

# JLBert: Japanese Light BERT for Cross-Domain Short Text Classification

LREC-COLING 2024

Chandrai Kayal, Sayantan Chattopadhyay, Aryan Gupta, Satyen Abrol, Archie Gugol  
Rakuten Institute of Technology, Japan

## Motivation

### Pre-Trained Language Models (PLMs)

- Computationally heavy models.
- Trained on: Wikipedia and Common Crawl English corpus
- Inferencing -> significant computational resources

### Short Texts

- **Review Titles:**
  - "まあまあ" (So-so)
  - "残念な結果" (Disappointing result)
- **Tweets:**
  - "今日の天気は最高だね" (Today's weather is great)
  - "この本、おすすめです！" (I recommend this book!)
- **Headlines:**
  - "地震、被害拡大" (Earthquake, damage expands)
  - "オリンピック、金メダル獲得" (Olympics, gold medal won)

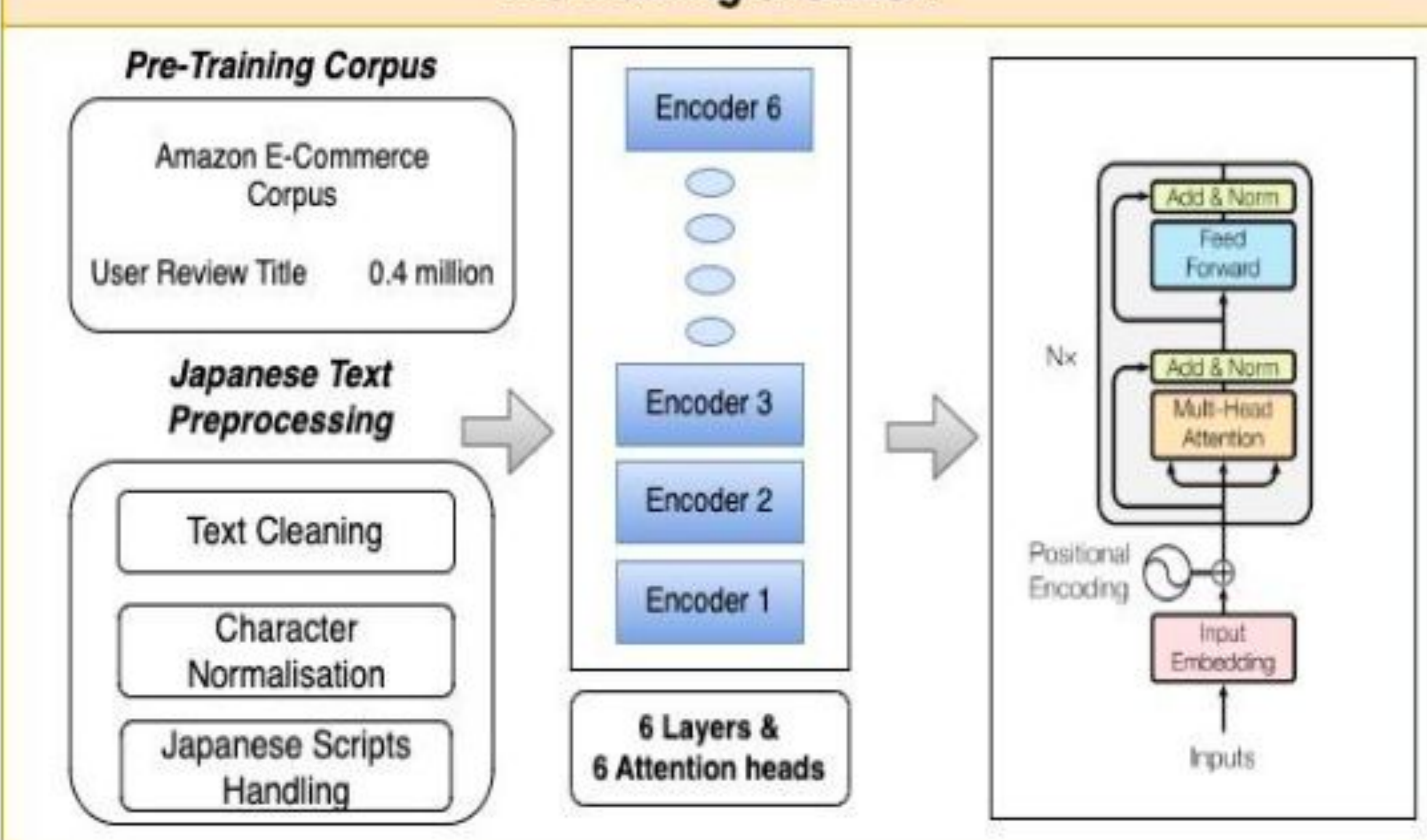
## Scope of Work

- Lack of lightweight models for Japanese texts
- Limited models trained on diverse datasets
- Improving Short Text Classification (STC) tasks performance in Japanese language

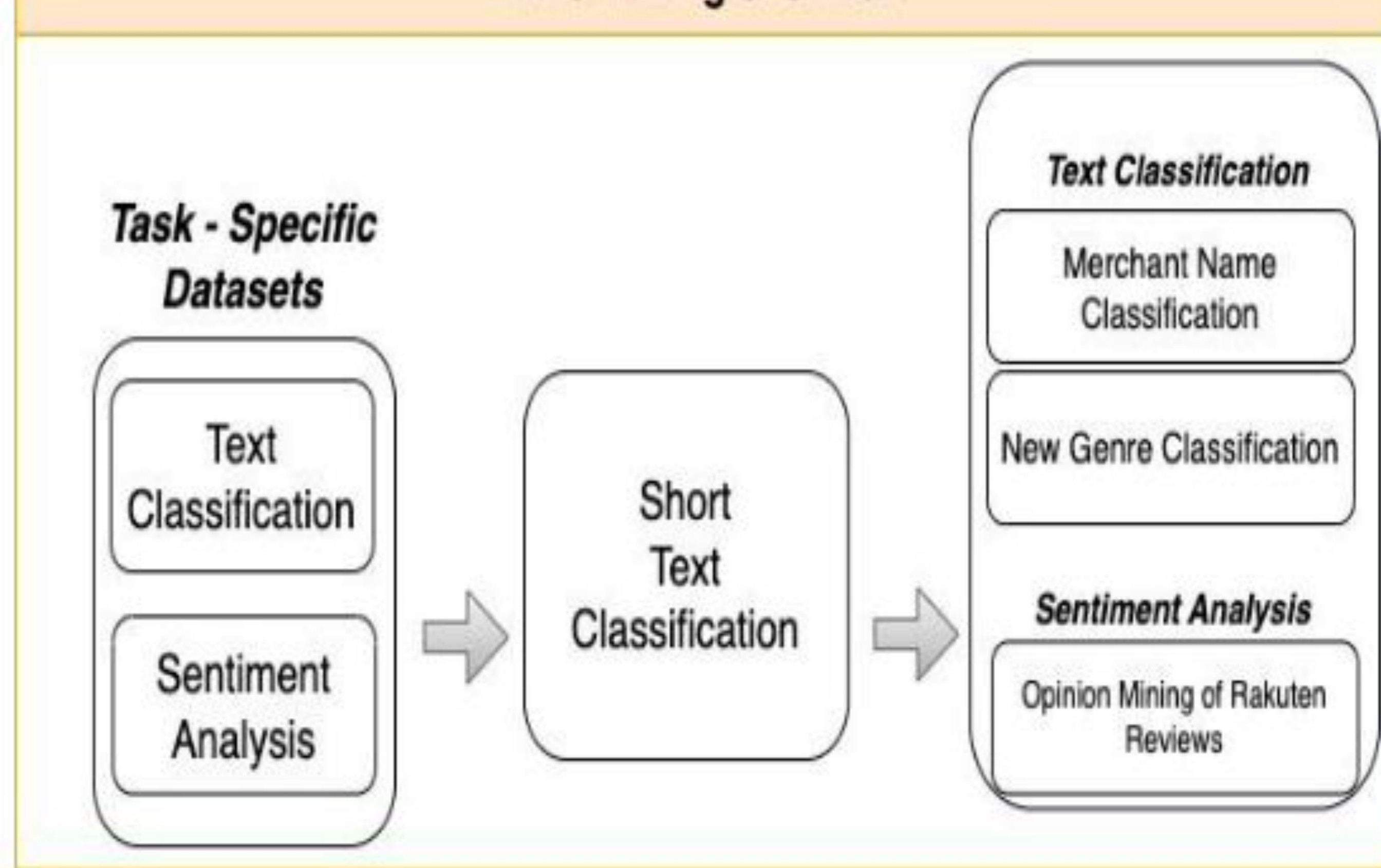
## What is JLBert?

- Lightweight "Greener" BERT Model
- Cross-domain compact and scalable model, specifically designed for STC tasks on Japanese language
- JLBert focusses on: Computational Efficiency, Proficiency in STC tasks and Cross-domain applicability

### Pre-Training of JLBert



### Fine-Tuning of JLBert



Transfer Learning Approach for Fine-Tuning on three Short Text datasets (1 industry + 2 open source datasets)

#### Pre-Training Dataset

- Built a training corpus of **short texts**.
- **0.4M Review Titles** from the Japanese Amazon User Review where the character length of each review title is maximum 20 characters.
- Corpus contains **rich and diverse vocabulary** (books and electronics to clothing and food items)
- Corpus written in 3 scripts: **Hiragana, Katakana and Kanji**.

#### Pre-Processing

- **Text Cleaning:** Removed Html tags, special characters and japanese stop words.
- **Punctuation Normalization:** Special symbols and numerals (Zenkaku or Hankaku). Used neologdn.
- **Character Normalization:** Convert all characters into a standard form. Used Normalization Form KC (NFKC).
- **Handling Japanese scripts:** Katakana-Hiragana conversion to standardize the japanese texts.

#### Training

- **Tokenization Strategy** - Byte-Pair Encoding (BPE)
- 6 hidden layers, 6 attention heads, and 768 hidden sizes (lighter than BERT)
- Vocabulary size is 30,522
- Parameters: 13M
- 4 Tesla V100 GPUs
- 4 \* V100 \* 8hours

#### → Merchant Name Classification (MNC)

- ◆ Credit Card Merchant Genre categorization model.
- ◆ Highly imbalanced data with 23 class classification.
- **Classification of Japan NHK shows**
  - ◆ News genre classification from short titles of a show. Contains 21,795 different show titles with 13 classes.
- **Sentiment Analysis**
  - ◆ Rakuten review titles is sentiment analysis dataset.
  - ◆ Contains 40k review titles with five user polarity classes

## Fine-Tuning Performance (SOTA BERT Models & LLMs)

Datasets	Metrics	JLBert	mBERT	JapBERT	mDistilBERT	JapDistilBERT
MNC	F1-Score	<b>0.8223</b>	0.8154	0.8092	0.8124	0.6250
	CO2 Emission	<b>0.11723</b>	0.39794	0.35720	0.28244	0.32319
	Runtime	<b>45 mins</b>	91 mins	87 mins	56 mins	58 mins
Japan NHK	F1-Score	<b>0.7268</b>	0.7166	0.7014	0.7101	0.5409
	CO2 Emission	<b>0.05076</b>	0.18375	0.17505	0.16745	0.17469
	Runtime	<b>20 mins</b>	45 mins	41 mins	38 mins	40 mins
Rakuten review titles	F1-Score	<b>0.7400</b>	0.7320	0.7380	0.7335	0.7082
	CO2 Emission	<b>0.10351</b>	0.42562	0.36824	0.29826	0.32561
	Runtime	<b>40 mins</b>	98 mins	92 mins	55 mins	60 mins

Methods	Llama-2	GPT-3.5	GPT-4
Zero-Shot	0.362	0.528	0.641
One-Shot	0.371	0.527	0.641
Few-Shot	0.377	0.538	0.698

Performance comparison of LLMs on MNC dataset for 2000 merchant names. JLBert F1-score for 2000 merchant is 0.855

## Summary

- Developed **compact model** specifically designed for the **Japanese language**.
- JLBert is energy-efficient, resulting in **least CO2 emissions**.
- JLBert outperforms SOTA BERT models by **approx 1.5%**.
- **Runtime for JLBert is minimal** for both fine-tuning and inferencing tasks across all datasets.
- **Few shot classification with LLM models underperform** when compared with JLBert by approx 15%.
- Experiment suggests these models may require substantial fine-tuning to effectively handle datasets like MNC

