An Investigation of Transfer Learning-Based Sentiment Analysis in Japanese

Enkhbold Bataa
ExaWizards, Inc. / Tokyo, Japan
enkhbold.bataa@exwzd.com

Joshua Wu
ExaWizards, Inc. / Tokyo, Japan
joshua.wu@exwzd.com

Abstract

Text classification approaches have usually required task-specific model architectures and huge labeled datasets. Recently, thanks to the rise of text-based transfer learning techniques, it is possible to pre-train a language model in an unsupervised manner and leverage them to perform effectively on downstream tasks. In this work we focus on Japanese and show the potential use of transfer learning techniques in text classification. Specifically, we perform binary and multi-class sentiment classification on the Rakuten product review and Yahoo movie review datasets. We show that transfer learning-based approaches perform better than task-specific models trained on 3 times as much data. Furthermore, these approaches perform just as well for language modeling pre-trained on $\frac{1}{30}$ of Wikipedia. We release our pre-trained models and code as open source.

1 Introduction

Sentiment analysis is a well-studied task in the field of natural language processing and information retrieval (Sadegh et al., 2012; Hussein, 2018). In the past few years, researchers have made significant progress from models that make use of deep learning techniques.(Kim, 2014; Lai et al., 2015; Chen et al., 2017; Lin et al., 2017). However, while there has been significant progress in sentiment analysis for English, not much effort has been invested in analyzing Japanese due to its sparse nature and the dependency on large datasets required by deep learning. Japanese script contains no whitespace, and sentences may be ambiguous such that there are multiple ways to split characters into words, each with a completely different meaning. To see if existing research can make progress in Japanese, we make use of recent transfer learning models such as ELMo (Peters et al., 2018), ULMFiT (Howard and Ruder, 2018), and BERT (Devlin et al., 2018) to each pre-train a language model which can then be used to perform downstream tasks. We test the models on binary and multi-class classification.

Figure 1: Transfer learning-based text classification. First, we train the LM on a large corpus. Then, we fine-tune it on a target corpus. Finally, we train the classifier using labeled examples.

The training process involves three stages as illustrated in Figure 1. The basic idea is similar to how fine-tuning ImageNet (Deng et al., 2009) helps many computer vision tasks (Huh et al., 2016). However, this model does not require labeled data for pre-training. Instead, we pre-train a language model in unsupervised manner and then fine-tune it on a domain-specific dataset to efficiently classify using much less data. This is highly desired since there is a lack of large labeled datasets in practice.

2 Contributions

The following are the primary contributions of this paper:

- We experiment ELMo, ULMFiT and BERT on Japanese datasets including binary and 5-class datasets.
- We do several ablation studies that are helpful for understanding the effectiveness of transfer learning in Japanese sentiment analysis.
- We release our pre-trained models and code[1][2]

[1] base.exawizards.com
[2] allennlp.org/elmo

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4652–4657
Florence, Italy, July 28 - August 2, 2019.

3 Related Work

Here we briefly review the popular neural embeddings and classification model architectures.

3.1 Word Embeddings

Word embedding is defined as the representation of a word as a dense vector. There have been many neural network implementations, including word2vec (Mikolov et al., 2013), fasttext (Joulin et al., 2016) and Glove (Pennington et al., 2014) that embed using a single layer and achieve state-of-the-art performance in various NLP tasks. However, these embeddings are not context-specific: in the phrases "I washed my dish" and "I ate my dish", the word "dish" refers to different things but are still represented by the same embedding.

3.2 Contextualized Word Embeddings

Instead of fixed vector embeddings, Cove (McCann et al., 2017) uses a machine translation model to embed each word within the context of its sentence. The model includes a bidirectional LSTM encoder and a unidirectional LSTM decoder with attention, and only the encoder is used for downstream task-specific models. However, pre-training is limited by the availability of parallel corpora. (e.g. English-French)

ELMo, short for Embeddings from Language Model (Peters et al., 2018) overcomes this issue by taking advantage of large monolingual data in an unsupervised way. The core foundation of ELMo is the bidirectional language model which learns to predict the probability of a target word given a sentence by combining forward and backward language models. ELMo also requires task-specific models for downstream tasks.

Howard and Ruder (2018) proposed a single-model architecture, ULMFiT, that can be used in both pre-training and task-specific fine-tuning. They use novel techniques such as discriminative fine-tuning and slanted triangular learning rates for stable fine-tuning. OpenAI extended the idea by introducing GPT, a multi-layer transformer decoder (Radford et al., 2018). While ELMo uses shallow concatenation of forward and backward language models, ULMFiT and OpenAI GPT are unidirectional.

Devlin et al. argues that this limits the power of pre-trained representations by not incorporating bidirectional context, crucial for word-level tasks such as question answering. They proposed a multi-layer transformer encoder-based model, BERT, trained on masked language modeling (MLM) and next sentence prediction (NSP) tasks. MLM allows bidirectional training by randomly masking 15% of words in each sentence in order to predict them, and NSP helps tasks such as question answering by predicting the order of two sentences.

3.3 Text Classification

Many models have been invented for English text classification, including KimCNN (Kim, 2014), LSTM (Chen et al., 2017), Attention (Chen et al., 2017), RCNN (Lai et al., 2015), etc. However, not much has been done for Japanese. To the best of our knowledge, the current state-of-the-art for Japanese text classification uses shallow (context-free) word embeddings for text classification (Peinan and Mamoru, 2015; Nio and Murakami, 2018). Sun et al. (2018) proposed the Super Characters method that converts sentence classification into image classification by projecting text into images.

Zhang and LeCun (2017) did an extensive study of different ways of encoding Chinese/Japanese/Korean (CJK) and English languages, covering 14 datasets and 473 combinations of different encodings including one-hot, character glyphs, and embeddings and linear, fasttext and CNN models.

This paper investigates transfer learning-based methods for sentiment analysis that is comparable to above mentioned models including Zhang and LeCun (2017) and Sun et al. (2018) for the Japanese language.

4 Dataset

Our work is based on the Japanese Rakuten product review binary and 5 class datasets, provided in Zhang and LeCun (2017) and an Yahoo movie review dataset.[3] Table 1 provides a summary. The Rakuten dataset is used for comparison purposes, while the Yahoo dataset is used for ablation studies due to its smaller size. For the Rakuten dataset, 80% is used for training, 20% for validation, and the test set is taken from Zhang and LeCun (2017); for the Yahoo dataset, 60% is used for training, 20% for validation, and 20% for testing. We used

[3] github.com/dennybritz/sentiment-analysis

Dataset	Classes	Train	Test
Rakuten full	5	4,000,000	500,000
Rakuten binary	2	3,400,000	400,000
Yahoo binary	2	30545	7637

Table 1: Datasets

the Japanese Wikipedia[4] for pre-training the language model for all models so that comparison would be fair.

5 Training

5.1 Pre-Training Language Model

Pre-training a language model is the most expensive part but we train it only once and fine-tune on a target task. We used 1 NVIDIA Quadro GV100 for training ULMFiT and 4 NVIDIA Tesla V100s for ELMo. Text extraction done by WikiExtractor[5], then tokenized by Mecab[6] with IPADIC neologism dictionary[7]. We didn't use the BERT multilingual model[8] due to its incompatible treatment of Japanese: it does not account for *okurigana* (verb conjugations) and diacritic signs which completely change the represented word (e.g. *aisu* "to love" vs. *aizu* "signal").[9][10] Instead, we use the pre-trained BERT$_{BASE}$ model by Kikuta (2019) which has been trained for 1 million steps with sequence length of 128 and 400 thousand additional steps with sequence length of 512. It used the SentencePiece subword tokenizer(Kudo and Richardson, 2018) for tokenization. The models trained with the most frequent 32000 tokens or subwords.

5.2 Fine-Tuning

We use a biattentive classification network (BCN) from McCann et al. (2017) with ELMo as it is known to be state-of-the-art[11] on SST (Socher et al., 2013) datasets. For fine-tuning all models on a target task, we follow the same parameters that were used in the original implementation.[12][13][14] And the same hardware used for pre-training ULMFiT and ELMo in fine-tuning. For BERT, we used single v2.8 TPU.[15]

Model	Rakuten Binary	Rakuten Full
GlyphNet	8.55	48.97
OnehotNet	5.93	45.1
EmbedNet	6.07	45.2
Linear Model	6.63	45.26
Fasttext	5.45	43.27
Super Character	5.15	42.30
BCN+ELMo	4.77	42.95
ULMFiT	**4.45**	41.39
BERT$_{BASE}$	4.68	**40.68**

Table 2: Rakuten test results, in error percentages. Best results from other models (GlyphNet to Super Character) obtained from Zhang and LeCun (2017) and Sun et al. (2018)

6 Results

In this section, we compare the results of ELMo+BCN, ULMFiT, and BERT with models reported in Zhang and LeCun (2017) and other previous state-of-the-art models we mentioned in 3.3. Note that none of the source LM is fine-tuned on a target dataset. Results with these models fine-tuned on target corpora are included in Section 7.1.

6.1 Rakuten Datasets

We trained ELMo+BCN and ULMFiT on the Rakuten datasets for 10 epochs each and selected the one that performed best. Since BERT fine-tunes all of its layers, we only train for 3 epochs as suggested by Devlin et al. (2018). Results are presented in Table 2. All transfer learning-based methods outperform previous methods on both datasets, showing that these methods still work well without being fine-tuned on target corpora.

6.2 Yahoo movie review dataset

The Yahoo dataset is approximately 112 times smaller than the Rakuten binary dataset. We believe that this dataset better represents real life/practical situations. For establishing a baseline, we trained a simple one-layer RNN and an LSTM with one linear layer on top for classification, as well as convolutional, self-attention, and hybrid state-of-the-art models we mentioned in Section 3.3 for comparison. Results shown on Table 3. Similar to rakuten datasets, transfer-learning based methods works better.

[4] dumps.wikimedia.org/
[5] github.com/attardi/wikiextractor
[6] taku910.github.io/mecab/
[7] github.com/neologd/mecab-ipadic-neologd
[8] github.com/google-research/bert/blob/master/multilingual.md
[9] github.com/google-research/bert/issues/133
[10] github.com/google-research/bert/issues/130
[11] nlpprogress.com/english/sentiment_analysis.html
[12] github.com/fastai/fastai
[13] github.com/allenai/allennlp
[14] github.com/google-research/bert#fine-tuning-with-bert
[15] cloud.google.com/tpu/

Model	Yahoo Binary
RNN Baseline	35.29
LSTM Baseline	32.41
KimCNN Kim (2014)	14.25
Self Attention Lin et al. (2017)	13.16
RCNN Lai et al. (2015)	12.67
BCN+ELMo	10.24
ULMFiT	12.20
BERT$_{BASE}$	**8.42**

Table 3: Yahoo test results, in error percentages.

Model	Rak B	Rak F	Yahoo B
BCN+ELMo	4.77	42.95	10.24
ULMFiT	4.45	41.39	12.20
BERT$_{BASE}$	4.68	40.68	8.42
BCN+ELMo$*$	4.65	43.12	8.76
ULMFiT$*$	**4.18**	41.05	**8.52**
BERT [10K steps]$*$	4.94	**40.52**	10.14
BERT [50K steps]$*$	5.52	40.57	-

Table 4: Domain adapted results. ULMFiT$*$ and ELMo$*$ are trained for 5 epochs, while BERT$*$ is trained for 10K and 50K steps.

7 Ablation Study

7.1 Domain Adaptation

Pre-trained language models are usually trained with general corpuses such as Wikipedia. However, the target domain corpus distribution is usually different(movie or product review in our case). Therefore, we fine-tune each source language model on the target corpus (without labels) for a few iterations before training each classifier. The results in Table 4 shows that fine-tuning ULMFiT improves the performance on all datasets while ELMo and BERT shows varied results. We believe that the huge performance improvement of ULMFiT is due to the discriminative fine-tuning and slanted triangular learning rates (Howard and Ruder, 2018) that are used during the domain adaptation process.

7.2 Low-Shot Learning

Low-shot learning refers to the practice of feeding a model with a small amount of training data, contrary to the normal practice of using a large amount of data. We chose the Yahoo dataset for this experiment due to its small size. Experimental results in Table 5 show that, with only $\frac{1}{3}$ of the total dataset, ULMFiT and BERT perform better than task-specific models, while BCN+ELMo shows a comparable result. Clearly, this shows that the models have learned significantly during the transfer learning process.

Model	Yahoo Binary
RNN Baseline	35.29
LSTM Baseline	32.41
KimCNN Kim (2014)	14.25
Self-Attention Lin et al. (2017)	13.16
RCNN Lai et al. (2015)	12.67
BCN+ELMo [$\frac{1}{3}$]	13.51
ULMFiT Adapted [$\frac{1}{3}$]	10.62
BERT$_{BASE}$ [$\frac{1}{3}$]	**10.14**

Table 5: Low-shot learning results for the Yahoo dataset, in error percentages. Transfer learning-based methods are trained on $\frac{1}{3}$ of the total dataset, while the other models are trained on the whole dataset.

Model	Yahoo Binary
BCN+ELMo	10.24
ULMFiT	12.20
ULMFiT Adapted	8.52
BERT$_{BASE}$	**8.42**
BCN+ELMo [100MB]	10.32
ULMFiT Adapted [100MB]	8.57
BERT$_{BASE}$ [100MB]	14.26

Table 6: Comparison of results using large and small corpora. The small corpus is uniformly sampled from the Japanese Wikipedia (100MB). The large corpus is the entire Japanese Wikipedia (2.9GB).

7.3 Size of Pre-Training Corpus

We also investigate whether the size of the source language model affects the sentiment analysis performance on the Yahoo dataset. This is especially important for low-resource languages that do not usually have large amounts of data available for training. We used the ja.text8[16] small text corpus (100MB) from the Japanese Wikipedia to compare with the whole Wikipedia (2.9GB) used in our previous experiments. Table 6 shows slightly lower performance for BCN+ELMo and ULMFiT while BERT performed much worse. Thus, for effective sentiment analysis, a large corpus is required for pre-training BERT.

7.4 Parameter Updating Methods

In its original implementation, when BERT is fine-tuned, all of its layers are trained. This is quite different from fine-tuning ELMo, where its layers are frozen and only task-specific models (BCN in our case) are updated. We experiment with the opposite case for both models and list the results on Table 7

- **BERT as a feature extractor** Pre-trained BERT weights are used for initialization and

[16] github.com/Hironsan/ja.text8

Model	Yahoo Binary
BCN+ELMo	10.24
BCN+ELMo unfreeze	8.65
BERT_{BASE}	8.42
BERT_{BASE} freeze	10.68

Table 7: Results from different parameter updating strategies. BCN+ELMo and BERT_{BASE} are original implementations. BCN+ELMo unfreeze shows experimental results of fine-tuning both BCN and ELMo layers on target dataset while BERT_{BASE} freeze is where BERT_{BASE} layers are frozen and only classifier layer fine-tuned on target dataset

will not be changed. The hidden state associated to the first character of the input is pooled and provided to a linear layer that sits on top. This way, BERT is computationally much cheaper and faster. Result shows that using BERT as a feature extractor shows competitive performance.

- **Unfreezing ELMo** Pre-trained ELMo weights are used for initialization as well; however, weights are changed along with BCN layers. This experiment allows us to compare the performance of freezing/unfreezing ELMo layers. Table 7 shows that fine-tuning ELMo improves performance, comparable to BERT.

8 Conclusion

Our work showed the possibility of using transfer learning techniques for addressing sentiment classification for the Japanese language. We draw following conclusions for future researchers in Japanese doing transfer learning for sentiment analysis task based on experiments we did in Rakuten product review and Yahoo movie review datasets:

1. Adapting domain for BERT likely will not yield good results when the task is binary classification. For all other cases, domain adaptation performs just as well or better.

2. ELMo and ULMFiT perform well even when trained on a small subset of the language model.

3. Fune-tuning both ELMo and BCN layers on a target task improves the performance.

9 Discussion and Future Considerations

This research is a work in progress and will be regularly updated with new benchmarks and baselines. We showed that with only $\frac{1}{3}$ of the total dataset, transfer learning approaches perform better than previous state-of-the-art models. ELMo and ULMFiT perform just as well trained on small corpora, but BERT performs worse since it is designed to be trained on MSM and NSP tasks. Finally, domain adaptation always improves the performance of ULMFiT. We believe that our ablation study and the release of pre-trained models will be particularly useful in Japanese text classification. It is important to note that we did not perform K-fold validation due to their high computational cost. In the future, we will investigate other NLP tasks such as named entity recognition (NER), question answering (QA) and aspect-based sentiment analysis (ABSA) (Pontiki et al., 2016) to see whether results we saw in sentiment analysis is consistent across these tasks. We hope that our experimental results inspire future research dedicated to Japanese.

References

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database.

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*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Minyoung Huh, Pulkit Agrawal, and Alexei A. Efros. 2016. What makes ImageNet good for transfer learning? *arXiv e-prints*, page arXiv:1608.08614.

Doaa Mohey El-Din Mohamed Hussein. 2018. A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, 30(4):330–338.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Yohei Kikuta. 2019. Bert pretrained model trained on japanese wikipedia articles. https://github.com/yoheikikuta/bert-japanese.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Lasguido Nio and Koji Murakami. 2018. Japanese sentiment classification using bidirectional long short-term memory recurrent neural network.

Zhang Peinan and Komachi Mamoru. 2015. Japanese sentiment classification with stacked denoising auto-encoder using distributed word representation.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Mohammad Sadegh, Roliana Ibrahim, and Zulaiha Ali Othman. 2012. Opinion mining and sentiment analysis: A survey. *International Journal of Computers & Technology*, 2(3):171–178.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. 2018. Super Characters: A Conversion from Sentiment Classification to Image Classification. *arXiv e-prints*, page arXiv:1810.07653.

Xiang Zhang and Yann LeCun. 2017. Which Encoding is the Best for Text Classification in Chinese, English, Japanese and Korean? *arXiv e-prints*, page arXiv:1708.02657.