: {', '.join(rea
der.fieldnames)}")
                sys.exit(1)

            data = []
            for row in reader:
                field_value = row[column]
                # Clean the text by removing internal newlines
                cleaned_value = clean_text(field_value)
                data.append(cleaned_value)

            if output_file:
                with open(output_file, 'w', encoding='utf-8') as outfile:
                    for item in data:
                        outfile.write(item + '\n')
                print(f"Successfully extracted '{column}' to {output_file}")
            else:
                for item in data:
                    print(item)

    except Exception as e:
        print(f"Error processing file: {e}")
        sys.exit(1)
```

Project Details & Guidance (SQuAD Team)

- clean_text() မပါပဲ run ရင် အောက်ပါလိုမျိုး ရလဒ်ထွက်လာလိမ့်မယ်။

```
Checking train files...

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ wc train_*.txt
  94140  714994 8706546 train_answers.txt
  88128 10491130 66304332 train_context.txt
  87599  87599 2189975 train_id.txt
  87599  881343 5307142 train_question.txt
  87599  87599 1322786 train_title.txt
 445065 12262665 83830781 total
```

original CSV ဖိုင်နဲ့ နှိုင်းယှဉ်ကြည့်တဲ့အခါမှာ...

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ wc train-00000-of-00001.csv
  94670 11915768 84323576 train-00000-of-00001.csv
```
```

Project Details & Guidance (SQuAD Team)

- SQuAD ဒေတာမှာက အောက်ပါလိုမျိုး field တစ်ခုတည်းမှာပဲ တစ်ကြောင်းထက်မကရှိတာမို့လို့

```
56beace93aeaaa14008c91e0,Super_Bowl_50,"Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the ""golden anniversary"" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as ""Super Bowl L""), so that the logo could prominently feature the Arabic numerals 50.",What venue did Super Bowl 50 take place in?,"{'text': array([""Levi's Stadium"", ""Levi's Stadium""],LF
      ""Levi's Stadium in the San Francisco Bay Area at Santa Clara""],LF
     dtype=object), 'answer_start': array([355, 355, 355], dtype=int32)}"LF
56beace93aeaaa14008c91e1,Super_Bowl_50,"Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football
```

Project Details & Guidance (SQuAD Team)

- လက်တွေ့ field တစ်ခုချင်းစီကို ဆွဲထုတ်ဖို့က python code ကို တစ်ကြိမ်ထက်မက run ရတာမို့လို့ bash shell script (extract_squad_fields.sh) ရေးပြီး run တယ်။

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ python ./extract_csv_field.py --help
usage: extract_csv_field.py [-h] --input INPUT --column COLUMN [--output OUTPUT]

Extract specific fields from SQUAD CSV files

optional arguments:
  -h, --help       show this help message and exit
  --input INPUT    Input CSV filename
  --column COLUMN  Column name to extract
  --output OUTPUT  Output filename (optional)
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

```bash
#!/bin/bash

# Function to process a single CSV file
process_file() {
    local input_file=$1
    local prefix=$2

    echo "Processing $input_file..."

    # Get all columns from the CSV file
    columns=$(python3 -c "
import csv
with open('$input_file', 'r', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    print(' '.join(reader.fieldnames))
")

    # Extract each column
    for col in $columns; do
        output_file="${prefix}_${col}.txt"
        python3 extract_csv_field.py --input "$input_file" --column "$col" --output "$output_file"
    done
}
```

Project Details & Guidance (SQuAD Team)

- ပြီးတော့မှ process_file ဆိုတဲ့ function ကို ခေါ်ပြီး run တဲ့ ပုံစံပါ။

```
# Main script
echo "Starting SQUAD dataset extraction..."

# Process training file
process_file "train-00000-of-00001.csv" "train"

# Process validation file
process_file "validation-00000-of-00001.csv" "valid"

echo "Extraction completed successfully."
```

Project Details & Guidance (SQuAD Team)

- Extracting all fields with prepared shell script and python code
- $ time ./extract_squad_fields.sh

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ wc -l train_*.txt
   87599 train_answers.txt
   87599 train_context.txt
   87599 train_id.txt
   87599 train_question.txt
   87599 train_title.txt
  437995 total
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ wc -l valid_*.txt
   10570 valid_answers.txt
   10570 valid_context.txt
   10570 valid_id.txt
   10570 valid_question.txt
   10570 valid_title.txt
   52850 total
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

- Field က စုစုပေါင်း ၅ခု ရှိတယ် (id,title,context,question,answers)
- Id, title ဆိုတဲ့ field နှစ်ခုက မြန်မာလို translation လုပ်စရာ မလိုဘူး

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ head -n 3 train_id.txt
5733be284776f41900661182
5733be284776f4190066117f
5733be284776f41900661180
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ head -n 3 train_title.txt
University_of_Notre_Dame
University_of_Notre_Dame
University_of_Notre_Dame
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ tail -n 3 valid_id.txt
5737aafd1c456719005744fd
5737aafd1c456719005744fe
5737aafd1c456719005744ff
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ tail -n 3 valid_title.txt
Force
Force
Force
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

- မြန်မာလို ဘာသာပြန်ဖို့ အဓိက လိုအပ်တာက context ရယ်၊ question ရယ်

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ head -n 3 ./train_context.txt | nl
     1  Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a gol
den statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statu
e of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Bas
ilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and re
flection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Sa
int Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects throug
h 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.
     2  Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a gol
den statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statu
e of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Bas
ilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and re
flection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Sa
int Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects throug
h 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.
     3  Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a gol
den statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statu
e of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Bas
ilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and re
flection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Sa
int Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects throug
h 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

- question field ကလည်း အရေးကြီးတယ်

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ head -n 15 ./train_question.txt | nl
     1  To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?
     2  What is in front of the Notre Dame Main Building?
     3  The Basilica of the Sacred heart at Notre Dame is beside to which structure?
     4  What is the Grotto at Notre Dame?
     5  What sits on top of the Main Building at Notre Dame?
     6  When did the Scholastic Magazine of Notre dame begin publishing?
     7  How often is Notre Dame's the Juggler published?
     8  What is the daily student paper at Notre Dame called?
     9  How many student news papers are found at Notre Dame?
    10  In what year did the student paper Common Sense begin publication at Notre Dame?
    11  Where is the headquarters of the Congregation of the Holy Cross?
    12  What is the primary seminary of the Congregation of the Holy Cross?
    13  What is the oldest structure at Notre Dame?
    14  What individuals live at Fatima House at Notre Dame?
    15  Which prize did Frederick Buechner create?
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

- answer field ရဲ့ format က အောက်ပါအတိုင်းမို့လို့၊ မြန်မာစာ အတွက်က သပ်သပ် ပြင်ဆင်ရလိမ့်မယ်

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$ head ./train_answers.txt | nl
     1  {'text': array(['Saint Bernadette Soubirous'], dtype=object), 'answer_start': array([515], dtype=
int32)}
     2  {'text': array(['a copper statue of Christ'], dtype=object), 'answer_start': array([188], dtype=i
nt32)}
     3  {'text': array(['the Main Building'], dtype=object), 'answer_start': array([279], dtype=int32)}
     4  {'text': array(['a Marian place of prayer and reflection'], dtype=object), 'answer_start': array(
[381], dtype=int32)}
     5  {'text': array(['a golden statue of the Virgin Mary'], dtype=object), 'answer_start': array([92],
 dtype=int32)}
     6  {'text': array(['September 1876'], dtype=object), 'answer_start': array([248], dtype=int32)}
     7  {'text': array(['twice'], dtype=object), 'answer_start': array([441], dtype=int32)}
     8  {'text': array(['The Observer'], dtype=object), 'answer_start': array([598], dtype=int32)}
     9  {'text': array(['three'], dtype=object), 'answer_start': array([126], dtype=int32)}
    10  {'text': array(['1987'], dtype=object), 'answer_start': array([908], dtype=int32)}
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/data/squad$
```

Project Details & Guidance (SQuAD Team)

arXiv > cs > arXiv:2207.04672

Help | Ad

Computer Science > Computation and Language

[Submitted on 11 Jul 2022 (v1), last revised 25 Aug 2022 (this version, v3)]

No Language Left Behind: Scaling Human-Centered Machine Translation

C[x] CODE | Notebook

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Jeff Wang (NLLB Team)

Ask the author(s) a question! :) | Ask

powered by CatalyzeX

Driven by the goal of eradicating language barriers on a global scale, machine translation has solidified itself as a key focus of artificial intelligence research today. However, such efforts have coalesced around a small subset of languages, leaving behind the vast majority of mostly low-resource languages. What does it take to break the 200 language barrier while ensuring safe, high quality results, all while keeping ethical considerations in mind? In No Language Left Behind, we took on this challenge by first contextualizing the need for low-resource language translation support through exploratory interviews with native speakers. Then, we created datasets and models aimed at narrowing the performance gap between low and high-resource languages. More specifically, we developed a conditional compute model based on Sparsely Gated Mixture of Experts that is trained on data obtained with novel and effective data mining techniques tailored for low-resource languages. We propose multiple architectural and training improvements to counteract overfitting while training on thousands of tasks. Critically, we evaluated the performance of over 40,000 different translation directions using a human-translated benchmark, Flores-200, and combined human evaluation with a novel toxicity benchmark covering all languages in Flores-200 to assess translation safety. Our model achieves an improvement of 44% BLEU relative to the previous state-of-the-art, laying important groundwork towards realizing a universal translation system. Finally, we open source all contributions described in this work, accessible at this https URL.

- a conditional compute model based on Sparsely Gated Mixture of Experts

Project Details & Guidance (SQuAD Team)

- NLLB အကြောင်း စိတ်ဝင်စားရင် စာတမ်းကို ဖတ်ပါ။ စာမျက်နှာ ၁၉၂ မျက်နှာ ရှိပါတယ်။
- GitHub Link: https://github.com/facebookresearch/fairseq/tree/nllb
- ဆရာ ကိုယ်တိုင် NLLB ကို သုံးပြီး translation experiment တချို့ စမ်းလုပ်ထားတာကို log မှတ်ထားတာပါတယ်။ အဲဒီ log ဖိုင်ကို လေ့လာရင်လည်း အဆင့်ဆင့် လက်တွေ့ လုပ်ဆောင်ရတဲ့ အပိုင်းကို အကြမ်း နားလည်သွားပါလိမ့်မယ်။
- https://github.com/ye-kyaw-thu/error-overflow/blob/master/nllb-exp.md

Project Details & Guidance (SQuAD Team)

- NLLB translation လုပ်ပေးတဲ့ shell script က ဆာဗာပေါ်မှာ ရှိပြီးသား
- squad_train2my.sh, squad_valid2my.sh နှစ်ဖိုင်ကို ပြင်ဆင်ခဲ့

```
#!/bin/bash

# Base directory for input files
INPUT_DIR="/home/ye/ye/exp/gpt-mt/nllb/data/squad/train/"

# Directory for output files
OUTPUT_DIR="/home/ye/ye/exp/gpt-mt/nllb/squad-my/"

# Create the output directory if it doesn't exist
mkdir -p "$OUTPUT_DIR"

# Iterate over each .src file in the input directory
for FILE in "$INPUT_DIR"/*.txt; do
    # Extract the base filename without the extension
    BASENAME=$(basename "$FILE" .txt)

    # Define the output file name
    OUTPUT_FILE="$OUTPUT_DIR/$BASENAME.my"

    # Print the command being executed (for debugging)
    echo "Running nllb-translate.sh for $FILE"

    # Run the translation command
    time ./nllb-translate.sh --input "$FILE" --source eng_Latn --target mya_Mymr --output "$OUTPUT_FILE"
done
```

Project Details & Guidance (SQuAD Team)

- translation process က အနည်းဆုံး တပတ်လောက် ကြာနိုင်တာမို့လို့ server ကို ချိတ်ထားတဲ့ terminal ရဲ့ connection က ပြတ်သွားရင် program က ရပ်သွားနိုင်တာမို့လို့ screen command ကို သုံးကြရအောင်။
- screen command ကို သုံးတတ်မှ ဖြစ်မယ်။
- $screen -S squad (screen တစ်ခု ဖန်တီးချင်ရင်)
- $screen -ls (လက်ရှိ run ထားတဲ့ sceen တွေကို ရှာကြည့်ချင်ရင်)
- -D က run နေတဲ့ screen ကို detached လုပ်ဖို့ -R က reattached လုပ်ဖို့

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb$ screen -ls
There are screens on:
        1955959.squad_valid     (06/12/2025 03:04:55 PM)        (Detached)
        1954893.squad   (06/12/2025 02:41:48 PM)        (Detached)
2 Sockets in /run/screen/S-ye.
```

Project Details & Guidance (SQuAD Team)

- ဒီ ဆလိုက်ကို ပြင်နေတဲ့အချိန်ထိ ဆာဗာပေါ်မှာ ဘာသာပြန်ပြီးတဲ့ စာကြောင်းအရေအတွက် က အောက်ပါအတိုင်း

```
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/squad-my$ ls
train_context.my  valid_context.my
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/squad-my$ date
Fri Jun 13 03:31:21 PM +07 2025
(base) ye@lst-gpu-server-197:~/ye/exp/gpt-mt/nllb/squad-my$ wc {train,valid}_context.my
    5347   882407 12800456 train_context.my
    5457   875469 12728282 valid_context.my
   10804  1757876 25528738 total
```

Project Details & Guidance (SQuAD Team)

- Machine translation ကို တချို့က အဆင်ပြေသွားပြီလို့ ထင်နေကြပေမဲ့ လက်တွေ့မှာ ရှင်းရတဲ့ ပြဿနာတွေ အများကြီးပါ။ ဘာသာပြန် အမှားတွေ အများကြီးပါ
- တခါတလေ NLLB က ဘာသာပြန်မပေးနိုင်တဲ့ စာကြောင်းတွေလည်း ရှိတာမို့လို့ trace လိုက်လို့ ရအောင် ဆရာ့ bash shell script မှာ parallel output (input ရော translated output ရော) အနေနဲ့ ဖိုင်မှာ သိမ်းအောင် ပြင်ထားပါတယ်။
- လက်ရှိ အချိန်ထိ ဘာသာပြန်ပြီးသလောက် စာကြောင်းတွေကို ဝင်လေ့လာကြည့်ကြရအောင်

Project Details & Guidance (SQuAD Team)

19 The College of Engineering was established in 1920, however, early courses in civil and mechanical engineering were a part of the College of Science since the 1870s. Today the college, housed in the Fitzpatrick, Cushing, and Stinson-Remick Halls of Engineering, includes five departments of study – aerospace and mechanical engineering, chemical and biomolecular engineering, civil engineering and geological sciences, computer science and engineering, and electrical engineering – with eight B.S. degrees offered. Additionally, the college offers five-year dual degree programs with the Colleges of Arts and Letters and of Business awarding additional B.A. and Master of Business Administration (MBA) degrees, respectively. ⟶ စက်မှုတက္ကသိုလ်သည် ၁၉၂၀ ခုနှစ်တွင် တည်ထောင်ခဲ့သော်လည်း ၁၈၇၀ ခုနှစ်မှစ၍ အရပ်ဘက်နှင့် စက်မှုအင်ဂျင်နီယာပညာရပ်များတွင် အစောပိုင်းသင်တန်းများသည် သိပ္ပံတက္ကစီ၏ အစိတ်အပိုင်းတစ်ခုဖြစ်သည်။ ယနေ့တွင် Fitzpatrick, Cushing နှင့် Stinson-Remick Halls of Engineering တွင် တည်ရှိသည့် ကောလိပ်တွင် ပညာရေးဌာနငါးခုပါဝင်သည်။ လေကြောင်းနှင့် စက်ပစ္စည်းအင်ဂျင်နီးယာ၊ ဓာတုနှင့် ဇီဝမော်လီကျူးအင်ဂျင်ဂျင်နီယာ၊ အရပ်ဖက်နှင့် ဘူမိဗေဒသိပ္ပံ၊ ကွန်ပျူတာသိပ္ပံနှင့် အင်ဂျင်နီယရီနှင့် လျှပ်စစ်အင်ဂျင်စီယာ ၈ ခုဖြင့် BS ဘွဲ့များပေးသည်။ ထို့ပြင် ကောလိယပ်သည်အခြားအထူးတန်းများဖြစ်သော B.A နှင့် Master of Business Administration (MBA) ဘွဲ့များကိုပေးသည်။

20 The College of Engineering was established in 1920, however, early courses in civil and

- ဘာသာပြန်ပြီး ရလာတဲ့ ဖိုင်တွေထဲက train_context.my ကိုဖွင့်ပြီး လိုင်းနံပါတ် ၁၉ ရဲ့ input/output ကို လေ့လာလို့ ရအောင် တွဲပြပေးထားတာပါ။
- သိပ္ပံတက္ကသိုလ် ⇒ သိပ္ပံတက္ကစီ ဆိုတဲ့ အမှားမျိုး၊ ကောလိယပ် ဆိုတဲ့ အမှားမျိုးနဲ့ တချို့ကို အင်္ဂလိပ်စာကြောင်း အတိုင်း ယူချလာတာမျိုးတွေ တွေ့ရပါလိမ့်မယ်

Project Details & Guidance (SQuAD Team)

```
4314  The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva
      Amazónica, Amazonía or usually Amazonia; French: Forêt amazonienne; Dutch:
      Amazoneregenwoud), also known in English as Amazonia or the Amazon Jungle, is a moist
      broadleaf forest that covers most of the Amazon basin of South America. This basin
      encompasses 7,000,000 square kilometres (2,700,000 sq mi), of which 5,500,000 square
      kilometres (2,100,000 sq mi) are covered by the rainforest. This region includes territory
      belonging to nine nations. The majority of the forest is contained within Brazil, with 60%
      of the rainforest, followed by Peru with 13%, Colombia with 10%, and with minor amounts in
      Venezuela, Ecuador, Bolivia, Guyana, Suriname and French Guiana. States or departments in
      four nations contain "Amazonas" in their names. The Amazon represents over half of the
      planet's remaining rainforests, and comprises the largest and most biodiverse tract of
      tropical rainforest in the world, with an estimated 390 billion individual trees divided
      into 16,000 species.⟶အမာဇုန်သစ်တော (အမေရိကန်: Amazon Rainforest) သည် တောင်အမေရိကနိုင်ငံရှိ
      အမာဇုနိုပင်လယ်အော်ဒေသ၏ အများစုကို ဖုံးအုပ်သည့် စိုထိုင်းသော သစ်တောဖြစ်သည်။ အမာဇုနိ ပင်လယ်အိုသည် စတုရန်းကီလိုမီတာ ၇၀၀၀၀၀ (၂၇၀၀၀၀၀
      စတုရာမိုင်) ကို ဖုံးလွှမ်းထားသည်။ ယင်းအနက် ၅၅၀၀၀၀၀ (၁၀၀၀၀၀) စတုဘီမိုင်သည် အာရွတ်သစ်တောဖြင့် ဖုံးကွယ်ထားသည်။ ဤဒေသတွင် နိုင်ငံ ၉
      နိုင်ငံနှင့်စပ်လျဉ်းသော နယ်မြေများပါဝင်သည်။ သစ်တောအများစုသည် ဘရာဇီးနိုင်ငံအတွင်းရှိရာ၊ အမာရွတ်သစ်ပင် ၆၀ ရာခိုင်နှုန်း၊ ပီရူးနိုင်ငံမှ ၁၃%, ဗင်နီဇွဲလားနိုင်ငံမှ ၁၀%,
      အီကွဒေါ၊ ဘိုလီဗီးယား၊ ဂိုင်ယာ၊ ဆူရီနမ်နှင့် ပြင်သစ်ဂိုင်နားနိုင်ငံများတွင် အနည်းငယ်ပါဝင်သည်။ ကောလံဘီယာနိုင်ငံရှိ နိုင်ငံ ၄ ခုသို့မဟုတ် "အမာဇုန်" နယ်ပယ်သည်
      ၎င်းတို့အမည်များတွင် ပါဝင်သည်။ အမာဆ
4315  The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva
```

- ဘာသာပြန်ပြီး ရလာတဲ့ ဖိုင်တွေထဲက valid_context.my ထဲက ဥပမာပါ။ အရမ်းရှည်တဲ့ စာကြောင်းတွေဆိုရင် NLLB က ဖြတ်ချပစ်လိုက်တာကို မြင်ကြရပါလိမ့်မယ်

Project Details & Guidance (SQuAD Team)

- SQuAD Team က NLLB အပြင် တခြား NMT (e.g. Google-Translate) ဒါမှမဟုတ် LLM (e.g. DeepSeek, ChatGPT) ကို သုံးပြီး SQuAD dataset တစ်ခုလုံးကို ဘာသာပြန်တာကို လက်တွေ့လုပ်ကြည့်စေချင်တယ်
- ဘာသာပြန်ပြီးထွက်လာတဲ့ အမှားတွေကို manual လိုက်စစ်ပြီး ပြင်နိုင်သလောက် ပြင်တာမျိုး လုပ်ဖို့ လိုအပ်လိမ့်မယ်။ တကယ်တမ်း အသုံးဝင်တဲ့ Myanmar SQuAD dataset အနေနဲ့ အများသုံးလို့ ရအောင် ရှဲချင်တယ် ဆိုရင်
- Modeling (ဥပမာ စာပိုဒ်ကို ဖတ်ခိုင်းပြီး မေးတာကို ဘယ်လောက်ဖြေနိုင်သလဲ experiment) ပါ လုပ်ကြည့်ကြဖို့အတွက် ပြင်ဆင်ဖို့ လိုနေတာ နောက်တစ်ခုက ဘာသာပြန်ထားတဲ့ မြန်မာစာ ဒေတာကနေ မှန်တဲ့ အဖြေတွေက ဘာတွေဆိုတာကို ကြိုတင် သတ်မှတ်ရန်

Project Details & Guidance (SQuAD Team)

- answer field ရဲ့ format ကို အရင်ဆုံး နားလည်အောင်လေ့လာပြီး၊ မြန်မာလို ပြင်ရမယ်

```
87581 {'text': array(['All Nepal Football Association'], dtype=object), 'answer_start': array([160], dtyp
    e=int32)}
87582 {'text': array(['25,000'], dtype=object), 'answer_start': array([498], dtype=int32)}
87583 {'text': array(['Tripureshwor'], dtype=object), 'answer_start': array([430], dtype=int32)}
87584 {'text': array(['Chinese'], dtype=object), 'answer_start': array([628], dtype=int32)}
87585 {'text': array(['17,182'], dtype=object), 'answer_start': array([54], dtype=int32)}
87586 {'text': array(['hilly terrain'], dtype=object), 'answer_start': array([289], dtype=int32)}
87587 {'text': array(['BP'], dtype=object), 'answer_start': array([500], dtype=int32)}
87588 {'text': array(['west'], dtype=object), 'answer_start': array([457], dtype=int32)}
87589 {'text': array(['Araniko'], dtype=object), 'answer_start': array([466], dtype=int32)}
87590 {'text': array(['Tribhuvan International Airport'], dtype=object), 'answer_start': array([71], dtyp
    e=int32)}
87591 {'text': array(['6'], dtype=object), 'answer_start': array([134], dtype=int32)}
87592 {'text': array(['22'], dtype=object), 'answer_start': array([297], dtype=int32)}
87593 {'text': array(['Amsterdam'], dtype=object), 'answer_start': array([698], dtype=int32)}
87594 {'text': array(['Turkish Airlines'], dtype=object), 'answer_start': array([734], dtype=int32)}
87595 {'text': array(['Oregon'], dtype=object), 'answer_start': array([229], dtype=int32)}
87596 {'text': array(['Rangoon'], dtype=object), 'answer_start': array([414], dtype=int32)}
87597 {'text': array(['Minsk'], dtype=object), 'answer_start': array([476], dtype=int32)}
87598 {'text': array(['1975'], dtype=object), 'answer_start': array([199], dtype=int32)}
87599 {'text': array(['Kathmandu Metropolitan City'], dtype=object), 'answer_start': array([0], dtype=int
    32)}
```

Project Details & Guidance (SQuAD Team)

- SQuAD ပရောဂျက်ကို အခြေခံပြီး လက်တွေ့ လုပ်ပြထားတဲ့ preprocessing, NLLB translation အပိုင်းက တခြား ပရောဂျက်ခေါင်းစဉ်ကို လုပ်ဖို့ စဉ်းစားထားတဲ့ သူတွေအတွက်လည်း အသုံးဝင်ပါလိမ့်မယ်။
- Linux command, Python programming, shell script writing တွေကတော့ ကွန်ပျူတာကျောင်းသားတိုင်းက မသိမဖြစ်လို့ ပြောလို့ ရပါတယ်။ အထူးသဖြင့်တော့ machine learning, NLP/AI အလုပ်တွေကို လုပ်ချင်တဲ့ သူတွေအတွက်က
- Linux command အသုံးပြုပုံတွေနဲ့ တခြား programming နဲ့ ပတ်သက်တာတွေက အချိန်သပ်သပ်ကြီး ယူပြီး သင်ပေးဖို့ အစီအစဉ် မရှိပေမဲ့ ကြုံရင် ကြုံသလို အခုလို လုပ်ပြသွားပါမယ်

Project Details & Guidance (ASR/TTS Team)

- PyQt6 module ကို installation လုပ်ဖို့ လိုအပ်လိမ့်မယ်

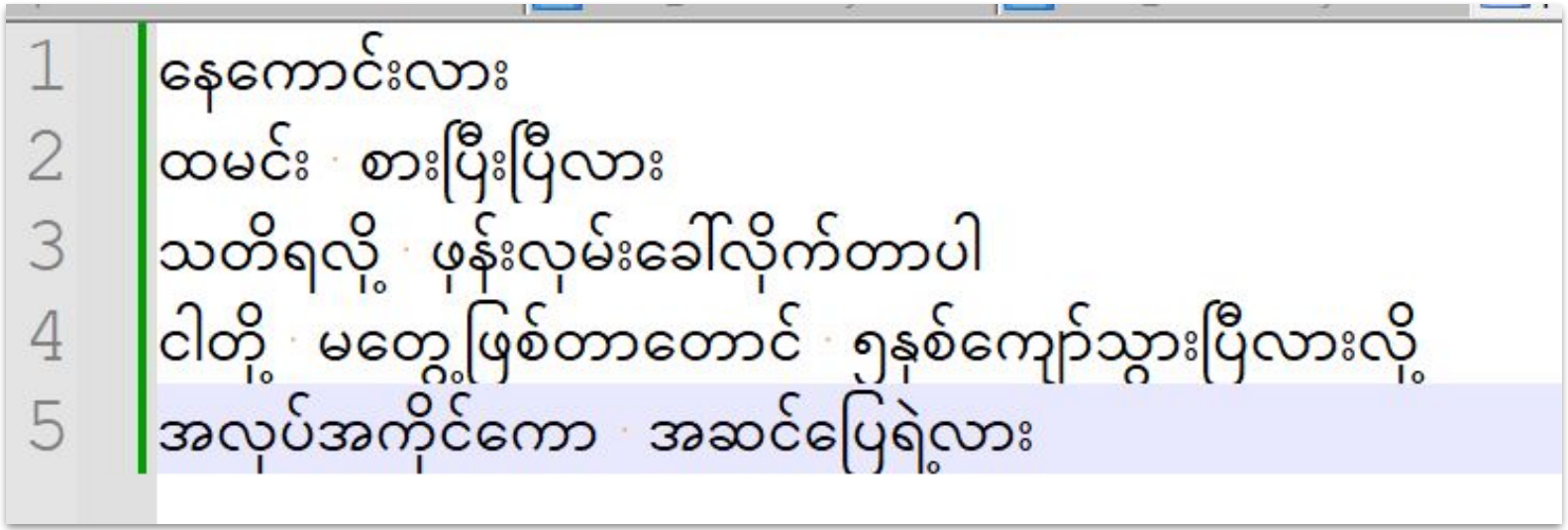
```
(base) C:\Users\801680\.spyder-py3\intern3>python ./recorder.py --help
Traceback (most recent call last):
  File "C:\Users\801680\.spyder-py3\intern3\recorder.py", line 7, in <module>
    from PyQt6.QtWidgets import (
ModuleNotFoundError: No module named 'PyQt6'

(base) C:\Users\801680\.spyder-py3\intern3>pip install PyQt6
WARNING: Ignoring invalid distribution -cipy (c:\users\801680\anaconda3\lib\site-packages)
Collecting PyQt6
  Downloading pyqt6-6.9.1-cp39-abi3-win_amd64.whl (25.7 MB)
     ---------------------------------------- 25.7/25.7 MB 880.7 kB/s eta 0:00:00
Collecting PyQt6-sip<14,>=13.8 (from PyQt6)
  Downloading pyqt6_sip-13.10.2-cp39-cp39-win_amd64.whl (53 kB)
     ---------------------------------------- 53.6/53.6 kB 398.1 kB/s eta 0:00:00
Collecting PyQt6-Qt6<6.10.0,>=6.9.0 (from PyQt6)
  Downloading pyqt6_qt6-6.9.1-py3-none-win_amd64.whl (73.8 MB)
     ---------------------------------------- 73.8/73.8 MB 812.2 kB/s eta 0:00:00
WARNING: Ignoring invalid distribution -cipy (c:\users\801680\anaconda3\lib\site-packages)
Installing collected packages: PyQt6-Qt6, PyQt6-sip, PyQt6
Successfully installed PyQt6-6.9.1 PyQt6-Qt6-6.9.1 PyQt6-sip-13.10.2

(base) C:\Users\801680\.spyder-py3\intern3>
```

Project Details & Guidance (ASR/TTS Team)

- Domain ကို သတ်မှတ်ကြရအောင်
- Recording ကို ဘယ်လိုလုပ်ကြမလဲ ဆိုတာကို တိုင်ပင်ကြရလိမ့်မယ်
- ASR အတွက်ဆိုရင် ကိုယ့် member တွေအပြင် တခြားသူတွေကိုပါ အကူအညီတောင်းနိုင်ရင် ပိုကောင်းတယ်
- Crowdsourcing လုပ်ကြမလား
- Mic ဘာညာမဝယ်နိုင်ကြရင်တောင်၊ notebook မှာ built in ပါတဲ့ mic နဲ့ သွင်းကြတာပေါ့
- PC ကိုပဲ သုံးပြီး Recording လုပ်ကြဖို့အတွက် recorder program ကိုတော့ ဆရာ ရေးပေးထားပါတယ်

Project Details & Guidance (ASR/TTS Team)

- Example prompt file, prompt.txt



Project Details & Guidance (ASR/TTS Team)

- Usage examples
- python recorder.py -p prompts.txt -d asr_recordings -m random -sr 16000
- python recorder.py -p chapter1.txt -d audiobook -m sequential -a
- python recorder.py -p phrases.txt -l 50 -m ordered -c 20
- -d option မပေးရင် recording လုပ်တဲ့ အချိန် ရက်စွဲနဲ့ folder အသစ်တစ်ခုဆောက်ပြီး သိမ်းပေးလိမ့်မယ် (e.g. rec_0518_14Jun2025)

Project Details & Guidance (ASR/TTS Team)

- ဒီ recorder ကတော့ python နဲ့ ရေးထားတာ
- Recording အတွက်က sounddevice library ကို သုံးခဲ့
- UI အတွက်က PyQt6 ကို သုံးခဲ့
- Mouse အပြင် long-term recording အတွက် အဆင်ပြေအောင် shortcut key ပါ ထည့်ပေးထား
- Recording လုပ်ခဲ့တဲ့အတွေ့အကြုံအရ အဆင်ပြေဆုံး ဖြစ်အောင် minimal လုပ်ပေးထားပါတယ်

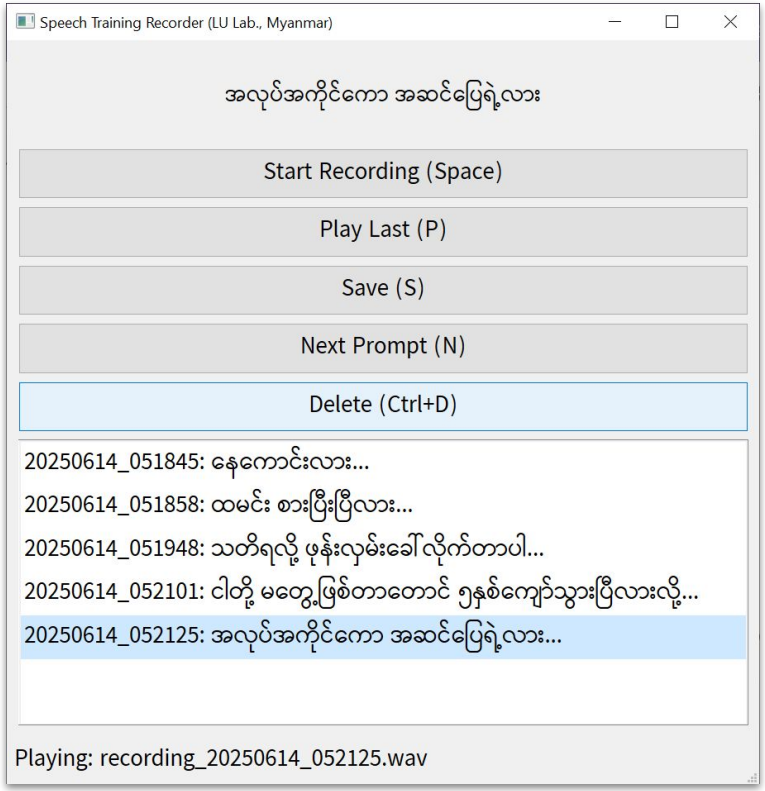


Fig. UI of Speech Training Recorder

Project Details & Guidance (ASR/TTS Team)

- Random, Ordered, Sequential အလုပ်လုပ်ပုံက

```
[Prompt File]
"နေကောင်းလား"
"ထမင်း စားပြီးပြီလား"
"သတိရလို့ ဖုန်းလုမ်းခေါ်လိုက်တာပါ"

Mode       | 1st N | 2nd N | 3rd N | 4th N
-----------|-------|-------|-------|------
random     | B     | A     | C     | (stops)
ordered    | A     | B     | C     | (stops)
sequential | A     | B     | C     | A (loops)
```

Project Details & Guidance (ASR/TTS Team)

- Example output folder

```
(base) C:\Users\801680\.spyder-py3\intern3>dir rec_0518_14Jun2025
 Volume in drive C has no label.
 Volume Serial Number is 9C54-A208

 Directory of C:\Users\801680\.spyder-py3\intern3\rec_0518_14Jun2025

06/14/2025  05:21 AM    <DIR>          .
06/14/2025  05:21 AM    <DIR>          ..
06/14/2025  05:21 AM               738 recordings.tsv
06/14/2025  05:18 AM            44,140 recording_20250614_051845.wav
06/14/2025  05:18 AM            78,252 recording_20250614_051858.wav
06/14/2025  05:19 AM            88,236 recording_20250614_051948.wav
06/14/2025  05:21 AM           119,020 recording_20250614_052101.wav
06/14/2025  05:21 AM            75,756 recording_20250614_052125.wav
               6 File(s)        406,142 bytes
               2 Dir(s)  29,852,176,384 bytes free

(base) C:\Users\801680\.spyder-py3\intern3>
```

Project Details & Guidance (ASR/TTS Team)

- နောက်ထပ် အရေးကြီးတဲ့ output ဖိုင်က recordings.tsv ဖိုင်ပါ
- Column or field ၅ခုခွဲပြီး သိမ်းပေးလိမ့်မယ်
- Delimiter က TAB
- filename, prompt, timestamp, sample_rate, bit_depth
- Example recordings.tsv file

train_context.my | valid_context.my | prompt.txt | recordings.tsv | language_teaching_workshop.md | recorder.py | recordings.tsv

```
1  filename→prompt→timestamp→sample_rate→bit_depth
2  recording_20250614_051845.wav→နေကောင်းလား→20250614_051845→16000→16
3  recording_20250614_051858.wav→ထမင်း စားပြီးပြီလား→20250614_051858→16000→16
4  recording_20250614_051948.wav→သတိရလို့ ဖုန်းလှမ်းခေါ်လိုက်တာပါ→20250614_051948→16000→16
5  recording_20250614_052101.wav→ငါတို့ မတွေ့ဖြစ်တာတောင် ၅နှစ်ကျော်သွားပြီလားလို့→20250614_052101→
   16000→16
6  recording_20250614_052125.wav→အလုပ်အကိုင်ကော အဆင်ပြေရဲ့လား→20250614_052125→16000→16
7
```

Project Details & Guidance (ASR/TTS Team)

- Linux မှာဆိုရင် sox, soxi ဆိုတဲ့ အသုံးဝင်တဲ့ command ရှိတယ်

```
(base) ye@lst-gpu-server-197:~/intern3/rec_0518_14Jun2025$ soxi ./recording_20250614_052125.wav

Input File     : './recording_20250614_052125.wav'
Channels       : 1
Sample Rate    : 16000
Precision      : 16-bit
Duration       : 00:00:02.37 = 37856 samples ~ 177.45 CDDA sectors
File Size      : 75.8k
Bit Rate       : 256k
Sample Encoding: 16-bit Signed Integer PCM

(base) ye@lst-gpu-server-197:~/intern3/rec_0518_14Jun2025$
```

Project Details & Guidance (ASR/TTS Team)

- အသုံးပြုပုံ အသေးစိတ်ကို command line help ခေါ်ကြည့်ပါ

```
(base) C:\Users\801680\.spyder-py3\intern3>python ./recorder.py --help
usage: recorder.py [-h] [-p PROMPTS_FILENAME] [-d SAVE_DIR] [-m {random,ordered,sequential}]
                   [-c PROMPTS_COUNT] [-l PROMPT_LEN_SOFT_MAX] [-a] [-sr {8000,16000,44100,48000}]
                   [-b {16,32}]

Speech Training Recorder (LU Lab., Myanmar) - Record prompted speech

optional arguments:
  -h, --help            show this help message and exit
  -p PROMPTS_FILENAME, --prompts_filename PROMPTS_FILENAME
                        text file containing prompts (one per line)
  -d SAVE_DIR, --save_dir SAVE_DIR
                        custom output directory (default: auto-generated)
  -m {random,ordered,sequential}, --prompt_selection {random,ordered,sequential}
                        prompt selection mode (default: random)
  -c PROMPTS_COUNT, --prompts_count PROMPTS_COUNT
                        max prompts to use (default: 100)
  -l PROMPT_LEN_SOFT_MAX, --prompt_len_soft_max PROMPT_LEN_SOFT_MAX
                        maximum prompt length in characters (0=no limit)
  -a, --auto_next       auto-advance to next prompt after save
  -sr {8000,16000,44100,48000}, --sample_rate {8000,16000,44100,48000}
                        sample rate in Hz (default: 16000)
  -b {16,32}, --bit_depth {16,32}
                        bit depth (16 or 32, default: 16)

Example usages:
  recorder.py -p prompts.txt -m random
  recorder.py -p script.txt -m ordered -d custom_folder
  recorder.py -p phrases.txt -m sequential -a
```

Project Details & Guidance (ASR/TTS Team)

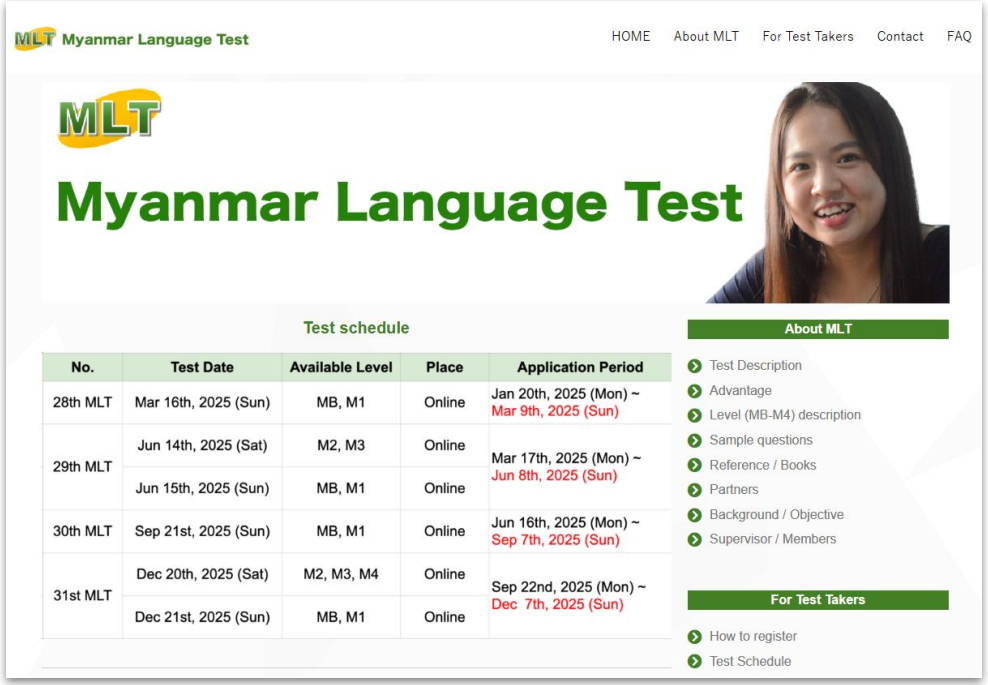
- ASR/TTS team အတွက်က ဘာရည်ရွယ်ချက်အတွက်လုပ်ကြမှာလဲ၊ ဘယ်လို အသုံးဝင်တဲ့ ပရောဂျက်ကို လုပ်ကြမှာလဲ ဆိုတာကို စဉ်းစားကြရလိမ့်မယ်
- ဒိုမိန်းကို သတ်မှတ်ပြီးတာနဲ့ text corpus ကို ပြင်ပြီး (i.e. prompt file) အသံစသွင်းနိုင်အောင် လုပ်ကြရအောင်
- အသံက ၂လ လောက်တော့ သွင်းလို့ ရလိမ့်မယ်
- Recording လည်း လုပ်ရင်းနဲ့ တခြား လေ့လာစရာရှိတာတွေကို လေ့လာသွားကြရအောင်
- ရှေ့မှာလည်း ပြောခဲ့သလိုပဲ တတ်နိုင်သ၍ speech processing R&D pipeline ထဲက အရေးကြီးတဲ့ အပိုင်းတချို့ကို သုံးလအတွင်း အတွေ့အကြုံရဖို့က အားလုံး ကြိုးစားကြမှ ဖြစ်လိမ့်မယ်

Project Details & Guidance (Readability Team)

- Readability Score အတွက် အနည်းဆုံး evaluation dataset ကို ပြင်ကြရလိမ့်မယ်
- Level တွေ ဘယ်လိုခွဲမလဲ၊ အဲဒီ ခွဲထားတဲ့ level တွေအတွက် လိုက်လျောညီထွေဖြစ်တဲ့ ဒေတာကို စဉ်းစားကြရလိမ့်မယ်
- ပထမဆုံး seminar မှာ အကြမ်းရှင်းပြခဲ့သလိုပဲ တခြား ဘာသာစကားတွေအတွက် readability score ကို ဘယ်လို သတ်မှတ်ကြသလဲ ဆိုတာကို literature review သေချာလုပ်ကြရလိမ့်မယ်။ မှတ်သားရလိမ့်မယ်
- မြန်မာစာအတွက် UFL က သတ်မှတ်ထားတဲ့ proficiency test ဆိုတာကိုရှာဖွေလို့ ရနိုင်မလား
- Myanmar Language Test ကလည်း တိုက်ရိုက် မသက်ဆိုင်ပေမဲ့ သူတို့ ဘယ်လို သတ်မှတ်ထားသလဲ လေ့လာသင့်တယ်

Project Details & Guidance (Readability Team)

- Homepage of MLT
- Link: https://www.mlt-myanmar.com/



No.	Test Date	Available Level	Place	Application Period
28th MLT	Mar 16th, 2025 (Sun)	MB, M1	Online	Jan 20th, 2025 (Mon) ~ Mar 9th, 2025 (Sun)
29th MLT	Jun 14th, 2025 (Sat)	M2, M3	Online	Mar 17th, 2025 (Mon) ~ Jun 8th, 2025 (Sun)
	Jun 15th, 2025 (Sun)	MB, M1	Online	
30th MLT	Sep 21st, 2025 (Sun)	MB, M1	Online	Jun 16th, 2025 (Mon) ~ Sep 7th, 2025 (Sun)
31st MLT	Dec 20th, 2025 (Sat)	M2, M3, M4	Online	Sep 22nd, 2025 (Mon) ~ Dec 7th, 2025 (Sun)
	Dec 21st, 2025 (Sun)	MB, M1	Online	

Project Details & Guidance (Readability Team)

Burmese alphabet and very basic vocabulary and grammar in daily life.

MB

Spoken / Written	Spoken: 100 %
Vocabulary range	200 – 300
Learning period	72 hours (3 months)

MB guideline | MB Sample Question | MB Reference / Books

Basic vocabulary and grammar to survive and communicate in daily life.

M1

Spoken / Written	Spoken: 100 %
Vocabulary range	500 – 700
Learning period	144 hours (6 months)

M1 guideline | M1 Sample Question | M1 Reference / Books

Be able to communicate in daily life and read simple paragraph.

M2

Spoken / Written	Spoken: 100 %
Vocabulary range	1200 – 1400
Learning period	288 hours (12 months)

M2 guideline | M2 Sample Question | M2 Reference / Books

Be able to communicate with native speaker and read normal paragraph.

M3

Spoken / Written	Spoken: 70 %, Written: 30 %
Vocabulary range	2500 – 3500
Learning period	432 hours (18 months)

M3 guideline | M3 Sample Question | M3 Reference / Books

Be able to communicate in Business situation and read difficult paragraph.

M4

Spoken / Written	Spoken: 50 %, Written: 50 %
Vocabulary range	6000 ++
Learning period	576 hours (24 months ++)

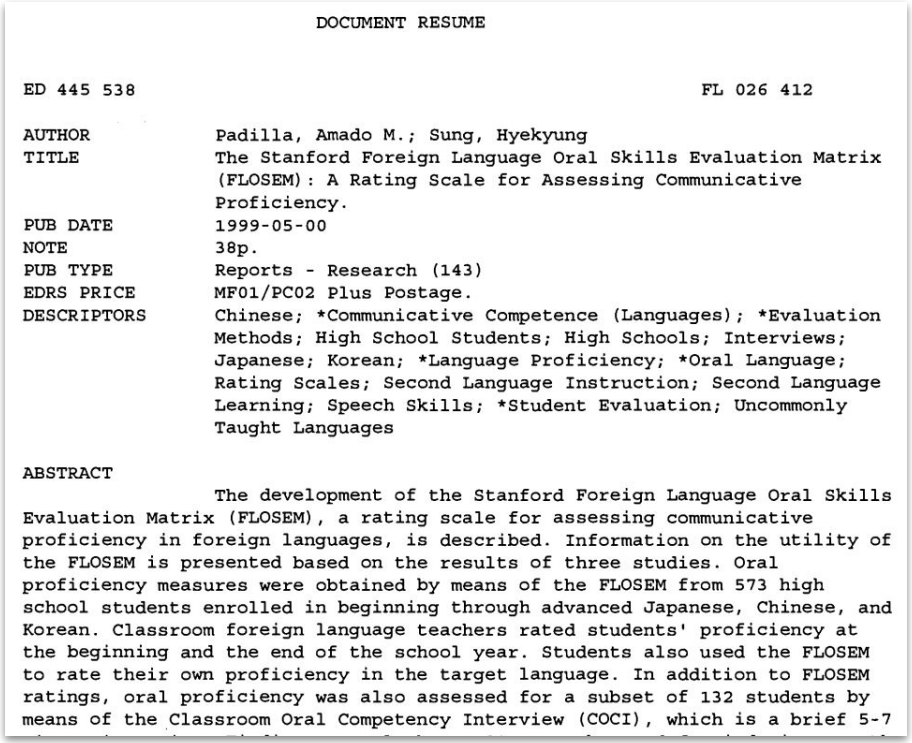
M4 guideline | M4 Sample Question | M4 Reference / Books

Project Details & Guidance (Readability Team)

- MB: https://www.mlt-myanmar.com/wp-content/uploads/2022/01/Guideline-for-the-test-taker-MB_202201.pdf
- M1: https://www.mlt-myanmar.com/wp-content/uploads/2022/01/Guideline-for-the-test-taker-M1_202201.pdf
- M2: https://www.mlt-myanmar.com/wp-content/uploads/2023/02/Guideline-for-the-test-taker-M2_202301.pdf
- M3: https://www.mlt-myanmar.com/wp-content/uploads/2022/01/Guideline-for-the-test-taker-M3_202201.pdf
- M4: https://www.mlt-myanmar.com/wp-content/uploads/2022/01/Guideline-for-the-test-taker-M4_202201.pdf

Project Details & Guidance (Readability Team)

- I read this The Stanford Foreign Language Oral Skills Evaluation Matrix (FLOSEM): A Rating Scale for Assessing Communicative
- သူက Oral
- proficiency measures ကို အဓိကထားတယ်
- ပြီးတော့ foreign language အနေနဲ့စဉ်းစားထားတယ်
- ငါတို့က text ကိုပဲ အဓိကထားရမယ်



DOCUMENT RESUME

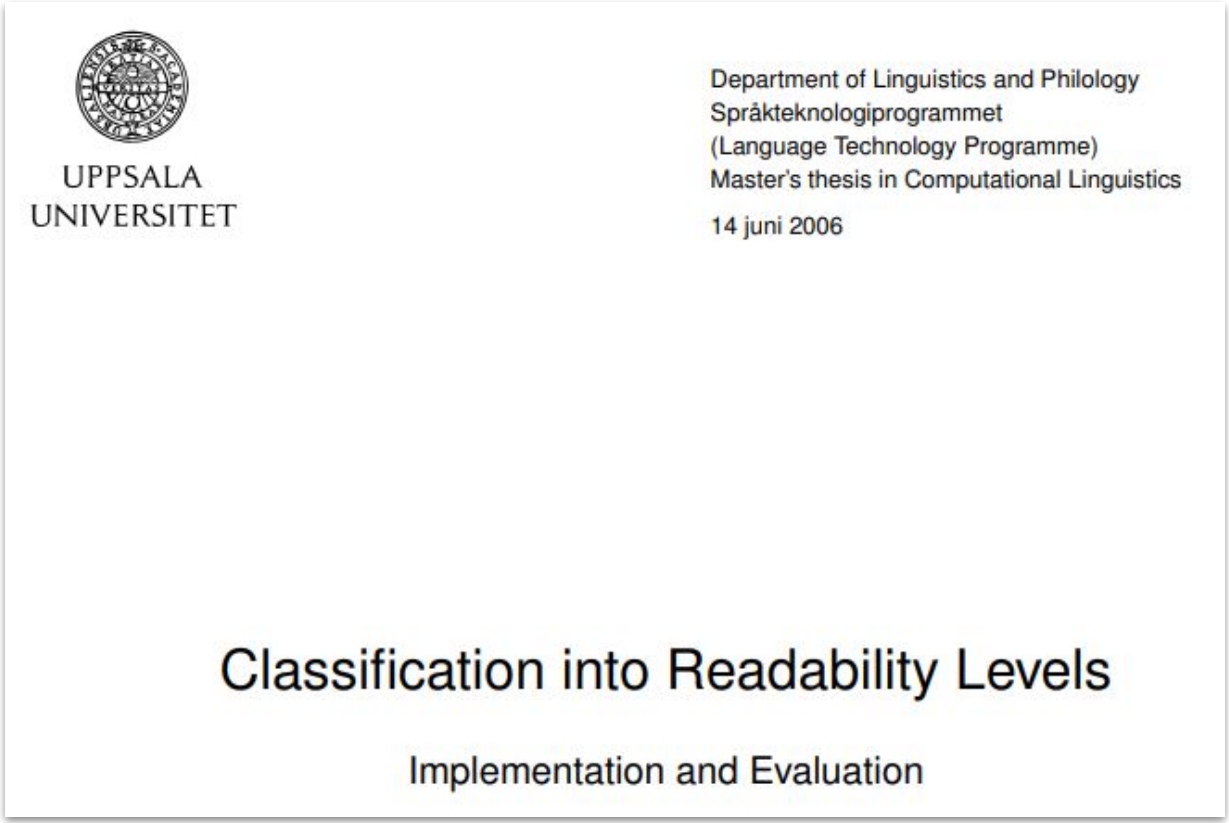
ED 445 538 FL 026 412

AUTHOR Padilla, Amado M.; Sung, Hyekyung
TITLE The Stanford Foreign Language Oral Skills Evaluation Matrix (FLOSEM): A Rating Scale for Assessing Communicative Proficiency.
PUB DATE 1999-05-00
NOTE 38p.
PUB TYPE Reports - Research (143)
EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Chinese; *Communicative Competence (Languages); *Evaluation Methods; High School Students; High Schools; Interviews; Japanese; Korean; *Language Proficiency; *Oral Language; Rating Scales; Second Language Instruction; Second Language Learning; Speech Skills; *Student Evaluation; Uncommonly Taught Languages

ABSTRACT

The development of the Stanford Foreign Language Oral Skills Evaluation Matrix (FLOSEM), a rating scale for assessing communicative proficiency in foreign languages, is described. Information on the utility of the FLOSEM is presented based on the results of three studies. Oral proficiency measures were obtained by means of the FLOSEM from 573 high school students enrolled in beginning through advanced Japanese, Chinese, and Korean. Classroom foreign language teachers rated students' proficiency at the beginning and the end of the school year. Students also used the FLOSEM to rate their own proficiency in the target language. In addition to FLOSEM ratings, oral proficiency was also assessed for a subset of 132 students by means of the Classroom Oral Competency Interview (COCI), which is a brief 5-7

Project Details & Guidance (Readability Team)



UPPSALA
UNIVERSITET

Department of Linguistics and Philology
Språkteknologiprogrammet
(Language Technology Programme)
Master's thesis in Computational Linguistics

14 juni 2006

Classification into Readability Levels

Implementation and Evaluation

- for Swedish
- By Patrik Larsson (2006)
- Supervised by Beata Megyesi
- ဒီ thesis က ဖတ်သင့်တယ်

Link: http://www.diva-portal.org/smash/get/diva2:131028/FULLTEXT01.pdf

Project Details & Guidance (Readability Team)

- စာမျက်နှာ ၁၀ မှာ

1. **Lorges Formula** (1939) revised in 1948, classifies texts into grades 3-12.

   $Grade = .07sl + .1073w_d + .1301pp + 1.6126$ where

   - sl = Average sentence length.
   - w_d = Number of different difficult words per 100 words. Difficult words are words not on the *Dale 769-word list*.
   - pp = Number of prepositional phrases per 100 words.

   Lorges formula is considered as the best among the earliest formulas. It is relatively straightforward to calculate, which made it popular.

Project Details & Guidance (Readability Team)

- စာမျက်နှာ ၁၀ မှာ

2. **Dale-Challs Formula** (1948) revised in 1995, classifies texts into grades 3-12.

$$Grade = .0596sl + .1579w_d + 3.6365$$

- sl = Average sentence length.
- w_d = The percentage of words not occurring on the *Dale list of 3000*.

In 1995, the *Dale list of 3000* was updated and the formula was changed. The reason why this formula is less known and used than, for example Fleschs Reading Ease formula is that it is more difficult to calculate, since checking of the 3000 words on the list is a time consuming task.

Project Details & Guidance (Readability Team)

- Dale-Challs 3000 words (ပို့ဝ်ဒ အစပို့ဝ်း)

a able aboard about above absent accept accident account ache aching acorn acre across act acts add address admire adventure afar →afraid after afternoon afterward afterwards again against age aged ago agree ah ahead aid aim air airfield airplane airport airship airy →alarm alike alive all alley alligator allow almost alone along aloud already also always am America American among amount an and →angel anger angry animal another answer ant any anybody anyhow anyone anything anyway anywhere apart apartment ape apiece appear apple April →apron are aren't arise arithmetic arm armful army arose around arrange arrive arrived arrow art artist as ash ashes aside ask →asleep at ate attack attend attention August aunt author auto automobile autumn avenue awake awaken away awful awfully awhile ax axebaa babe babies back background backward backwards bacon bad badge badly bag bake baker bakery baking ball balloon banana band bandage bang banjo bank banker bar barber bare barefoot barely bark barn barrel base baseball basement basket bat batch bath bathe bathing bathroom bathtub →battle battleship bay be beach bead beam bean bear beard beast beat beating beautiful beautify beauty became because become becoming bed bedbug bedroom bedspread bedtime bee beech beef beefsteak beehive been beer beet before beg began beggar begged begin beginning begun behave behind being → believe bell belong below belt bench bend beneath bent berries berry beside besides best bet better between bib bible bicycle bid big bigger bill billboard

Project Details & Guidance (Readability Team)

- Dale-Challs 3000 words (ဖိုင် အဆုံးပိုင်း)

```
understand underwear ⟶undress unfair unfinished unfold unfriendly unhappy ⟶
unhurt uniform United States unkind unknown ⟶unless unpleasant until
unwilling up upon →upper upset upside upstairs uptown upward →us use used
usefulvalentine valley valuable ⟶value vase vegetable ⟶velvet very vessel →
victory view village ⟶vine violet visit →visitor voice votewag wagon waist
wait wake waken walk wall walnut want war warm warn was wash washer washtub
wasn't waste watch watchman water watermelon waterproof wave wax →way wayside
we weak weakness weaken wealth weapon wear weary weather weave web we'd wedding
Wednesday wee weed week we'll weep weigh welcome well went were ⟶we're west
western wet we've whale what what's wheat wheel when whenever where which while
whip whipped whirl whisky whiskey whisper whistle white who who'd whole →who'll
whom who's whose why wicked wide wife wiggle wild wildcat will willing willow
win wind windy windmill window wine wing wink winner winter wipe wire →wise
wish wit witch with without woke wolf woman women won wonder wonderful won't
wood wooden woodpecker woods wool woolen word wore work worker workman world →
worm worn worry worse worst worth would wouldn't wound wove wrap wrapped wreck
wren wring write writing written wrong wrote wrungyard yarn year yell ⟶yellow
yes yesterday yet ⟶yolk yonder you you'd →you'll young youngster your ⟶
yours you're yourself yourselves ⟶youth you've
```

Project Details & Guidance (Readability Team)

3. **Flesch Reading Ease formula** (1948) revised several times, returns a score where a higher score indicates a more difficult text.

$$ReadingEase = 206.835 - 1.015sl - .846wl$$

- sl = Average number of word per sentence.
- wl = Number of syllables per 100 words.

Flesch Reading Ease formula returns a number between 0-100, where a higher score indicates that the text is harder to read. The formula is very simple to calculate, since the text passage to analyze has to be only 100 words and only two features need to be investigated. Reading Ease has become U.S governmental standard and most states in the U.S, require insurance forms to score at a certain level (around 40-50) to be valid. The formula is also used in several word processors as a service to test the documents readability. Flesch Reading Ease formula is the most used and well-known formula and it has influenced formulas for other languages because of its high correlation score and the simple calculation.

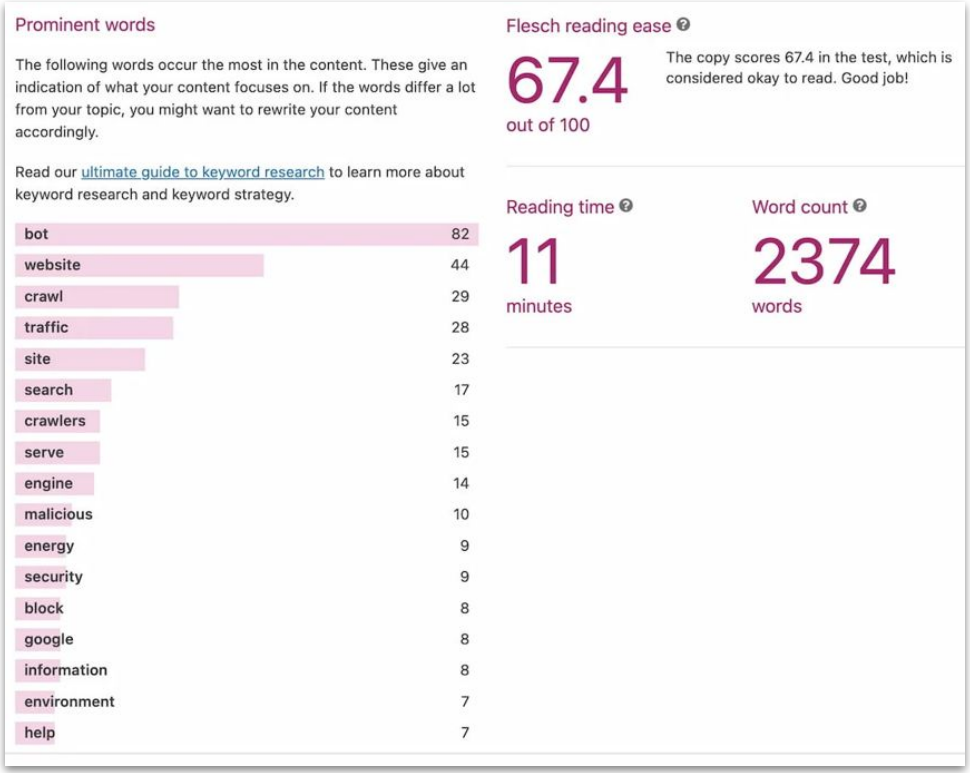
- စာမျက်နှာ ၁၀ နဲ့ ၁၁ မှာ

Project Details & Guidance (Readability Team)

- the Flesch reading ease score
- Source: https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests

Score	School level (US)	Notes
100.00–90.00	5th grade	Very easy to read. Easily understood by an average 11-year-old student.
90.0–80.0	6th grade	Easy to read. Conversational English for consumers.
80.0–70.0	7th grade	Fairly easy to read.
70.0–60.0	8th & 9th grade	Plain English. Easily understood by 13- to 15-year-old students.
60.0–50.0	10th to 12th grade	Fairly difficult to read.
50.0–30.0	College	Difficult to read.
30.0–10.0	College graduate	Very difficult to read. Best understood by university graduates.
10.0–0.0	Professional	Extremely difficult to read. Best understood by university graduates.

Project Details & Guidance (Readability Team)



- Reading time က ဘယ်လောက်ကြာသလဲဆိုတဲ့ အချက်ကို ခန့်မှန်းတာကလည်း စိတ်ဝင်စားဖို့ကောင်းတယ်

- Source: https://yoast.com/yoast-seo-july-12-2022/

Project Details & Guidance (Readability Team)

- စာမျက်နှာ ၁၁ မှာ

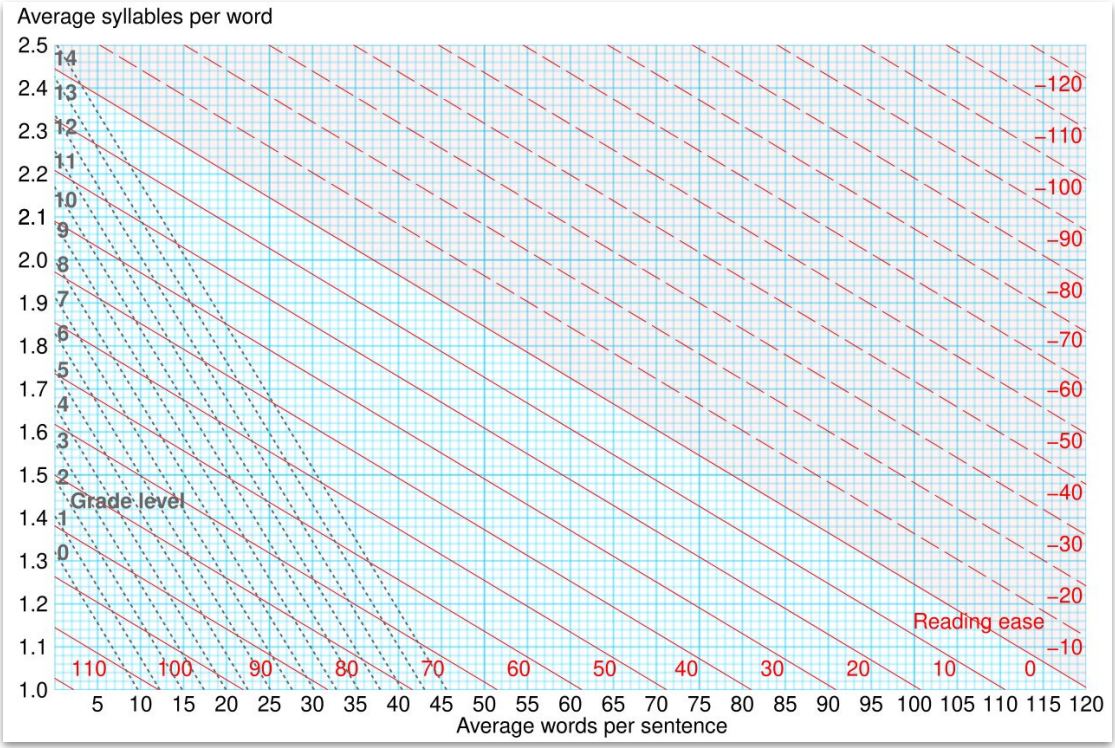
4. **Flesch-Kincaid Grade Level** (1975), classifies texts into American school grades.

$$Grade = .39sl + 11.8wl - 15.59$$

- sl = Average number of word per sentence.
- wl = Average number of syllables per word.

Flesch-Kincaid Grade Level is a modification of Fleschs *Reading Ease Formula*. It translates the former formula to an U.S grade level. A score of 10.2 indicates that the text is understandable for a 10th grade American student. Flesch-Kincaid Grade Level formula is the most used formula that classifies texts into a grade level.

Project Details & Guidance (Readability Team)



- Graphs of Flesch-Kincaid reading ease (red) and grade level (gray) scores against average syllables per word and average words per sentence

Project Details & Guidance (Readability Team)

- How to use the Flesch reading ease score to improve your writing?
  - အလွယ်ပြောရရင် စာကြောင်းတွေကို တိုတိုရေး
  - နားလည်ရ ခက်တဲ့ စာလုံးတွေကို တတ်နိုင်သ၍ လျှော့သုံး
- Flesch Reading-Ease ရော Flesch–Kincaid Grade Level ရောက score တွက်တာက တူတယ် မတူတာက weighting factors

Project Details & Guidance (Readability Team)

- စာမျက်နှာ ၁၁ မှာ

5. **LIX (Läsbarhetsindex)** (1968) revised a few times, developed for Swedish.

$$LIX = wl/s + 100 * w_d/wl$$

- wl = Number of words in the text.
- s = Number of sentences in the text.
- w_d = Number of difficult words in the text, where difficult words is defined as words consisting of more than six letters.

The value from *LIX* has to be interpreted with a *LIX-interpreter*. There are several available, two examples of interpreters is represented in Figure 2.1 [Björnsson, 1971]. Depending on which *LIX-interpreter* is used, *LIX* can be applied to any level of text, just by adjusting the scale. *LIX* have successfully been applied to several other languages with the good results, by simply adjusting the scale of the interpreter [Cedergren, 1992].

Project Details & Guidance (Readability Team)

- စာမျက်နှာ ၁၂ မှာ

LIX value	Description	LIX value	Description
20	Very easy	20-25	Children's books
30	Easy	31-35	Fiction
40	Average	40-45	Newspapers
50	Hard	50-55	Science reports
60	Very Hard	60-	Government texts, law texts ...

Figure 2.1: Two LIX-interpreters, the left is a simple version and the right an adapted interpreter.

- ရှေ့သွေး၊ တေဇ၊ ဝတ္တု၊ သတင်းစာ၊ သိပ္ပံနှင့်နည်းပညာ၊ ဥပဒေ စတာတွေနဲ့ခွဲနိုင်

Project Details & Guidance (Readability Team)

For “Readability,” Amazon lists Fog Index and two indices developed by Rudolf Flesch under commission by the Navy. Keeping it brief, the Fog Index estimates the number of years of formal education required to understand the book. The Flesch-Kincaid Index similarly measures the U.S. grade level likely needed to understand the book, meaning Fog and Flesh-Kincaid numbers should be, in a perfect world, quite similar. The regular Flesch Index is based on a 100 point scale, with 100 being the easiest to read (for frame of reference, a college degree is considered necessary to read a book with a score of 0 to 30, while a 5th grader should be able to understand a book with a score of 90 to 100).

Got all that? Here’s what our sample books scored (Fog, Flesch-Kincaid, Flesch):

Finnegans Wake: 11.8, 9.3, 57.8

Where I’m Calling From: 5.7, 3.9, 84.2

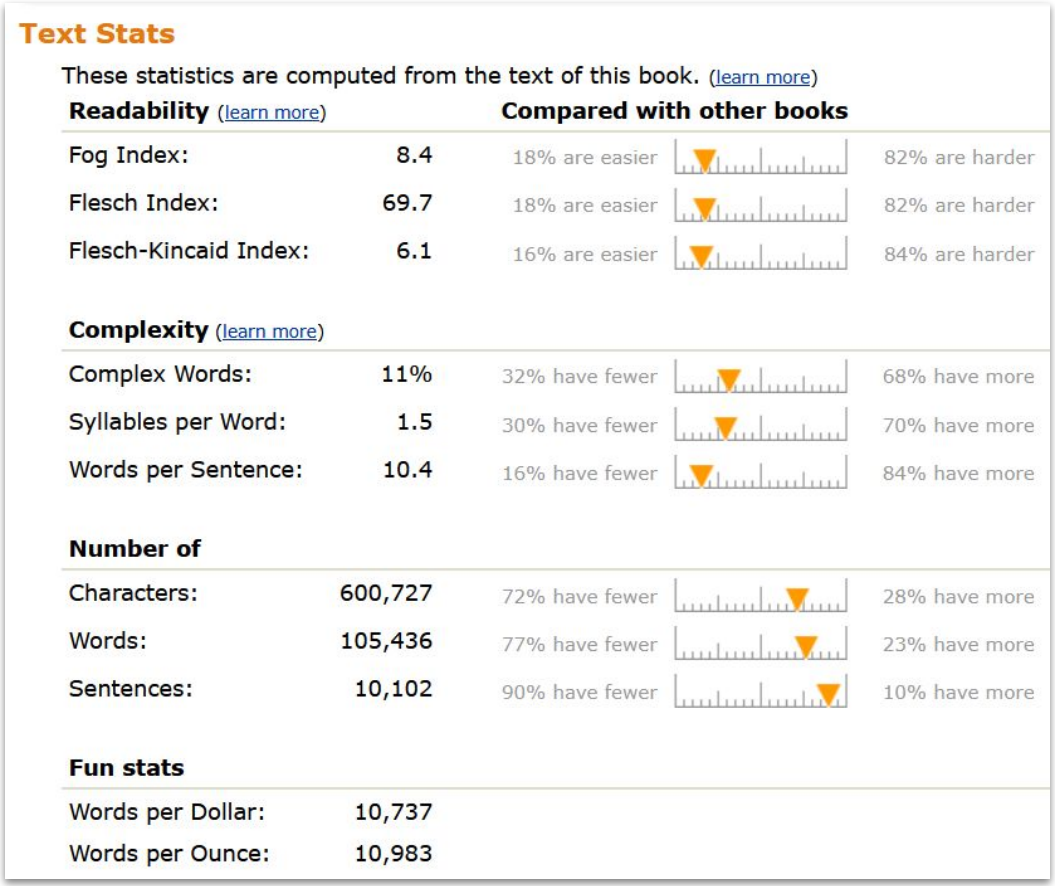
The Great Gatsby: 14.4, 11.8, 48.8

The Memory Keeper’s Daughter: 7.5, 5.6, 76.7

The Tipping Point: 12.6, 10.1, 55.7

- လူတွေ သိပ်ဂရုမပြုမိကြတဲ့ Amazon ရဲ့ Text Stats (i.e. Readability+)
- အခုနောက်ပိုင်း Amazon website မှာ မတွေ့တော့ဘူး

Project Details & Guidance (Readability Team)



Text Stats

These statistics are computed from the text of this book. (learn more)

Readability (learn more)		**Compared with other books**	
Fog Index:	8.4	18% are easier	82% are harder
Flesch Index:	69.7	18% are easier	82% are harder
Flesch-Kincaid Index:	6.1	16% are easier	84% are harder
Complexity (learn more)			
Complex Words:	11%	32% have fewer	68% have more
Syllables per Word:	1.5	30% have fewer	70% have more
Words per Sentence:	10.4	16% have fewer	84% have more
Number of			
Characters:	600,727	72% have fewer	28% have more
Words:	105,436	77% have fewer	23% have more
Sentences:	10,102	90% have fewer	10% have more
Fun stats			
Words per Dollar:	10,737		
Words per Ounce:	10,983		

- လေ့လာကြည့်ပြီး ငါတို့ မြန်မာစာအတွက် Readability Score ကို ဘယ်လိုတွက်ကျမလဲ ဆိုတာကို စဉ်းစားစေချင်တယ်
- Team member တွေအကြားမှာ တိုင်ပင် စေချင်တယ်

Project Details & Guidance (Readability Team)

Readability Formulas: Useful or Useless?

GLENDA M. McCLURE

Abstract – **This interview with Dr. J. Peter Kincaid explores the value of readability formulas and computer editing systems to the engineer who writes on the job. Dr. Kincaid developed the Kincaid Readability Formula, the standard used in judging the reading levels of Department of Defense manuals. Currently a research psychologist for the Army Research Institute, Dr. Kincaid is working on automating the writing and delivery of technical information. Previously, he was a team leader with the Navy's Training Analysis and Evaluation Group [1].**

Among other things which we can reasonably measure: the number of commonly understood words, sentence complexity, the number of abstract ideas, and the use of personal pronouns. Beyond these factors, it takes the expertise of the writer and editor to judge organization of the text and whether or not the text conveys the proper information.

How accurate is the Kincaid formula in determining the correct reading grade level of text?

As a rule of thumb, it is accurate to within plus or minus one reading grade level.

Project Details & Guidance (Readability Team)

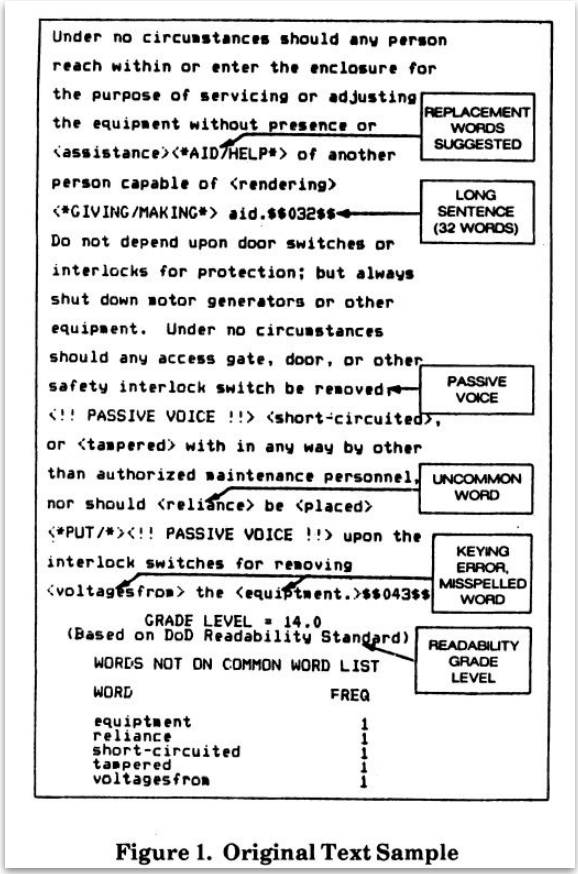


Figure 1. Original Text Sample

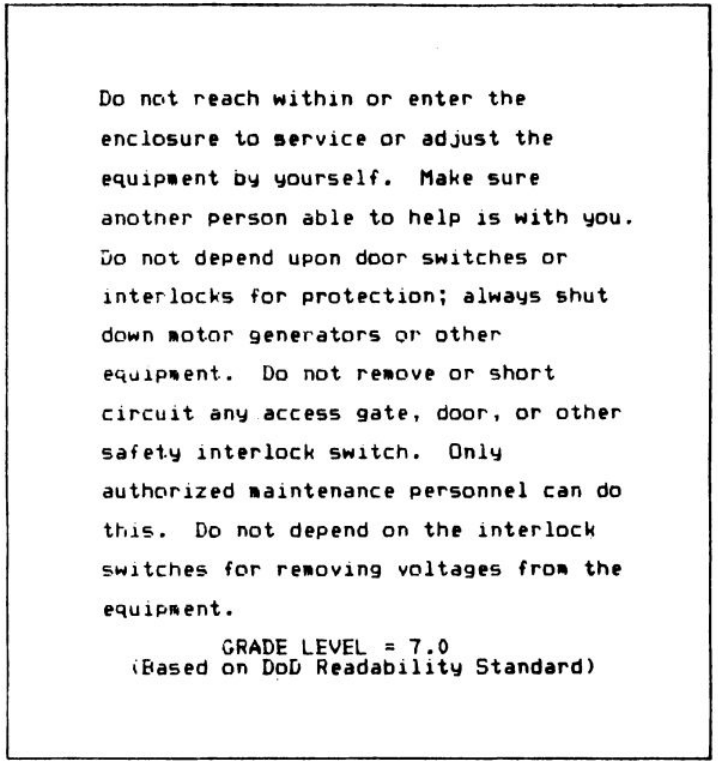


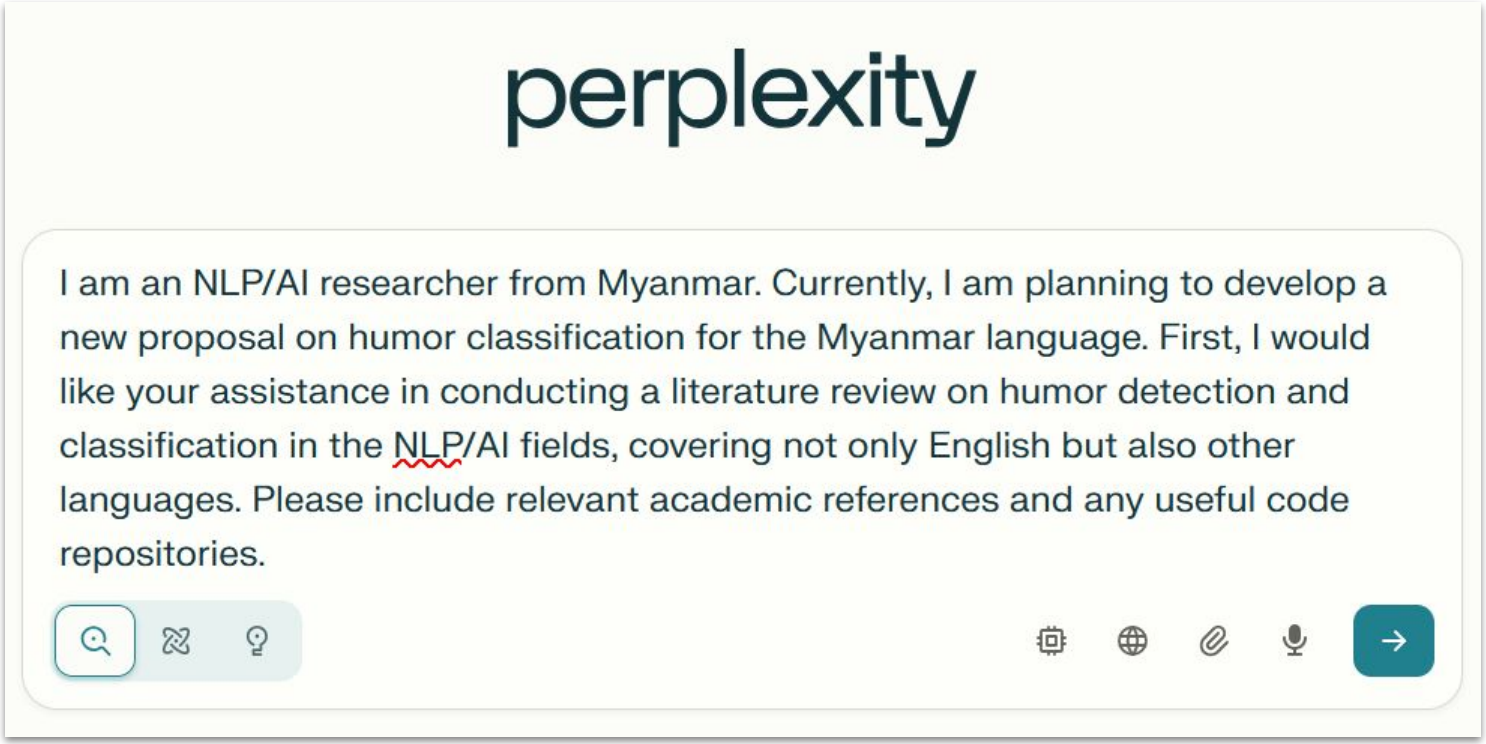
Figure 2. Rewritten Text Sample

Project Details & Guidance (Readability Team)

- စဉ်းစားမိတာက ပထမဆုံး အဆင့်အနေနဲ့ readability ကို user-study or ဆာဗေးလုပ်မယ်
- ဥပမာ ရွှေသွေး၊ တေဇ၊ ဝတ္ထု၊ သတင်းစာ၊ သိပ္ပံနှင့်နည်းပညာ၊ ဥပဒေ စာအုပ်တွေထဲကနေ စာကြောင်း၊ စာပိုဒ်တွေကို ရွေးထုတ်ထားပြီး ကွက်လပ်ဖြည့်ခိုင်းတာမျိုး လုပ်ခိုင်းပြီး ရလဒ်ကို ကြည့်မယ်
- ဒုတိယ တဆင့်အနေနဲ့က data driven approach ဖြစ်တဲ့ readability corpus ကို ဆောက်ထားပြီးမှ model ကနေ classification ဘယ်လောက်လုပ်ပေးနိုင်သလဲ ဆိုတာကို experiment လုပ်ကြည့်တာမျိုး
- အဲဒီ ရလဒ်တွေကိုမှ ငါတို့ရဲ့ proposed formula နဲ့ confirm လုပ်ကြည့်တာမျိုး
- လိုအပ်ရင် formula ကို ပြန် revise လုပ်တာမျိုး (လိုအပ်မယ်လို့ နားလည်)

Project Details & Guidance (Humor Team)

- မသိသေးတဲ့ ကျောင်းသားတွေလည်း ရှိနိုင်လို့။ Literature review အတွက်သုံးပါ။



Project Details & Guidance (Humor Team)

- Check conda environment

```
(base) ye@lst-gpu-server-197:~/intern3/humor$ conda info --envs
# conda environments:
#
base                  *  /home/ye/anaconda3
wordWranglers               /home/ye/anaconda3/envs/wordWranglers
bi_lstm_ner                 /home/ye/anaconda3/envs/bi_lstm_ner
btg-seq2seq                 /home/ye/anaconda3/envs/btg-seq2seq
experiments-tc              /home/ye/anaconda3/envs/experiments-tc
hs-fasttext                 /home/ye/anaconda3/envs/hs-fasttext
mcrf                        /home/ye/anaconda3/envs/mcrf
py2.7                       /home/ye/anaconda3/envs/py2.7
py3.6                       /home/ye/anaconda3/envs/py3.6
py3.7                       /home/ye/anaconda3/envs/py3.7
py3.8                       /home/ye/anaconda3/envs/py3.8
st                          /home/ye/anaconda3/envs/st
textgen                     /home/ye/anaconda3/envs/textgen

(base) ye@lst-gpu-server-197:~/intern3/humor$
```

Project Details & Guidance (Humor Team)

- Activate an existing environment

```
(base) ye@lst-gpu-server-197:~/intern3/humor$ conda activate py3.8
(py3.8) ye@lst-gpu-server-197:~/intern3/humor$
```

- Create a new conda environment

```
(py3.8) ye@lst-gpu-server-197:~/intern3/humor$ conda create -n "humor" python=3.13.5
Collecting package metadata (current_repodata.json): /
```

Project Details & Guidance (Humor Team)

- တချို့ ဆာဗာတွေမှာက ပေးမလုပ်တာမျိုးလည်း ရှိနိုင်

```
(py3.8) ye@lst-gpu-server-197:~/intern3/humor$ conda create -n "humor" python=3.13.5
Collecting package metadata (current_repodata.json): failed

CondaHTTPError: HTTP 000 CONNECTION FAILED for url <https://repo.anaconda.com/pkgs/main/linux-64/current_
repodata.json>
Elapsed: -

An HTTP error occurred when trying to retrieve this URL.
HTTP errors are often intermittent, and a simple retry will get you on your way.

If your current network has https://www.anaconda.com blocked, please file
a support request with your network engineering team.

'https://repo.anaconda.com/pkgs/main/linux-64'
```

Project Details & Guidance (Humor Team)

- ရုံးက GPU ဆာဗာတစ်လုံးမှာ ပေးမလုပ်လို့ တခြား ဆာဗာမှာ ပြောင်းလုပ်ခဲ့တယ်

```
  libzlib             conda-forge/linux-64::libzlib-1.3.1-hb9d3cd8_2
  ncurses             conda-forge/linux-64::ncurses-6.5-h2d0b736_3
  openssl             conda-forge/linux-64::openssl-3.5.0-h7b32b05_1
  pip                 conda-forge/noarch::pip-25.1.1-pyh145f28c_0
  python              conda-forge/linux-64::python-3.13.5-hf636f53_101_cp313
  python_abi          conda-forge/noarch::python_abi-3.13-7_cp313
  readline            conda-forge/linux-64::readline-8.2-h8c095d6_2
  tk                  conda-forge/linux-64::tk-8.6.13-noxft_hd72426e_102
  tzdata              conda-forge/noarch::tzdata-2025b-h78e105d_0


Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate humor
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) ye@lst-hpc3090:~$ conda activate humor
(humor) ye@lst-hpc3090:~$
```

Project Details & Guidance (Humor Team)

- Kaggle binary classification humor dataset ကို download လုပ်ဖို့ပြင်ဆင်
- pip install kaggle

```
Collecting urllib3>=1.15.1 (from kaggle)
  Downloading urllib3-2.4.0-py3-none-any.whl.metadata (6.5 kB)
Collecting webencodings (from kaggle)
  Using cached webencodings-0.5.1-py2.py3-none-any.whl.metadata (2.1 kB)
Downloading kaggle-1.7.4.5-py3-none-any.whl (181 kB)
Downloading certifi-2025.4.26-py3-none-any.whl (159 kB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading setuptools-80.9.0-py3-none-any.whl (1.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 23.1 MB/s eta 0:00:00
Using cached six-1.17.0-py2.py3-none-any.whl (11 kB)
Downloading urllib3-2.4.0-py3-none-any.whl (128 kB)
Using cached bleach-6.2.0-py3-none-any.whl (163 kB)
Downloading charset_normalizer-3.4.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (148 kB)
Using cached idna-3.10-py3-none-any.whl (70 kB)
Downloading protobuf-6.31.1-cp39-abi3-manylinux2014_x86_64.whl (321 kB)
Using cached python_slugify-8.0.4-py2.py3-none-any.whl (10 kB)
Using cached text_unidecode-1.3-py2.py3-none-any.whl (78 kB)
Downloading requests-2.32.4-py3-none-any.whl (64 kB)
Using cached tqdm-4.67.1-py3-none-any.whl (78 kB)
Using cached webencodings-0.5.1-py2.py3-none-any.whl (11 kB)
Installing collected packages: webencodings, text-unidecode, urllib3, tqdm, six, setuptools, python-slugi
fy, protobuf, idna, charset-normalizer, certifi, bleach, requests, python-dateutil, kaggle
Successfully installed bleach-6.2.0 certifi-2025.4.26 charset-normalizer-3.4.2 idna-3.10 kaggle-1.7.4.5 p
rotobuf-6.31.1 python-dateutil-2.9.0.post0 python-slugify-8.0.4 requests-2.32.4 setuptools-80.9.0 six-1.1
7.0 text-unidecode-1.3 tqdm-4.67.1 urllib3-2.4.0 webencodings-0.5.1
(humor) ye@lst-hpc3090:~$
```

Project Details & Guidance (Humor Team)

- Kaggle account profile ဖိုင်ကို တခြား user တွေ access လုပ်လို့မရအောင်ပြောင်း

```
(humor) ye@lst-hpc3090:~$ ls .kaggle/kaggle.json
.kaggle/kaggle.json
(humor) ye@lst-hpc3090:~$
(humor) ye@lst-hpc3090:~$ chmod 600 .kaggle/kaggle.json
(humor) ye@lst-hpc3090:~$
```

Project Details & Guidance (Humor Team)

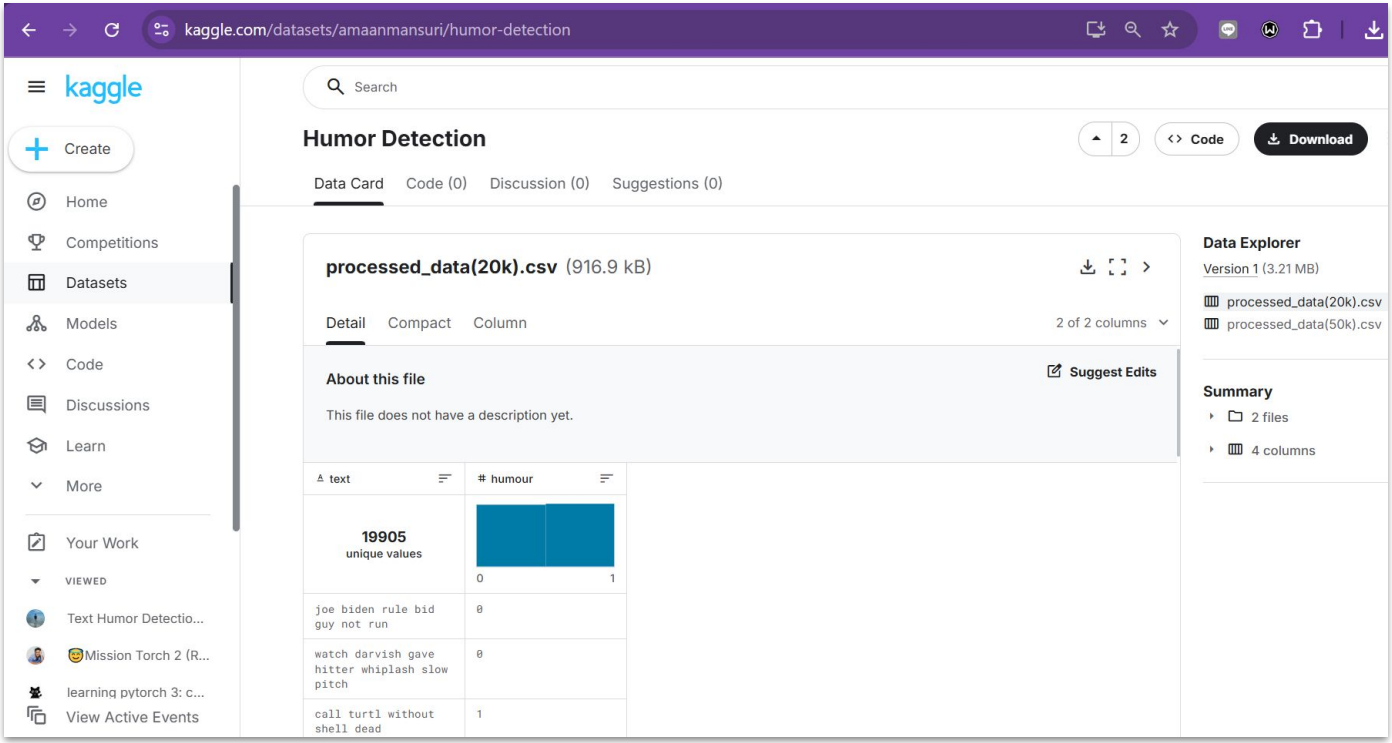
- kaggle command သုံးလို့ ရမရ စမ်းကြည့်

```
(humor) ye@lst-hpc3090:~$ kaggle competitions list
ref                                                                            deadline             cat
egory                reward  teamCount  userHasEntered
-----------------------------------------------------------------------------  -------------------  ---
-----------  -------------  ---------  --------------
https://www.kaggle.com/competitions/arc-prize-2025                            2025-11-03 23:59:00  Fea
tured         1,000,000 Usd        507           False
https://www.kaggle.com/competitions/openai-to-z-challenge                      2025-06-29 23:59:00  Fea
tured           400,000 Usd          0           False
https://www.kaggle.com/competitions/make-data-count-finding-data-references    2025-09-09 23:59:00  Res
earch           100,000 Usd        103           False
https://www.kaggle.com/competitions/waveform-inversion                         2025-06-30 23:59:00  Res
earch            50,000 Usd       1226           False
https://www.kaggle.com/competitions/cmi-detect-behavior-with-sensor-data       2025-09-02 23:59:00  Fea
tured            50,000 Usd        944           False
```

- သို့သော် Kaggle account လုပ်ထားရမယ်
- ဒေတာ download လုပ်ဖို့ဆိုရင် competition အတွက် agreement လိုအပ်တယ်

Project Details & Guidance (Humor Team)

- Humor detection dataset ကို download လုပ်ခဲ့ (binary classification dataset)



Project Details & Guidance (Humor Team)

```
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle$ wc *.csv
  20001  142032  916899 processed_data(20k).csv
  50001  354846 2290501 processed_data(50k).csv
  70002  496878 3207400 total
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle$ head ./processed_data\(20k\).csv
text,humour
joe biden rule bid guy not run,0
watch darvish gave hitter whiplash slow pitch,0
call turtl without shell dead,1
reason elect feel person,0
pasco polic shot mexican migrant behind new autopsi show,0
martha stewart tweet hideou food photo twitter respond accordingli,0
pokemon master favorit kind pasta wartortellini,1
nativ american hate rain april bring mayflow,1
obama climat chang legaci impress imperfect vulner,0
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle$ head ./processed_data\(50k\).csv
text,humour
joe biden rule bid guy not run,0
watch darvish gave hitter whiplash slow pitch,0
call turtl without shell dead,1
reason elect feel person,0
pasco polic shot mexican migrant behind new autopsi show,0
martha stewart tweet hideou food photo twitter respond accordingli,0
pokemon master favorit kind pasta wartortellini,1
nativ american hate rain april bring mayflow,1
obama climat chang legaci impress imperfect vulner,0
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle$
```

Project Details & Guidance (Humor Team)

- 50k က 20k ကို extend လုပ်ထားတဲ့ပုံရှိတယ်
- 20k ထဲက စာကြောင်းတွေက 50k ဒေတာထဲမှာ ထပ်ပါနေတယ်
- အဲဒါကြောင့် 50k ကိုပဲ သုံးမယ်
- လက်ရှိ ဖိုင်နာမည်ကြီးက အဆင်မပြေဘူး၊ ဖိုင်နာမည်ပြောင်းမယ်

```
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ mv processed_data\(50k\).csv humor_50k.csv
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ wc ./humor_50k.csv
  50001  354846 2290501 ./humor_50k.csv
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ file ./humor_50k.csv
./humor_50k.csv: CSV Unicode text, UTF-8 text
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$
```

Project Details & Guidance (Humor Team)

- Training data, Test data ခွဲ။ 90:10 ထားခဲ့။

```
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ shuf ./humor_50k.csv > ./humor_50k.shuf.csv
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ head -n 45001 ./humor_50k.shuf.csv > train.csv
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ tail -n 5000 ./humor_50k.shuf.csv > test.csv
```

```
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ head ./train.csv
truli headscratch moment donald trump press confer infrastructur,0
indian chief friend got indian chief tattoo arm arm never work,1
destin wed tip bridesmaid budget,0
dark alley johnni optimist beat half life,1
horni pirat worst nightmar sunken chest booti,1
mathematician work home function domain,1
bought drug shoe dealer unlac still got high heel,1
virginia museum open costa concordia exhibit memori day weekend photo,0
peopl nt realiz chickpea get everi manpea make,1
betabrand hire graduat student model result pretti great,0
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$ tail ./test.csv
rob gray man bun ncaa tournament,0
texa put undu burden women choic abort clinic tell suprem court,0
alabama gop told gay sheriff candid run republican,0
nt phone sex might get hear aid,1
not lazi energi save mode,1
q nt blond like butter toast ca nt figur side butter goe,1
like coffe like like tea hot splash milk,1
hope nt take joke liter pleas return later,1
syrian increasingli desper escap wartorn countri,0
herb chicken thigh roast paper bag,0
(humor) ye@lst-hpc3090:~/intern3/humor/kaggle/data$
```

Project Details & Guidance (Humor Team)

```
(humor) ye@lst-hpc3090:~/intern3/humor$ python3.13 ./ml_humor_detection.py --help
usage: ml_humor_detection.py [-h] [--train_file TRAIN_FILE] [--test_file TEST_FILE]
                             [--classifier {all,svm,random_forest,logistic_regression,naive_bayes,knn,dec
ision_tree,adaboost,gradient_boosting,voting}]
                             [--svm_kernel {linear,rbf,poly,sigmoid}] [--svm_c SVM_C]
                             [--rf_n_estimators RF_N_ESTIMATORS] [--rf_max_depth RF_MAX_DEPTH]
                             [--lr_c LR_C] [--lr_penalty {l1,l2,elasticnet,none}] [--nb_alpha NB_ALPHA]
                             [--knn_n_neighbors KNN_N_NEIGHBORS] [--dt_max_depth DT_MAX_DEPTH]
                             [--ab_n_estimators AB_N_ESTIMATORS] [--ab_learning_rate AB_LEARNING_RATE]
                             [--gb_n_estimators GB_N_ESTIMATORS] [--gb_learning_rate GB_LEARNING_RATE]

Humor Detection using Traditional ML Techniques

options:
  -h, --help            show this help message and exit
  --train_file TRAIN_FILE
                        Path to training CSV file (default: train.csv)
  --test_file TEST_FILE
                        Path to testing CSV file (default: test.csv)
  --classifier {all,svm,random_forest,logistic_regression,naive_bayes,knn,decision_tree,adaboost,gradient
_boosting,voting}
                        Classifier to use (default: all)
  --svm_kernel {linear,rbf,poly,sigmoid}
                        Kernel type for SVM (default: linear)
  --svm_c SVM_C         Regularization parameter for SVM (default: 1.0)
  --rf_n_estimators RF_N_ESTIMATORS
                        Number of trees in Random Forest (default: 100)
  --rf_max_depth RF_MAX_DEPTH
                        Maximum depth of trees in Random Forest (default: None)
  --lr_c LR_C           Inverse of regularization strength for Logistic Regression (default: 1.0)
  --lr_penalty {l1,l2,elasticnet,none}
                        Penalty norm for Logistic Regression (default: l2)
  --nb_alpha NB_ALPHA   Additive smoothing parameter for Naive Bayes (default: 1.0)
  --knn_n_neighbors KNN_N_NEIGHBORS
                        Number of neighbors for KNN (default: 5)
  --dt_max_depth DT_MAX_DEPTH
                        Maximum depth for Decision Tree (default: None)
  --ab_n_estimators AB_N_ESTIMATORS
                        Number of estimators for AdaBoost (default: 50)
  --ab_learning_rate AB_LEARNING_RATE
                        Learning rate for AdaBoost (default: 1.0)
  --gb_n_estimators GB_N_ESTIMATORS
                        Number of estimators for Gradient Boosting (default: 100)
  --gb_learning_rate GB_LEARNING_RATE
                        Learning rate for Gradient Boosting (default: 0.1)
```

- Coding for machine learning approaches

Project Details & Guidance (Humor Team)

- လိုအပ်တဲ့ python library တွေကို installation လုပ်ခဲ့တယ်

```
(humor) ye@lst-hpc3090:~/intern3/humor$ pip install scikit-learn
Collecting scikit-learn
  Downloading scikit_learn-1.7.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (17
kB)
Collecting numpy>=1.22.0 (from scikit-learn)
  Downloading numpy-2.3.0-cp313-cp313-manylinux_2_28_x86_64.whl.metadata (62 kB)
Collecting scipy>=1.8.0 (from scikit-learn)
  Downloading scipy-1.15.3-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (61 kB)
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.5.1-py3-none-any.whl.metadata (5.6 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Using cached threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Downloading scikit_learn-1.7.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.5/12.5 MB 56.7 MB/s eta 0:00:00
Downloading joblib-1.5.1-py3-none-any.whl (307 kB)
Downloading numpy-2.3.0-cp313-cp313-manylinux_2_28_x86_64.whl (16.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.6/16.6 MB 68.0 MB/s eta 0:00:00
Downloading scipy-1.15.3-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (37.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 37.3/37.3 MB 74.6 MB/s eta 0:00:00
Using cached threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Installing collected packages: threadpoolctl, numpy, joblib, scipy, scikit-learn
Successfully installed joblib-1.5.1 numpy-2.3.0 scikit-learn-1.7.0 scipy-1.15.3 threadpoolctl-3.6.0
(humor) ye@lst-hpc3090:~/intern3/humor$
```

Project Details & Guidance (Humor Team)

- Pandas installation

```
(humor) ye@lst-hpc3090:~/intern3/humor$ pip install pandas
Collecting pandas
  Downloading pandas-2.3.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (91 kB)
Requirement already satisfied: numpy>=1.26.0 in /home/ye/miniforge3/envs/humor/lib/python3.13/site-packag
es (from pandas) (2.3.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/ye/miniforge3/envs/humor/lib/python3.13/si
te-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Using cached pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /home/ye/miniforge3/envs/humor/lib/python3.13/site-packages (f
rom python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading pandas-2.3.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.0 MB)
   ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 12.0/12.0 MB 57.8 MB/s eta 0:00:00
Using cached pytz-2025.2-py2.py3-none-any.whl (509 kB)
Using cached tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.3.0 pytz-2025.2 tzdata-2025.2
(humor) ye@lst-hpc3090:~/intern3/humor$
```

Project Details & Guidance (Humor Team)

- tabulate installation

```
(humor) ye@lst-hpc3090:~/intern3/humor$ pip install tabulate
Collecting tabulate
  Using cached tabulate-0.9.0-py3-none-any.whl.metadata (34 kB)
Using cached tabulate-0.9.0-py3-none-any.whl (35 kB)
Installing collected packages: tabulate
Successfully installed tabulate-0.9.0
(humor) ye@lst-hpc3090:~/intern3/humor$
```

Project Details & Guidance (Humor Team)

- Training/Testing with ML approaches

```
Comparison of All Classifiers:
+---------------------+------------+-------------+----------+------------+---------+
| Model               |   Accuracy |   Precision |   Recall |   F1 Score | Time    |
+=====================+============+=============+==========+============+=========+
| Svm                 |     0.8618 |      0.8619 |   0.8677 |     0.8648 | 243.74s |
+---------------------+------------+-------------+----------+------------+---------+
| Random Forest       |     0.8416 |      0.8295 |   0.8673 |     0.848  | 29.79s  |
+---------------------+------------+-------------+----------+------------+---------+
| Logistic Regression |     0.8652 |      0.866  |   0.87   |     0.868  | 0.04s   |
+---------------------+------------+-------------+----------+------------+---------+
| Naive Bayes         |     0.8658 |      0.8658 |   0.8716 |     0.8687 | 0.01s   |
+---------------------+------------+-------------+----------+------------+---------+
| Knn                 |     0.6062 |      0.5691 |   0.934  |     0.7073 | 1.56s   |
+---------------------+------------+-------------+----------+------------+---------+
| Decision Tree       |     0.7998 |      0.8024 |   0.8053 |     0.8038 | 5.79s   |
+---------------------+------------+-------------+----------+------------+---------+
| Adaboost            |     0.6028 |      0.5639 |   0.9717 |     0.7137 | 1.22s   |
+---------------------+------------+-------------+----------+------------+---------+
| Gradient Boosting   |     0.7214 |      0.8359 |   0.5638 |     0.6734 | 5.97s   |
+---------------------+------------+-------------+----------+------------+---------+
| Voting              |     0.8728 |      0.8691 |   0.8834 |     0.8762 | 277.82s |
+---------------------+------------+-------------+----------+------------+---------+

Best model: Voting with F1 score: 0.8762

real    9m26.984s
user    9m33.239s
sys     0m1.215s
(humor) ye@lst-hpc3090:~/intern3/humor$
```

Project Details & Guidance (Humor Team)

- ဆရာ လုပ်ပြခဲ့တာကို အခြေခံပြီး ML approaches တွေကို နားလည်အောင်ကြိုးစားပါ
- Neural Network based approach တွေဖြစ်တဲ့ Bi-LSTM, textCNN, Transformer တွေနဲ့ လုပ်စမ်းကြည့်ပါ
- မြန်မာစာအတွက်က ဒေတာက မရှိသေးတာမို့လို့ myHumor corpus ကို အနည်းဆုံး binary တော့ ဆောက်ကြရလိမ့်မယ်
- ဖြစ်နိုင်ရင် multi class model အတွက် class တွေကို ဘယ်လို သတ်မှတ်ကြရင် ကောင်းမလဲ စဉ်းစားကြ၊ တိုင်ပင်ကြရလိမ့်မယ်
- ဒေတာကို ဘယ်လို စုကြမလဲ၊ လက်တွေ့ ကြုံရမယ့် အခက်အခဲတွေကို ဘယ်လို engineering နည်းလမ်းတွေနဲ့ ကျော်လွှားကြမလဲ တိုင်ပင်ကြမှ ဖြစ်လိမ့်မယ်

Project Details & Guidance (Image Team)

- မြန်မာ အစားအသောက်ပုံတွေကို နေ့စဉ်ဓာတ်ပုံရိုက် စုသွားလို့လည်း ရတယ်

Project Details & Guidance (Image Team)

- ဥပမာ ပုစွန်ကွက်ကြော်သုပ်ဆိုရင်လည်း တစ်ဆိုင်တည်းမှာပဲ အမျိုးမျိုးရိုက်နိုင်

Project Details & Guidance (Image Team)

- မြန်မာမုန့်တွေ ဆိုရင်လည်း ဗန်းထဲမှာ ထည့်ထားတဲ့ပုံ

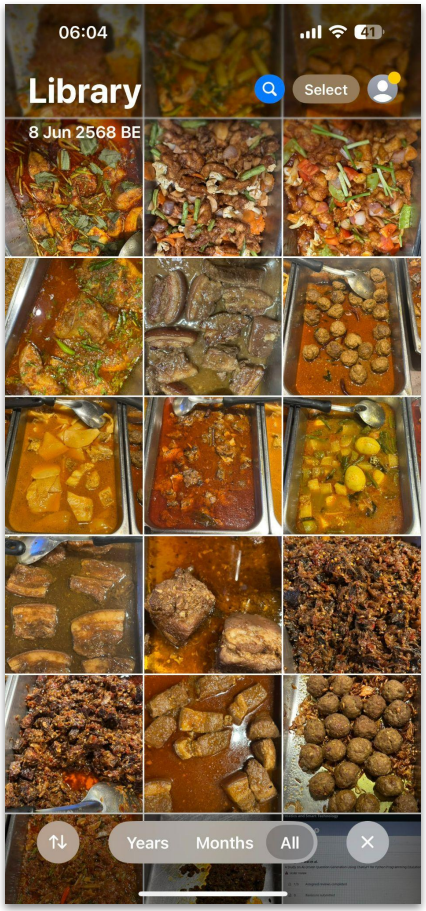
Project Details & Guidance (Image Team)

- ဖော့ဗူးထဲမှာ ထည့်ထားတဲ့ပုံ၊ ပန်းကန်ထဲမှာ ထည့်ထားတဲ့ပုံ စသည်ဖြင့်

Project Details & Guidance (Image Team)

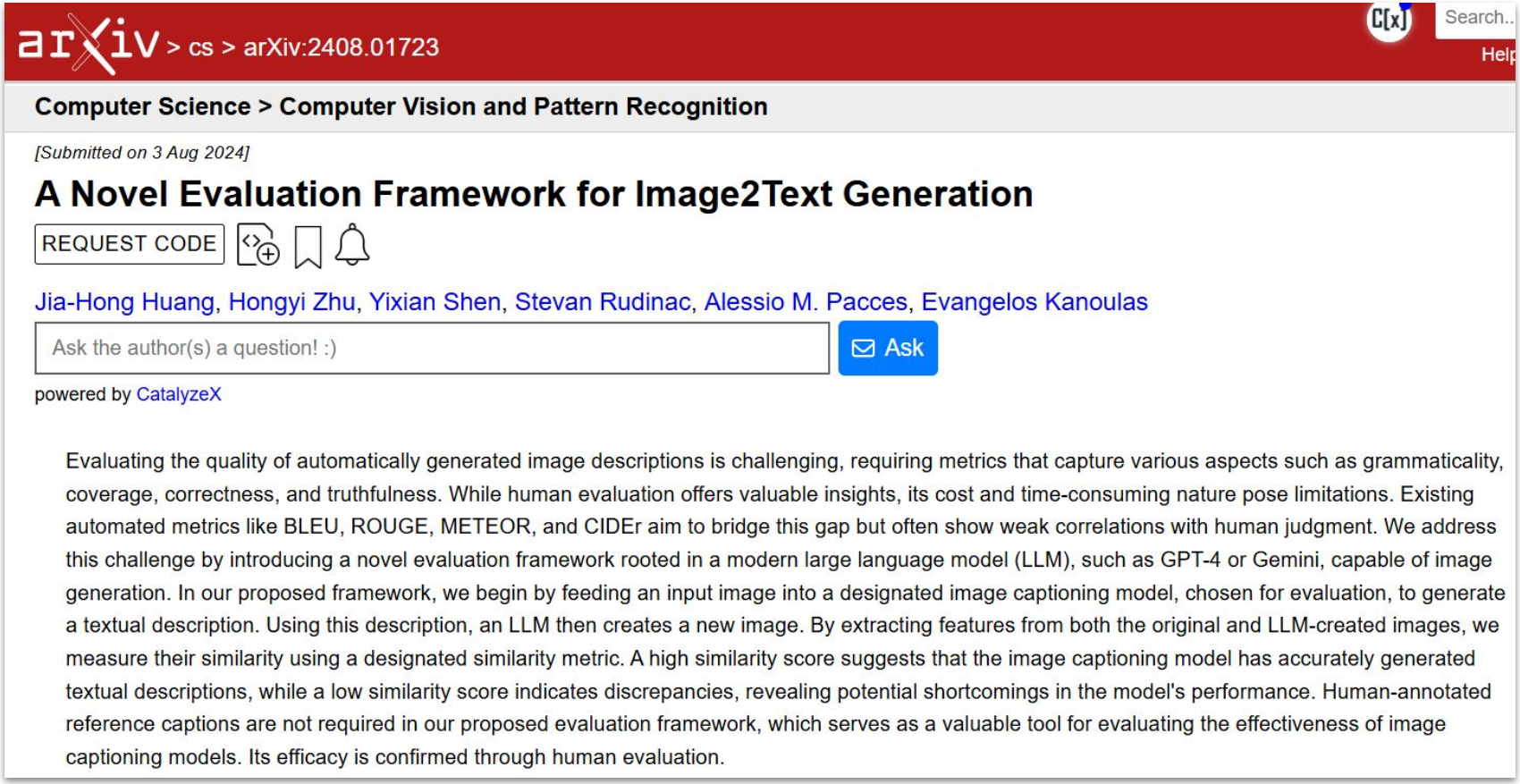
- ဟင်းတွေ ဆိုရင်လည်း အိုးထဲမှာ ထည့်ထားတဲ့ပုံ၊ ချက်လက်စ အငွေ့ထွက်နေတဲ့ပုံ

Project Details & Guidance (Image Team)



- ထမင်းဆိုင်တွေမှာ အကူအညီတောင်းပြီး ရိုက်တာမျိုးလည်း ရလိမ့်မယ်
- တရက်တည်းမှာတင် ကိုယ်စားတဲ့ မနက်စာ၊ နေ့လည်စာ၊ ညနေစာ မှာ ဓာတ်ပုံစုနိုင်
- အရင်ဆုံး မြန်မာအစားအသောက် ကိုပဲ ဘယ်လို အမျိုးအစား ခွဲကြမှာလဲ ဆိုတာကို သေသေချာချာ သတ်မှတ်ကြရလိမ့်မယ်
- ပြီးရင် ဘာ NLP or AI task အတွက်လည်း စဉ်းစားရမယ်
- ဘယ်လို label ထိုးမလဲဆိုတာကို အသေးစိတ် စဉ်းစားကြရမယ်

Project Details & Guidance (Image Team)



Next Steps & Action Items

- ပြင်ထားတဲ့ ဆလိုက်တွေကို ကြည့်ရင်ပဲ ဆရာ တတ်နိုင်သလောက် အချိန်ပေး ပြင်ဆင်ထားတယ် ဆိုတာကို ခန့်မှန်းနိုင်ပါလိမ့်မယ်
- နောက် တစ်ပတ် 3rd Seminar မှာတော့ တစ်ဖွဲ့ချင်းစီက ကိုယ့် proposal/project အတွက် ဘယ်လိုစဉ်းစားထားတယ်၊ ဘယ်လိုပုံစံနဲ့ ဒေတာကို စုကြမယ်၊ ဘယ်လို schedule နဲ့ သွားမယ် ဆိုတာကို ဆရာ နားထောင်ချင်တယ်
- Project ခေါင်းစဉ် တစ်ခုချင်းစီအတွက် လက်ရှိ တခြားသူတွေ ဘယ်လို နည်းလမ်းတွေနဲ့ လုပ်နေကြတယ် ဆိုတာကိုလည်း literature review စလုပ်ကြရအောင်
- ဒေတာကိုလည်း စပြင်ဆင်နိုင်တဲ့သူက ပြင်ဆင်ကြရအောင်
- Senior member တွေကလည်း team member တွေနဲ့ စတင် စကားပြောတာ လုပ်ပါ

Open Discussion
&
Q&A

References

1. Dale-chall-3000-words.txt: https://gist.github.com/Abhishek-P/e00edcc6f508640fe24f263f5836a7dc
2. The Flesch reading ease score: why and how to use it: https://yoast.com/flesch-reading-ease-score/
3. Flesch–Kincaid readability tests: https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests
4. Yoast SEO 19.3: Schema improvements, new word complexity assessment: https://yoast.com/yoast-seo-july-12-2022/
5. Derivation of new readability formulas, Kincaid, J.P.; Fishburne, R.P.; Rogers, R.L.; Chissom, B.S. (1975) https://apps.dtic.mil/sti/pdfs/ADA006655.pdf

References

6. G. M. McClure, "Readability formulas: Useful or useless?," in IEEE Transactions on Professional Communication, vol. PC-30, no. 1, pp. 12-15, March 1987, doi: 10.1109/TPC.1987.6449109. keywords: {Readability metrics;Computers;Training;Personnel;Materials;Educational institutions},
7. Book Lies: Readability is Impossible to Measure, Gabe Habash -- July 20th, 2011, https://web.archive.org/web/20140521073931/http://blogs.publishersweekly.com/blogs/PWxyz/2011/07/20/book-lies-readability-is-impossible-to-measure
8. Spacy_readability: https://www.kaggle.com/code/bond005/spacy-readability
9. Setting up Kaggle API in Linux: https://medium.com/@c.venkataramanan1/setting-up-kaggle-api-in-linux-b05332cde53a