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

Chandrai Kayal, Sayantan Chattopadhyay, Aryan Gupta, Satyen Abrol, Archie Gugol
Rakuten Institute of Technology, Tokyo, Japan
{chandrai.kayal, sayantan.c, aryan.gupta, satyen.abrol, archie.gugol}@rakuten.com

Abstract

Models, such as BERT Devlin et al. (2018), have made a significant breakthrough in the Natural Language Processing (NLP) domain solving 11+ tasks. This is achieved by training on a large scale of unlabelled text resources and leveraging Transformers architecture making it the "Jack of all NLP trades". However, one of the popular and challenging tasks in Sequence Classification is Short Text Classification (STC). Short Texts face the problem of being short, equivocal, and non-standard. In this paper, we address two major problems: 1. Improving STC tasks performance in Japanese language which consists of many varieties and dialects. 2. Building a light-weight Japanese BERT model with cross-domain functionality and comparable accuracy with State of the Art (SOTA) BERT models. To solve this, we propose a novel cross-domain scalable model called JLBert, which is pre-trained on a rich, diverse and less explored Japanese e-commerce corpus. We present results from extensive experiments to show that JLBert is outperforming SOTA Multilingual and Japanese specialized BERT models on three Short Text datasets by ≈1.5% across various domain.

Keywords: Information Extraction, Transfer Learning, Transformers, BERT

1. Introduction

In recent years, NLP community has witnessed significant advancements, primarily due to the emergence of highly efficient Pre-trained Language Models (PLMs). However, PLMs are computationally heavy which are mostly trained on vast, encyclopedic datasets such as Wikipedia [1] and the Common Crawl corpus etc Raffel et al. (2020). This has led to two significant gaps: First, there is an absence of lightweight models specifically designed for Japanese text. Second, there are limited models that have been trained on diverse datasets beyond Wikipedia and Common Crawl, limiting experimentation with other forms of text data.

Moreover, with the English language being the focus of research interest, the work done on Japanese language is very scarce. The models used for the English language cannot be used for Japanese because of the structural differences in both languages. Japanese language can be written in different Scripts and fonts like hiragana, katakana, kanji, half-width, full-width, roman etc. Same character will have different meanings in different fonts owing to the fact that classification becomes challenging. For example, some kanjis will have the same hiragana when they have different meanings. 飴 and 雨 are both あめ when converted to hiragana but one means candy and other means rain in Kanji. Therefore, we present JLBert, a novel Japanese compact model designed specifically for STC tasks. JLBert offers a unique approach to language processing by focusing on three main areas:

Computational Efficiency: JLBert is designed with computational cost-effectiveness in mind, making it a "greener" choice in the realm of NLP. By reducing the number of layers and attention heads (6 layers-6 attention heads) compared to the BERT Devlin et al. (2018)(12 layers-12 attention heads), JLBert emerges as a lightweight, fast, and scalable solution, therefore reducing fine-tuning time by 50% across various datasets. This reduction does not compromise on performance, as JLBert outperforms SOTA BERT models by approx 1.5%.

Proficiency in STC tasks: Another uniqueness of JLBert lies in its pre-training data: Japanese Amazon User Review Titles. In contrast to SOTA BERT models, which are pre-trained on longer text articles from Wikipedia, JLBert leverages user review titles to focus on STC tasks, a critical aspect of many real-world applications, from social media sentiment analysis to customer feedback interpretation. Ecommerce data is a rich and underutilized source of short text data that provides a more consumer-centric perspective and offers various insights into user sentiment, preferences, and language usage, which can be vital for various NLP tasks.

Cross-domain applicability: Despite its pretraining on a specific domain (e-commerce) data, JLBert demonstrates impressive fine-tuning adaptability across various domains such as fintech, news, etc as opposed to domain-specific models like FinBERT Yang et al. (2020), BioBERT Lee et al. (2020), Patentbert Lee and Hsiang (2019), SciBERT Beltagy et al. (2019).

[1] BERT model is trained with Wikipedia data

LREC-COLING 2024, pages 9536–9542
20-25 May, 2024.

In summary, the novelty of JLBert lies in its unique pretraining on short texts, its cross-domain nature, and its focus on lightweight "greener" model.

2. Related Work

With the explosive growth of e-commerce and social media, STC has become a topic of conversation in recent years and it is widely used in many applications such as tweet categorisation Sriram et al. (2010), merchant name classification, news classification, and other social networking applications Al Sulaimani and Starkey (2021). Although the applications are extensive, STC can be a really challenging and hard task. For example, Chen et al. (2011) proposed a method to derive latent topics at multiple granularities that can be used as features to enrich the representation of short text. Few years later, Chen et al. (2019) retrieved conceptual information from external knowledge sources and incorporated it into deep neural networks to enhance the semantic representation of short texts. Dandan et al. (2021) proposed a Chinese STC algorithm which uses the BERT model to generate eigenvector representations of short text on the sentence level. This is then used as an input into the Softmax regression model for training and classification purposes. Recently, Hu et al. (2022) proposed a method which combines the user's mental features from Maslow's hierarchy with the short text content. Hence, prior research mostly focuses on building contextual information from an internal data source or extracting external knowledge for model building mostly centering on the English language. However, the main highlight of this research paper is the study of JLBert to solve STC in the less explored Japanese language.

3. Methodology

3.1. Pre-training Dataset

In this study, we focus on building a training corpus of short texts. Short text refers to textual data that is brief and concise, typically composed of few words or sentences where the character length of the sentence is not more than 100 characters. Examples of such include review titles, tweets, or headlines. Our training corpus consists of 400,000 Review Titles from the Japanese Amazon User Review where the character length of each review title is maximum 20 characters Keung et al. (2020). This corpus provides a rich and diverse vocabulary, owing to the wide range of product types reviewed, which range from books and electronics to clothing and food items. This diversity of topics results in a broad spectrum of language use, sentiment, and context, making it an ideal source for short text analysis. Moreover, the most challenging task in Japanese language pre-training is preparing a corpus written in 3 scripts: Hiragana, Katakana and Kanji. Majority of the available corpus is not written in a combination of all scripts. However, the Amazon review corpus is able to capture the challenging nature as it has a mixture of all Japanese scripts.

3.2. Pre-processing

Pre-processing for Japanese text data is slightly more complex compared to English due to the presence of different scripts. Our pre-processing [2] [3] includes the following steps:
1. Text Cleaning: Basic cleaning of the texts are performed which includes removing Html tags, special characters and japanese stop words.
2. Punctuation Normalization: In Japanese, special symbols and numericals can be either in Zenkaku or Hankaku format. Although their meanings are the same, they are recognized as different characters which adds unnecessary information. We used neologdn module for this task.
3. Character Normalization: This process convert all characters into a standard form. For instance, full-width characters (全角) can be converted to half-width characters (半角). For our work, we used Normalization Form KC (NFKC) where characters are decomposed by compatibility, then re-composed by canonical equivalence. NFKC is preferred for japanese text processing over Normalization Form KD (NFKD) as the former maintains the integrity of the original text while still ensuring a high degree of normalization.
4. Handling Japanese scripts: We performed Katakana-Hiragana conversion to standardize the japanese texts. As converting Kanji to Hiragana might lose some information, so we avoided Kanji-Hiragana conversion. We used Pykakasi library to handle scripts conversion.

3.3. Tokenization Strategy

The texts are tokenized using a byte version of Byte-Pair Encoding (BPE) with a vocabulary size of 30,522. The inputs of the model take pieces of 512 contiguous token that may span over documents. The beginning of a new document is marked with <s> and the end of one by </s>.

3.4. Training

JLBert is based on the BERT algorithm that includes a multi-layer bidirectional Transformer. JLBert has 6 hidden layers, 6 attention heads, and 768 hidden sizes, making it to be lighter than the BERT. The vocabulary size is 30,522 including unique BERT-specific tokens: [PAD], [UNK], [CLS], [MASK] [SEP]. JLBert model is trained on Masked Language Modelling (MLM) tasks without using

[2] Japanese preprocessing: japanese stop words, normalization techniques

[3] libraries used: neologdn, pykakasi

Comparison	**mBERT**	**JapBERT**	**mDistilBERT**	**JLBert**
Parameters	110M	110M	66M	13M
Layers/ Hidden Dimensions/ Attention Heads	12/768/12	12/768/12	6/768/12	6/768/6
Pre-Training Data	BooksCorpus + Wikipedia	Japanese Wikipedia	BooksCorpus + Wikipedia	Japanese Amazon Reviews
Training Time	4 TPUs*4 days	1 TPU*5 days	8*V100*3.5 days	2*V100*8hours
Method	Transformer, MLM and NSP	BERT without NSP	BERT-Distillation	BERT without NSP

Table 1: Characteristics of JLBert with other SOTA BERT models

Dataset	**Classes**	**Dataset Size**	**Train Size**	**Test Size**	**Baseline Models**	**Batch Size**	**Epochs**	**Learning Rate**	**Max Sequence Length**
MNC	23	41,096	32,876	8,220	mBERT, JapBERT, mDistilBERT, Jap-DistilBERT	16	10	2e-5	128
Japan NHK	13	21,795	17,436	4,359					
Rakuten review titles	5	40,000	32,000	8,000					

Table 2: Fine-Tuning Dataset Information and Experimental Settings

Next Sentence Prediction (NSP). Additionally, the model is trained with the whole word masking enabled for the MLM objective where 20% of each review is masked. The model is pre-trained excluding any supervised training. The input sequence length is taken as 512. The total number of parameters in this configuration is 13M. The batch size is set to 16, the learning rate is set to 5e-5 with 5 epochs. The entire training is done with the help of 2 Tesla V100 GPUs where we utilized multi-GPU training. JLBert utilizes 2 * V100 * 8hours which is 9X faster than DistilBert (8 * V100 * 3.5days) Sanh et al. (2019). Detailed characteristics of JLBert with other SOTA BERT models are shown in table 1.

4. Fine-Tuning Setup

JLBert is evaluated on Short Text Sequence Classification which includes two downstream tasks: Sentiment Analysis and Text Classification. The dataset statistics and the detailed fine-tuning parameters are shown in table 2.

1. Merchant Name Classification (MNC) is a credit card merchant categorization model. It is an industry dataset on a real-world use-case in which we predict industry categories/ sub-categories from merchant name string which is a combination of Japanese & English characters. Merchant names are short texts not exceeding 35 characters per input text. This is a 23 class classification with a highly imbalanced dataset of 41,096 merchants.

2. Classification of Japan NHK (Japan Broadcasting Corporation) shows into multiple genres. Through NHK's Show Schedule API, we can get [title, subtitle, content, genre] for all the shows scheduled for the next 7 days for TV, Radio, and Internet Radio all across Japan. By using the title of a show, we will try to predict its genre. This news dataset contains 21,795 different show titles with 13 classes. JapanNHK

3. Rakuten review titles is a sentiment analysis dataset containing 40,000 review titles not exceeding 40 characters per title with five classes indicating five different user polarities. Rakutenreview

4.1. Baselines

We considered the following SOTA models as our baseline models. 1. BERT Multilingual model (mBERT) (Devlin et al. (2018)). 2. Japanese BERT model (JapBERT). 3. DistilBERT model (mDistilBERT) (Sanh et al. (2019)). 4. Japanese DistilBERT model (JapDistilBERT).

We also experimented with three Large Language models (LLMs) for Zero-shot, One-shot & Few-shot classification with MNC dataset. 1. Llama-2-13b (Touvron et al. (2023)) 2. GPT-3.5 (Ye et al. (2023)) 3. GPT-4 (Achiam et al. (2023))

4.2. CodeCarbon Comparison

The carbon dioxide emissions estimation (CO_2eq) for each model is calculated as below by CodeCarbon: CO_2eq=Power_consumption(kilowatt-hours)*Carbon_Intensity(kg of CO_2/kilowatt-hour). CodeCarbon uses a world average of 475 gCO_2.eq/KWh when the carbon intensity is not

Datasets	Metrics	JLBert	mBERT	JapBERT	mDistilBERT	JapDistilBERT
MNC	Precision	**0.8220**	0.8151	0.8077	0.8172	0.5853
	Recall	**0.8279**	0.8266	0.8193	0.8222	0.6819
	F1-Score	**0.8223**	0.8154	0.8092	0.8124	0.6250
	CO_2 Emission	**0.11723**	0.39794	0.35720	0.28244	0.32319
	Runtime	**45mins**	91mins	87mins	56mins	58mins
Japan NHK	Precision	**0.7330**	0.7265	0.7045	0.7139	0.5268
	Recall	**0.7284**	0.7107	0.7064	0.7160	0.6082
	F1-Score	**0.7268**	0.7166	0.7014	0.7101	0.5409
	CO_2 Emission	**0.05076**	0.18375	0.17505	0.16745	0.17469
	Runtime	**20mins**	45mins	41mins	38mins	40mins
Rakuten review titles	Precision	**0.7071**	0.6994	0.7029	0.6994	0.6507
	Recall	**0.7896**	0.7862	0.7561	0.7795	0.7785
	F1-Score	**0.7400**	0.7320	0.7380	0.7335	0.7082
	CO_2 Emission	**0.10351**	0.42562	0.36824	0.29826	0.32561
	Runtime	**40mins**	98mins	92mins	55mins	60mins

Table 3: Fine-Tuning Performance comparison of JLBert with other SOTA models for three short text datasets. Numbers marked in bold indicate highest performance for specific dataset.

Datasets	Metrics	JLBert	mBERT	JapBERT	mDistilBERT	JapDistilBERT
MNC	CO_2 Emission	**0.00140**	0.00314	0.00284	0.00263	0.00271
	Inference Time	**39secs**	78secs	72secs	66secs	69secs

Table 4: Inferencing on 5000 data points for MNC dataset comparing JLBert with other SOTA models.

available. For Japan it is approximately 0.55 kg of CO_2 per kilowatt-hour. Schmidt et al. (2021)
We conducted CodeCarbon experiments during fine-tuning each of the datasets. Subsequently, we performed inferencing on the MNC dataset, targeting a sample of 5000 data points.

5. RESULTS

5.1. Short Text Classification Results

From table 3, we can see that for all fine-tuned datasets, JLBert outperformed SOTA mBERT by $\approx$ 1% F1-score, JapBERT by $\approx$ 2.5%, mDistilBERT by $\approx$ 1.5% and JapDistilBERT by $\approx$ 18% for MNC & Japan NHK datasets and $\approx$ 3% for Rakuten review titles. Tables 3 and 4 also present a detailed analysis of CO_2 emissions and runtime for each model during the fine-tuning and inferencing tasks (inferencing is done on MNC dataset for 5000 data points). These tables significantly highlight that JLBert is more energy-efficient, resulting in lower CO_2 emissions compared to the other models. Furthermore, the runtime for JLBert model is minimal in comparison to the other SOTA models for both fine-tuning and inferencing tasks across all datasets. This indicates that JLBert not only demonstrates superior performance in terms of environmental sustainability but also exhibits enhanced efficiency in computational tasks.
Business Impact: MNC is an industry dataset which has been deployed as an automated airflow service for a period of 1 year. The usage of the JLBert model led to a 75% increase in targeted members for advertising and marketing. As a result, yearly revenue for this sector saw a 2.2x increase. The model runs monthly on millions of credit card transaction data without any GPU resources.
In an industrial context, we typically infer approx 5 million data points per month using language models. To provide a comparative perspective, if JLBert takes 39 seconds to process 5000 data points, it would require approx 10 hours to handle 5 million data points. On the other hand, if BERT takes 78 seconds for 5000 data points, the time required to process 5 million data points would be approx 21 hours. Therefore, the use of a more compact and efficient model such as JLBert can result in significant savings in terms of both energy consumption and computational time. This highlights the importance of model selection in large-scale, resource-intensive industrial applications.

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

Table 5: Performance comparison of LLMs on MNC dataset for 2000 merchant names. JLBert F1-score for 2000 merchant names is 0.855

5.2. Short Text Classification with LLMs

In this study, we performed Zero-shot, One-shot, and 15-shot classification strategies with three prominent LLMs: Meta's Llama-2, Open AI's GPT-3.5, and GPT-4. These models were evaluated on the MNC industry dataset. The prompts were employed in various iterations, utilizing different templates. The first approach involved furnishing the entire prompt in English, with predefined genres designated for classification. The second approach incorporated a similar way, however, the prompts were supplied entirely in Japanese. Lastly, a combined approach was adopted, wherein the prompts were composed of a mixture of English and Japanese which proved to be the most effective prompt.

From table 5, we observed GPT-4 F1-score got increased by 5% using few-shots prompting. But still LLM models underperform when compared with JLBert by approx 15%. JLBert F1-score for 2000 merchant names is 0.855. In addition to performance metrics, a cost analysis was also conducted between GPT-4 and JLBert. The GPT-4 model incurs an approximate annual cost of 4.6 million JPY for inferencing on 23 million data points. On the other hand, for the JLBert model, even if we include only the cost of one V-100 GPU for inferencing purposes, it results in a significantly lower annual cost of approximately 1.36 million JPY for inferencing on the same number of data points.

LLMs have been trained on extensive datasets, with the capability to perform general-purpose tasks such as text classification, summarization, etc. Theoretically, these LLMs are expected to demonstrate significant competence in zero-shot or few-shot learning scenarios, given their exposure to diverse training data. However, empirical evidence shows that despite the general nature of tasks, LLMs exhibit suboptimal performance in zero-shot and few-shot learning contexts for MNC dataset. Hence the study suggests that these models may require substantial fine-tuning to effectively handle datasets like the MNC.

6. Ablation Study

We present how the sizes of layers-attention heads affect the fine-tuning performance in table 6. For the pre-training of JLBert, we started with a minimal number of layers while maintaining the number of attention heads constant at six. The number of layers was incrementally increased to observe the resultant changes in the F1 score. Our observations indicated a phenomenon of diminishing returns beyond the sixth layer, which means, the incorporation of further layers did not enhance performance Sajjad et al. (2023). This suggests that the model, by this point, had captured the maximum information it could, rendering additional layers ineffective. If a model possesses an adequate number of parameters for learning all signals, augmenting the layers or heads will not contribute to performance improvement Michel et al. (2019). Additionally, the inclusion of extra layers might lead to over-fitting. Hence, after careful consideration, we decided to go ahead with a lighter architecture of 6 layers-6 attention heads as larger layer models did not help in improving performance drastically and more layers require more computational resources and time to train.

Datasets	Japan NHK	Rakuten-Review Titles
2 L - 6 H	0.7004	0.7126
4 L - 6 H	0.7129	0.7347
6 L - 6 H	**0.7268**	**0.7400**
8 L - 6 H	0.7254	0.7334
10 L - 6 H	0.7218	0.7401
12 L - 6 H	0.7260	0.7324
12 L - 12 H	0.7119	0.7289

Table 6: Ablation Study for two STC datasets with respect to no of layers-no of attention heads (L-H) with JLBert model. Results are shown in terms of F1 score.

7. Conclusion

JLBert represents a compact model specifically designed for the Japanese language, which exhibits unique characteristics compared to the English language. Among these unique features are the distinct scripts (katakana, kanji, hiragana) associated with the texts, the varying methods of writing punctuations and symbols, the presence of numerous dialects, and the absence of spaces between words, which all contribute to the complexity of preprocessing and establishing a model that is effective for diverse types of Japanese tasks.

What sets JLBert apart from conventional Japanese BERT models is its pre-training on short-text e-commerce domain data, a domain that has seen little exploration in the context of pre-training models. JLBert presents considerable advantages to the research community, not merely in terms of the reduced time required for pre-training in comparison to other BERT models, but also in the substantially shorter time required for fine-tuning on various NLP tasks and inferencing, which is particularly advantageous for industry-level use-cases.

The development of JLBert was driven by an unmet need in the research community for a smaller, faster Japanese language model. The use of a more compact and efficient model not only conserves time and energy but also allows researchers and industry professionals to quickly utilize this model for general-purpose Japanese NLP tasks such as classification and sentiment analysis, even in the absence of GPU resources. [4]

[4] JLBert huggingface

8. Ethical Considerations

The purpose of an ethical statement in research is to maintain the credibility of the research, protects the rights and welfare of the participants, and ensures the validity of the research findings. Ethics in research is of utmost importance because it promotes trust, collaboration, and mutual respect among researchers and participants. Without ethics, the research may be biased, inaccurate, or harmful, undermining its value and purpose. In this section, we portray how we adhere to ethic rules for our JLBert research work.

1. Data Collection & Usage: All datasets used in our experiment are open-source and publicly available on the internet, ensuring legal and ethical usage. We have properly cited the sources of these datasets in our work. We have also incorporated an internal dataset, the specifics of which remain undisclosed due to security concerns.

2. Models Usage: Our research extensively utilizes open-source models from Hugging Face, the details of which are provided in our paper. We have obtained necessary permissions from Meta and OpenAI for the use of LLM models like Llama-2, as evidenced by signed consent forms.

3. Potential Misuse: We aim to synchronize our model with Hugging Face for public use. We request that users cite our paper when employing our model for research or industrial purposes. This is to prevent potential misuse of our work and to ensure rightful acknowledgment.

4. Mitigation Measures: To mitigate any form of misuse of research work, we have provided references or citations for all datasets, models, and libraries used in our research. Our paper has been thoroughly reviewed by a compliance team to ensure that no company-specific private information or resources have been made public.

In conclusion, our research adheres to all standards of ethical considerations. We encourage the same from users of our model and research findings, promoting a culture of respect for intellectual property and responsible use of AI technologies.

9. Bibliographical References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sami Al Sulaimani and Andrew Starkey. 2021. Short text classification using contextual analysis. *IEEE Access*, 9:149619–149629.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Jindong Chen, Yizhou Hu, Jingping Liu, Yanghua Xiao, and Haiyun Jiang. 2019. Deep short text classification with knowledge powered attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6252–6259.

Mengen Chen, Xiaoming Jin, and Dou Shen. 2011. Short text classification improved by learning multi-granularity topics. In *Twenty-second international joint conference on artificial intelligence*.

Duan Dandan, Tang Jiashan, Wen Yong, Yuan Kehai, T Jiashan, and W Yong. 2021. Chinese short text classification algorithm based on bert model. *Computer engineering*, 47(1):79–86.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

OpenAI GPT-3.5. Chatgpt model released in 2022. https://platform.openai.com/docs/models/gpt-3-5-turbo.

OpenAI GPT-4. Model released in 2023. https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo.

Yongjun Hu, Jia Ding, Zixin Dou, and Huiyou Chang. 2022. Short-text classification detector: a bert-based mental approach. *Computational Intelligence and Neuroscience*, 2022.

JapanNHK. Japan broadcasting corporation public news dataset. https://api-portal.nhk.or.jp/.

JapBERT. Japanese bert model. https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking.

JapDistilBERT. Japanese distilbert model. https://huggingface.co/bandainamco-mirai/distilbert-base-japanese.

Phillip Keung, Yichao Lu, Gy'űrgy Szarvas, and Noah A. Smith. 2020. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Jieh-Sheng Lee and Jieh Hsiang. 2019. Patentbert: Patent classification with fine-tuning a pre-trained bert model. *arXiv preprint arXiv:1906.02124*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Llama-2-13b. Meta ai llama2 model released in 2023. https://huggingface.co/meta-llama/Llama-2-13b-hf.

mBERT. Bert multilingual model released in 2018. https://huggingface.co/bert-base-multilingual-uncased.

mDistilBERT. Distilbert multilingual model released in 2019. https://huggingface.co/distilbert-base-multilingual-cased.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Rakutenreview. Ichiba rakuten ecommerce website user review public dataset. https://review.rakuten.co.jp/item/1/239152_10217770/1.1/?l2-id=item_review.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. 2021. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing.

Japanese Scripts. Japanese unicode for different scripts. http://www.rikai.com/library/kanjitables/kanji_codes.unicode.shtml.

Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.