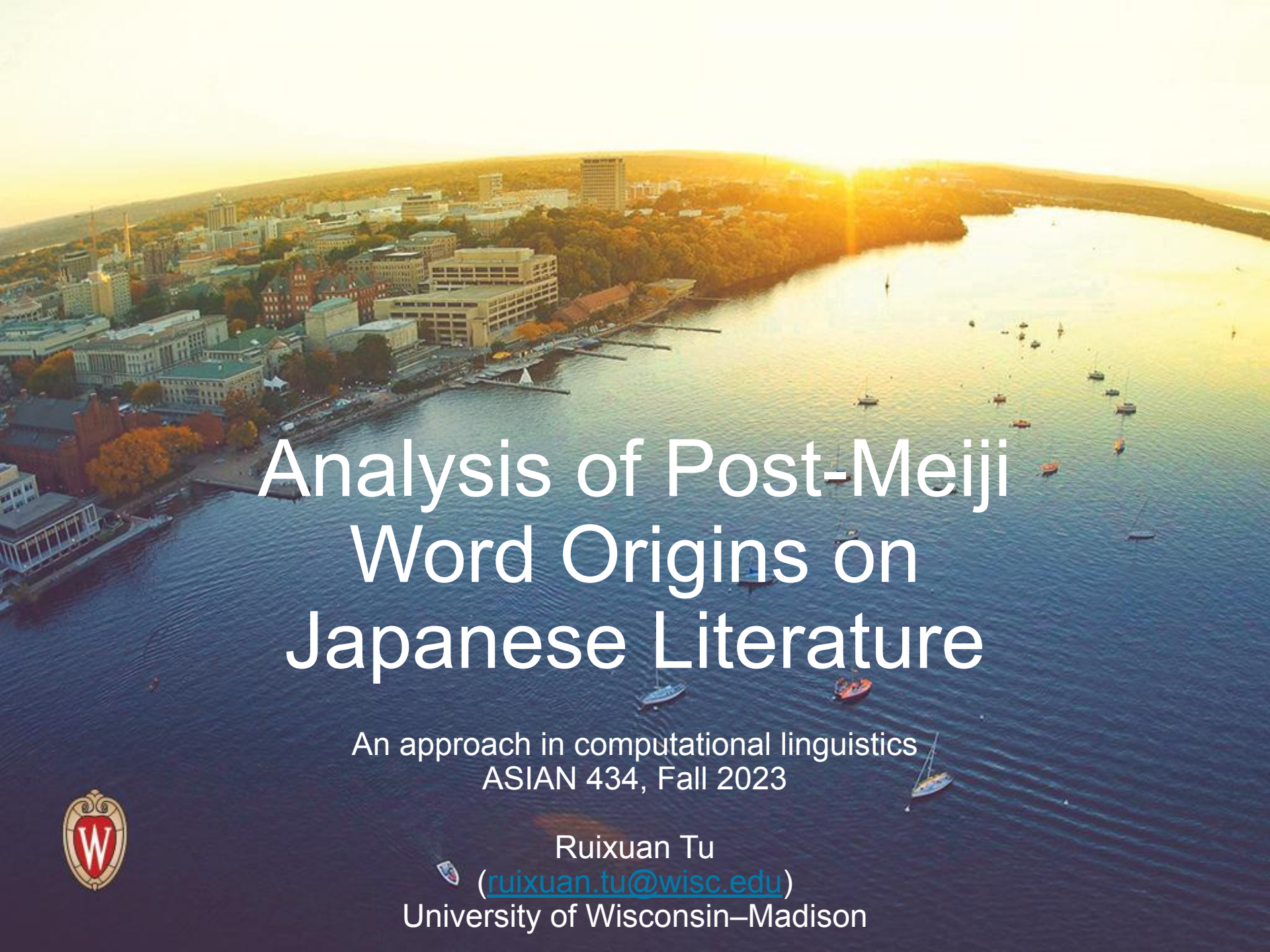


Analysis of Post-Meiji Word Origins on Japanese Literature

An approach in computational linguistics
ASIAN 434, Fall 2023

Ruixuan Tu
(ruixuan.tu@wisc.edu)
University of Wisconsin–Madison

Motivation

Main work: Reproduce statistics of distribution of SJ, native, and foreign words (like magazine in Hasegawa) in the specialized literature area. I further investigate older texts and a continuum of texts.

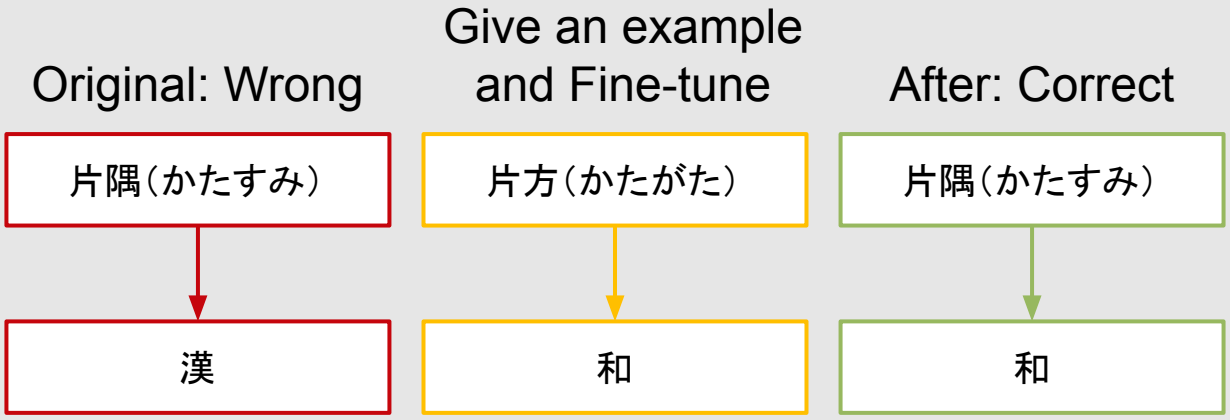
Assumption: the Japanese government advocates the usage of SJ and native words before/in WWII, and western culture becomes more popular after WWII, is this reflected in literature data? No, for mixed (SJ & foreign) words.

Terminologies / Tools

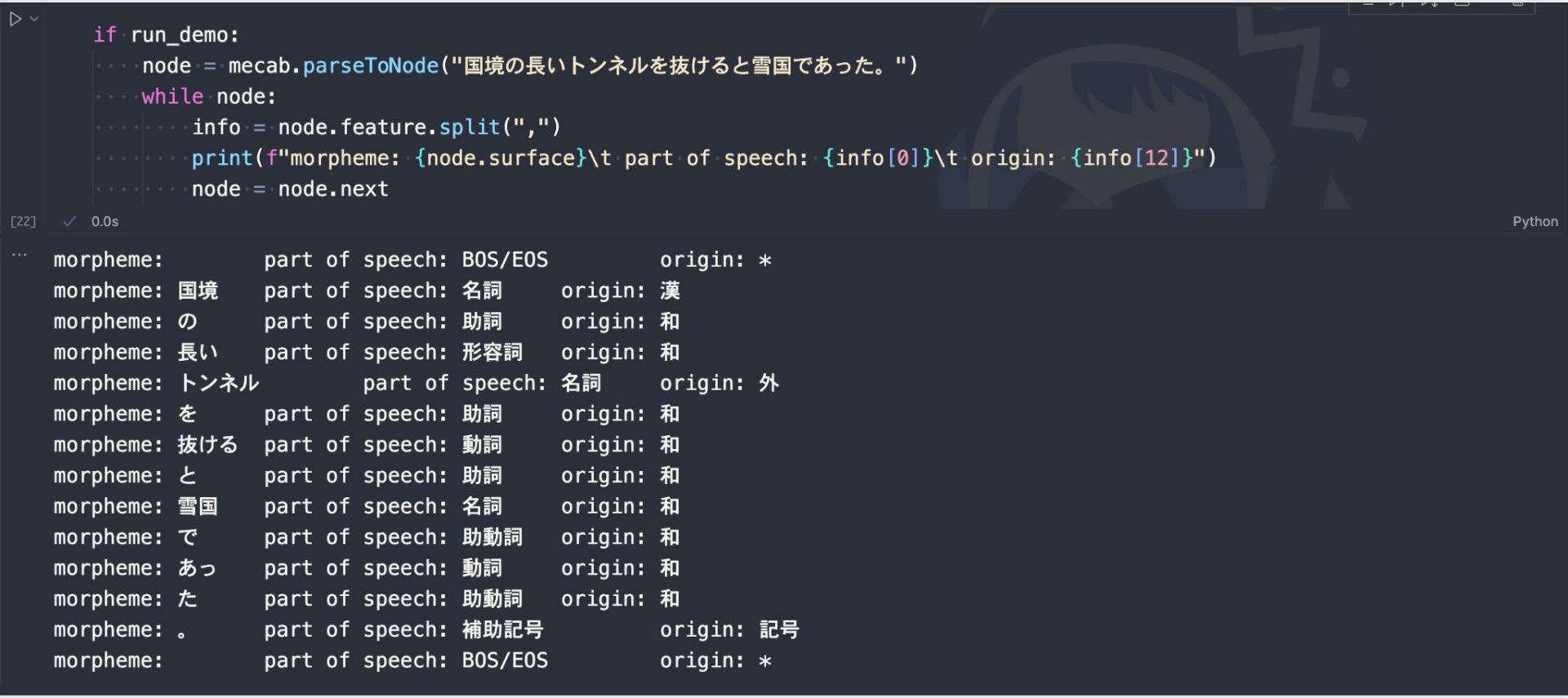
- **morpheme analyzer**: used MeCab/Juman++
  - MeCab uses a dictionary (UniDic) for most time, and predict if there is not a hit in dictionary, using Bi-gram (the current word is conditioned on the previous 2 words) to analyze
  - Juman++ is newer than MeCab, using RNN to analyze (handles longer context), but does not have built-in origin data (need model)
  - According to this paper, Juman++ has better accuracy in tokenization task (used here for morpheme segmentation)
  - e.g., “日本語を勉強する” →
    “[日本語][を][勉強する]” ⭕
    “[日本][語][を][勉強][する]” ❌
- **POS**: **p**art **o**f **s**peech
  - e.g., 名詞 (Noun), 動詞 (Verb), 形容詞 (Adj), 副詞 (Adv)
- **Aozora Bunko**: online library for public domain literature
  - “青空文庫は、誰にでもアクセスできる自由な電子本を、図書館のようにインターネット上に集めようとする活動です。
  - 著作権の消滅した作品と、「自由に読んでもらってかまわない」とされたものを、テキストとXHTML（一部はHTML）形式に電子化した上で揃えています。” ー 青空文庫

Terminology / Tools

- **model**: Given (x, y), where x is a (tokenized) morpheme and y is the origin of the morpheme (in "和", "固", "漢", "外", "混", "記号", and "NA") as in UniDic; the task predicts $\hat{y}$ (predicted origin of morpheme) from x which we might not have seen. The model is a function $f: x \mapsto \hat{y}$. [Input morpheme, output origin]
  - e.g., “会議（かいぎ）” → “漢” (SJ)
    “戯（たわむ）れる” → “和” (native)
    “アップル” → “外” (foreign)
- **fine-tuning**: Given a model with some parameters, provide with new data pairs, and train the model (change the parameters) on the additional data with a small learning rate



Demo: MeCab + UniDic



```
if run_demo:
    node = mecab.parseToNode("国境の長いトンネルを抜けると雪国であった。")
    while node:
        info = node.feature.split(",")
        print(f"morpheme: {node.surface}\t part of speech: {info[0]}\t origin: {info[12]}")
        node = node.next
```

[22] ✓ 0.0s Python

```
morpheme:         part of speech: BOS/EOS          origin: *
morpheme: 国境    part of speech: 名詞     origin: 漢
morpheme: の      part of speech: 助詞     origin: 和
morpheme: 長い    part of speech: 形容詞   origin: 和
morpheme: トンネル          part of speech: 名詞     origin: 外
morpheme: を      part of speech: 助詞     origin: 和
morpheme: 抜ける  part of speech: 動詞     origin: 和
morpheme: と      part of speech: 助詞     origin: 和
morpheme: 雪国    part of speech: 名詞     origin: 和
morpheme: で      part of speech: 助動詞   origin: 和
morpheme: あっ    part of speech: 動詞     origin: 和
morpheme: た      part of speech: 助動詞   origin: 和
morpheme: 。      part of speech: 補助記号          origin: 記号
morpheme:         part of speech: BOS/EOS          origin: *
```

Demo: Juman++ + Model

only consider Noun, Verb, Adj, Adv in origin classification
but it suggests better segmentation

```
if run_demo:
    morphemes = jumanpp.apply("国境の長いトンネルを抜けると雪国であった。").morphemes
    poses = [morpheme.pos for morpheme in morphemes]
    surfs = [morpheme.surf for morpheme in morphemes]
    origins = [x["label"] for x in pipe(surfs, batch_size=n_proc)]
    print(f"surfs: {surfs}")
    print(f"poses: {poses}")
    print(f"origins: {origins}")
```

[25] ✓ 0.1s Python

clever pick!

```
surfs: ['国境', 'の', '長い', 'トンネル', 'を', '抜ける', 'と', '雪国', 'であった', '。']
poses: ['名詞', '助詞', '形容詞', '名詞', '助詞', '動詞', '助詞', '名詞', '判定詞', '特殊']
origins: ['漢', '和', '和', '外', '和', '和', '和', '固', '和', '記号']
```

Main Structure

- Usage of MeCab and Juman++ as preliminary morpheme splitter
- Usage of MeCab as dictionary
- Enhancement: use a fine-tuned model (classifier) on UniDic to predict origins for Juman++-splitted morphemes
- Analysis of morphemes

Procedure of Analysis

1. Create a dataset containing the book ID, the book published year, the author IDs, and the average activity year of the authors.
2. Create another dataset containing the book ID and the content of the book.
3. For every book, run a morpheme analyzer on the content of the book, and check every morpheme on a dictionary or a predictor/classifier for its origin (Sino-Japanese, native, or foreign). Create statistics for content (book ID, part of speech, origin, and frequency).
4. Merge the two datasets by the book ID. Analyze the data by the average activity year of the authors.

- **book id**
- first published year
- average activity year

+

- **book id**
- ~~content of book~~
- part of speech
- origin
- frequency

=

- **book id**
- first published year
- average activity year
- part of speech
- origin
- frequency

analysis

In activity year preprocessing

- Take only books with average activity year >= 1868 (starting of Meiji period, since Japanese has not gone over 言文一致, and the classical grammar could cause inaccuracies in models trained only on modern grammar.
- For the authors less than 120 years old without a death year (2023 - birth year ≤ 120), we estimate activity year = birth year + 60 years.

Methods of Morpheme Analysis

1. Use Juman++ to split the morphemes, and use RoBERTa fine-tuned on UniDic to predict the origin of the morphemes.
2. Use MeCab and UniDic together to split the morphemes and predict the origin of the morphemes.

Procedure of Fine-tuning

1. **Create** a training/validation/test split dataset for categorizing morphemes, from UniDic with 876803 items.
2. **Fine-tune** DeBERTa-v2-base-Japanese model on the dataset.
3. **Evaluate** the model on the evaluation dataset (to get an accuracy for a subset the model have not seen).

accuracy: 0.9162760461217367 out of 87681 samples

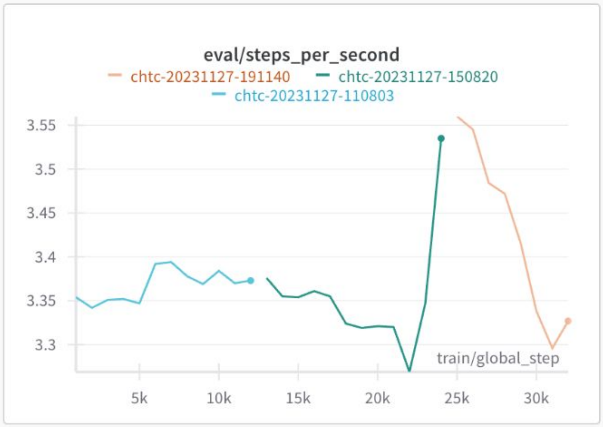
Procedure of Fine-tuning

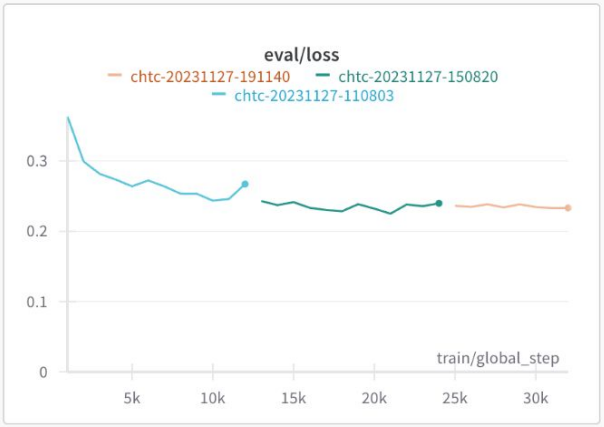
1. Create a training/validation/test split dataset for categorizing morphemes, from UniDic with 876803 items.
2. Fine-tune DeBERTa-v2-base-Japanese model on the dataset.
   a. the original parameters are pre-trained in masking task
   b. we adjust, continue to train the parameters to this task of text classification for morpheme origins by providing examples in training set
   c. we are optimizing the cross-entropy loss that measures our distance between model output logits and the target label
3. Evaluate the model on the evaluation dataset (to get an accuracy for a subset the model have not seen).

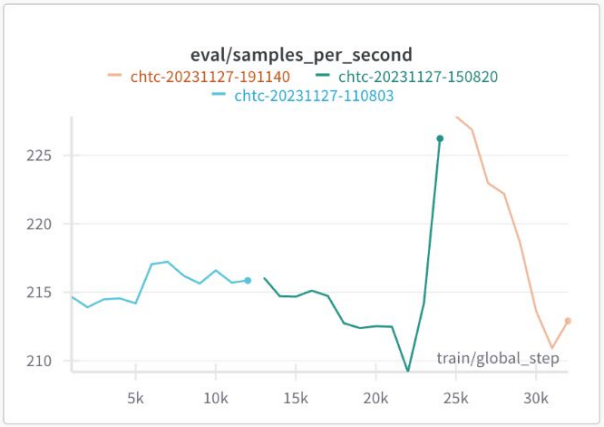
Technical Details of Fine-tuning

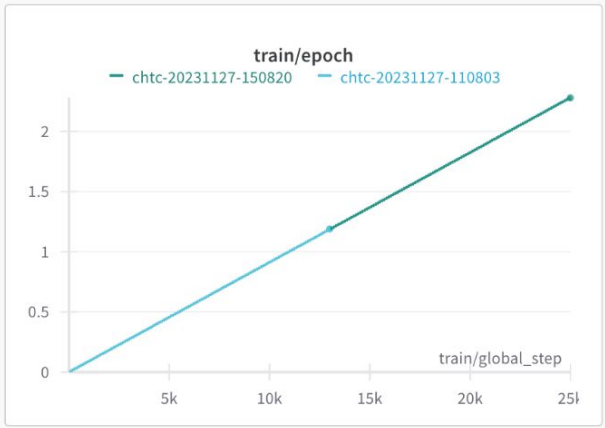
- **foundation model**: ku-nlp/deberta-v2-base-japanese
- **split strategy**: train 80%, validation (in training) 10%, evaluation (after training) 10%
- **train epochs**: 3
- **batch size**: 16
- **GPUs**: 4x NVIDIA A100 40GB @ CHTC
- **evaluation steps**: per 1000 steps
- **save steps**: per 1000 steps
- **total steps**: 32000
- **save strategy**: best (for use) and last (for continue training)
- **learning rate**: 5e-5 (0.00005)
- **weight decay**: 0.01
- **warmup ratio**: 0.1
- **(after-train) evaluation accuracy**: 0.9162760461217367 out of 87681 samples

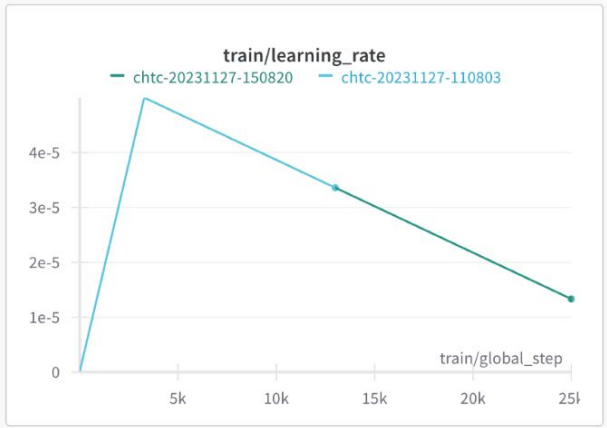
Technical Details of Fine-tuning

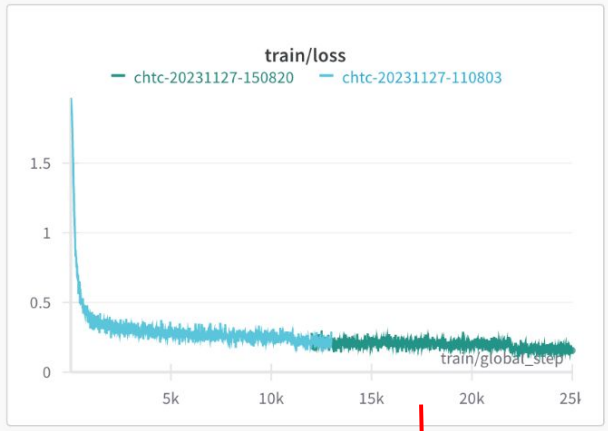


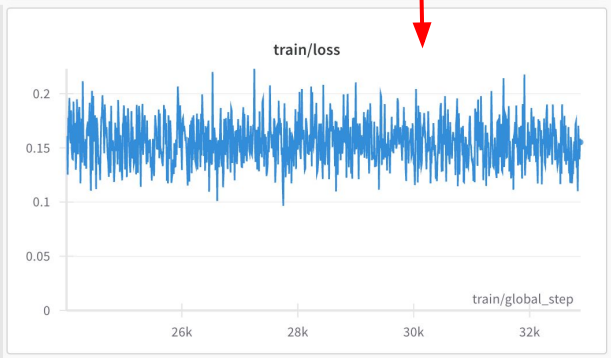












The model's perspective

Demo: some inconsistency in model

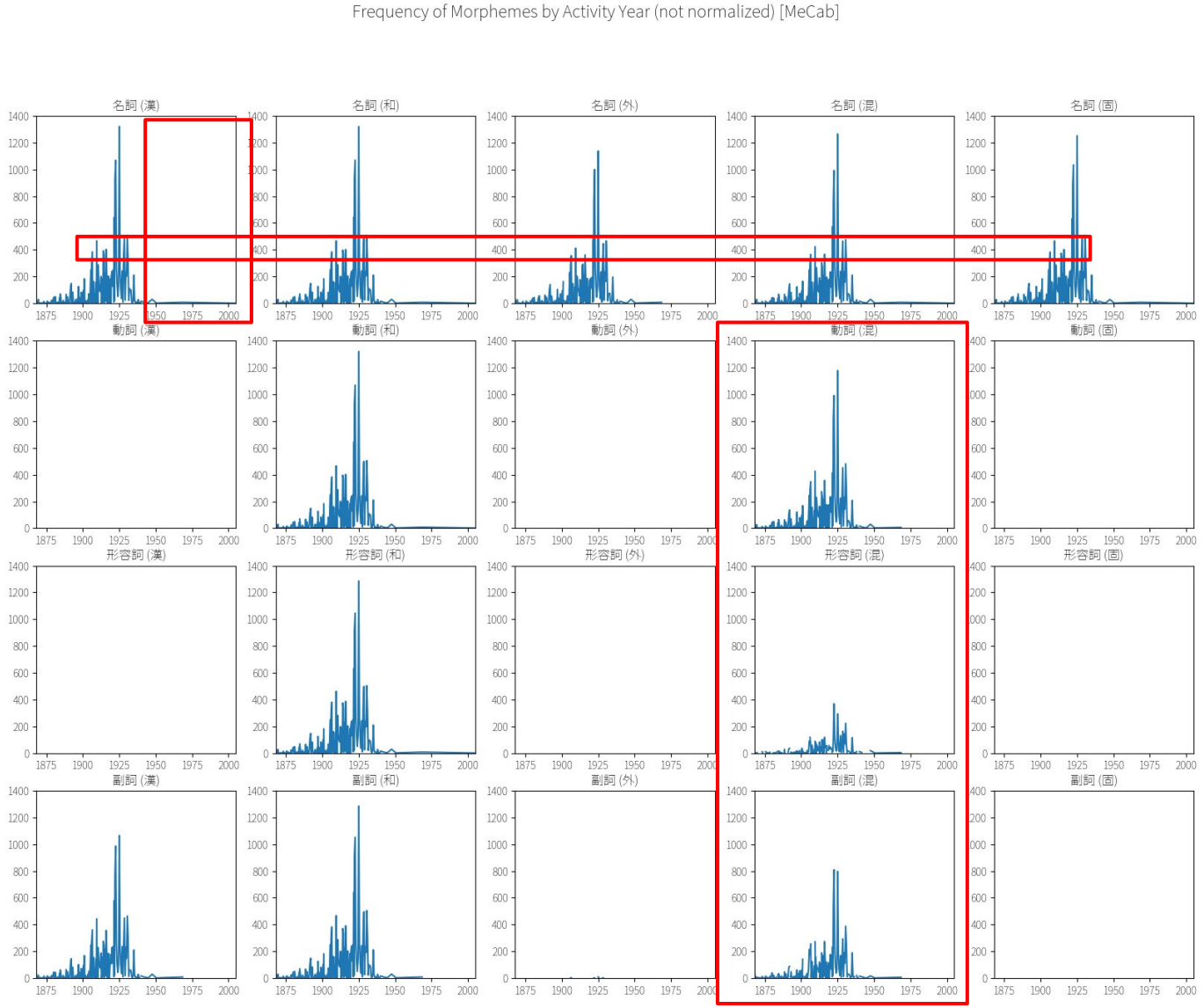
The model is still not perfect …

```
if run_demo:
    surfs = ["簡単", "簡単な", "簡単に"]
    origins = [x["label"] for x in pipe(surfs, batch_size=n_proc)]
    print(f"surfs: {surfs}")
    print(f"origins: {origins}")
```

[46] ✓ 0.0s Python

```
surfs: ['簡単', '簡単な', '簡単に']
origins: ['漢', '和', '和']
```

Plot: frequency



Analysis: frequency

- Axes
  - x = activity year
  - y = frequency (unconditioned)
- Low data availability after 1950 (time to digitize books & post-1968: needs authorization)

> “著作権法改正によって、従来の保護期間である死後50年は死後70年へと延長されました。そのため、たとえば1968年に亡くなった作家の作品がパブリック・ドメインになるのは、2039年の元旦になります。
>
> 今回の保護期間延長が遡って適用されることはありません。そのため、1967（昭和42）年以前に亡くなった作家の著作権は復活しません。” ー 青空文庫

- Verb (mixed), Adj (mixed) are likely to be caused by the final する/な, but we can treat them as SJ & foreign [demo follows]
- For Noun, the frequencies of SJ, native, and fixed are similar, while the frequencies of foreign and mixed are similar but slightly lower
- For mixed Verb, Adj, and Adv, especially for Adj, native is the dominant, and mixed has slightly (Verb) and significantly (Adj, Adv) lower frequencies

Demo: Treat mixed as SJ & foreign

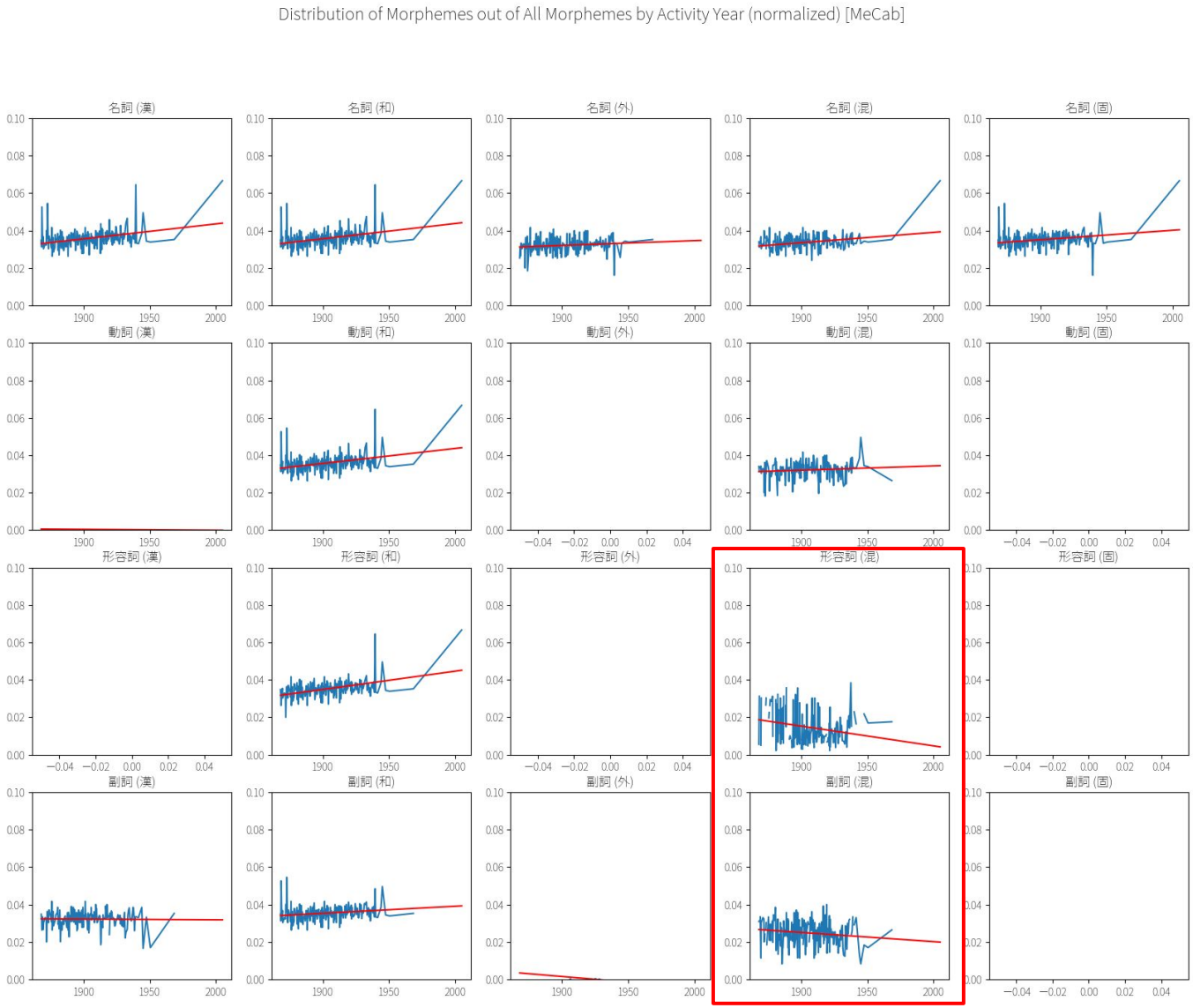
This also reflects from model, learned the UniDic data, so it is due to the labeling of data

```
if run_demo:
    surfs = ["白い", "綺麗な", "頑固な", "頑固", "使う", "学習する", "学習", "ラーニングする"]
    origins = [x["label"] for x in pipe(surfs, batch_size=n_proc)]
    print(f"surfs: {surfs}")
    print(f"origins: {origins}")
```

[37] ✓ 0.1s Python

```
surfs: ['白い', '綺麗な', '頑固な', '頑固', '使う', '学習する', '学習', 'ラーニングする']
origins: ['和', '和', '混', '漢', '和', '混', '漢', '混']
```

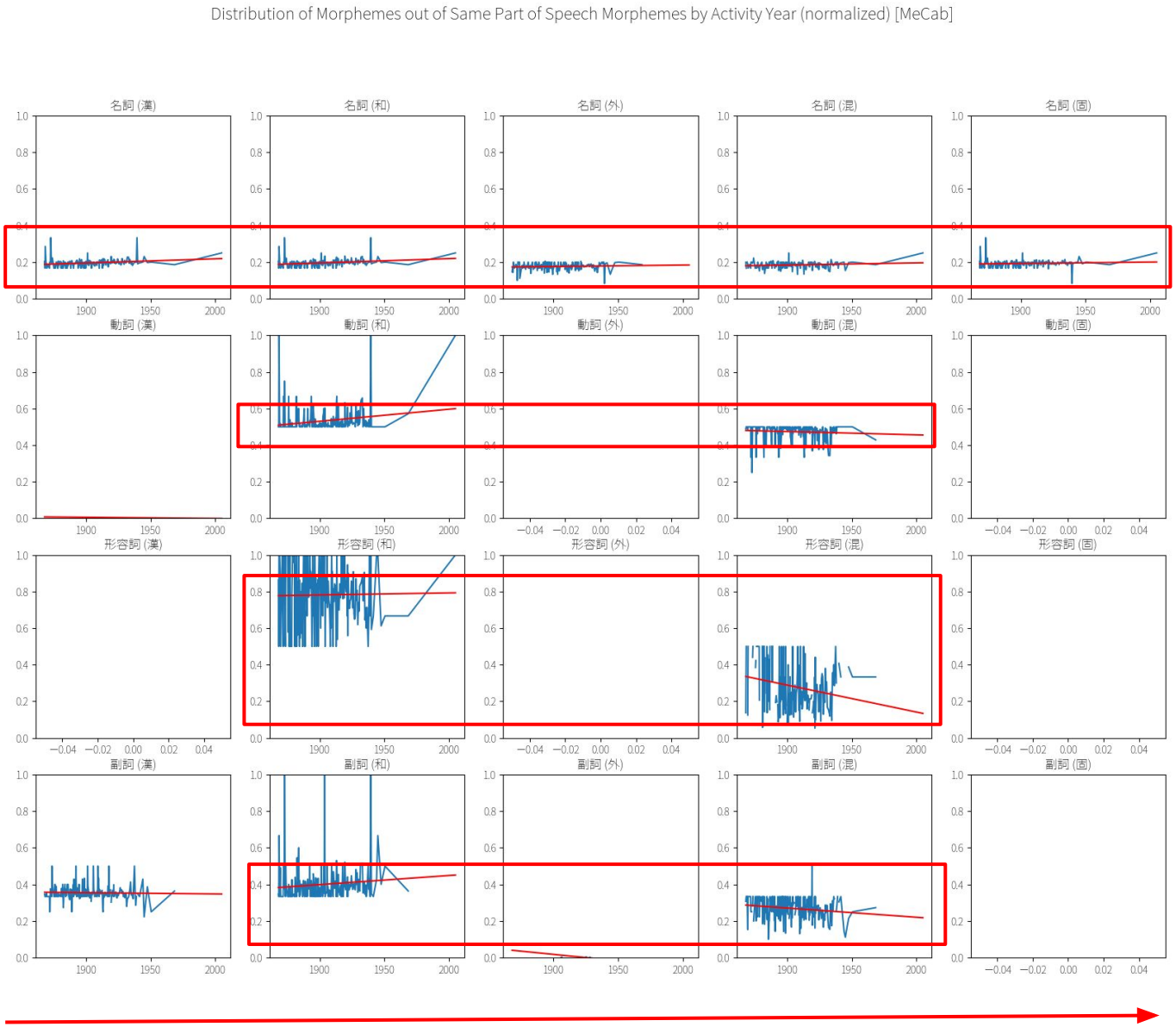
Plot: distribution over all morphemes



Analysis: distribution over all morphemes

- Axes
  - x = activity year
  - y = frequency (conditioned on POS and Origin) / frequency (unconditioned) [proportion]
- To better see the trends, we plot the proportions to alleviate the effect of low data
- Across all words in a year, the proportions of Adj (mixed) and Adv (mixed) are decreasing
- The proportions of all kinds of Noun are increasing
- The proportions of all kinds of native words are increasing
- The proportions of Adv (SJ) decreases a little

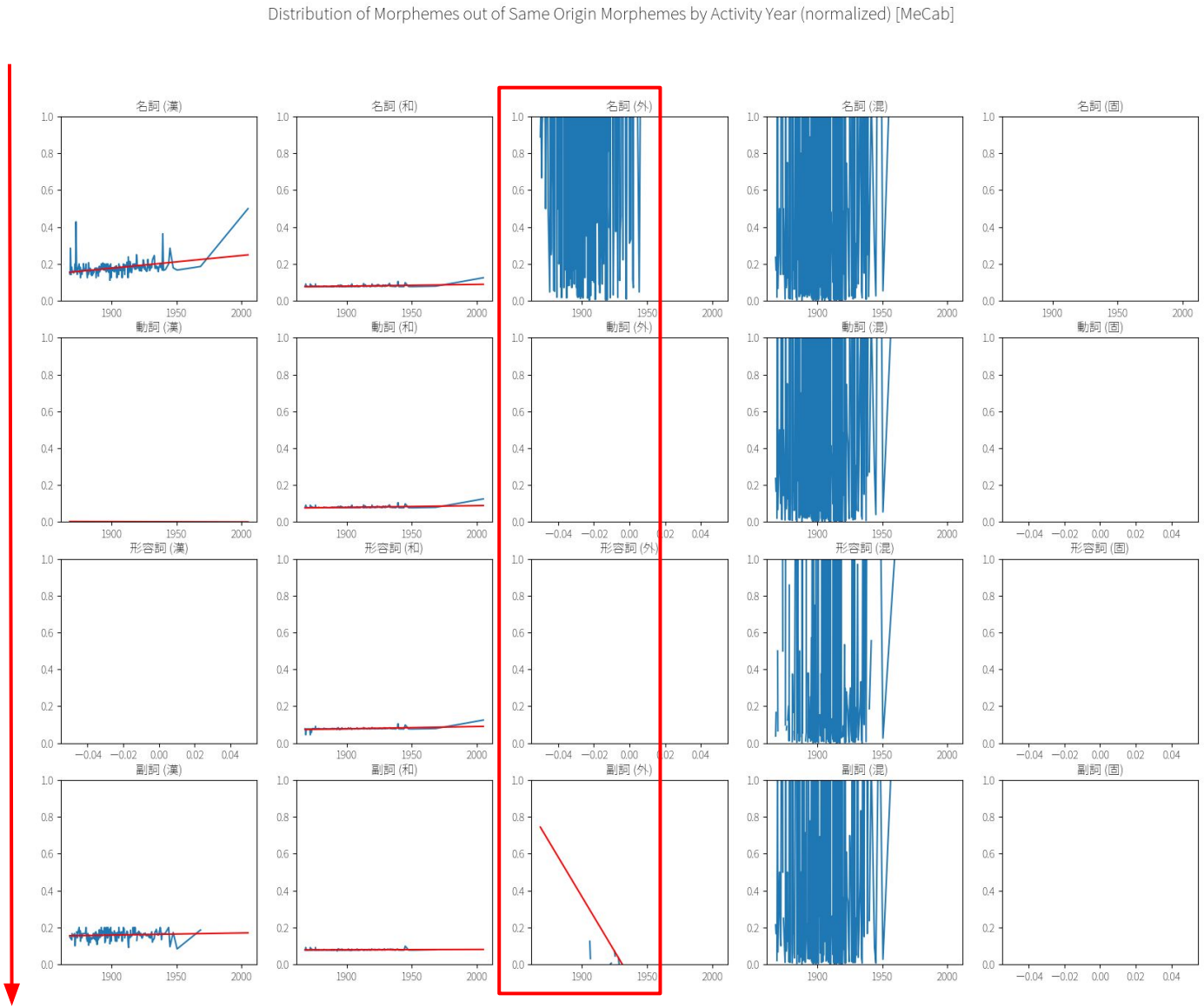
Plot: distribution over part of speech



Analysis: distribution over part of speech

- Axes
  - x = activity year
  - y = frequency (conditioned on POS and Origin) / frequency (conditioned on POS) [proportion]
- To see the distribution of origin specified with a POS, so that we are not affected by the total change of a POS, we plot conditional distributions
- If we see Noun itself, then the distribution is almost constant, so the increase we have just seen is actually the increase of use of Noun without changing conditional distribution of origin
- Inside Verb, there is a 10% increase for native words, and the corresponding 10% decrease for the total of SJ and mixed words, with the more dominance of native words
- Similar trends observed for Adj and Adv, but native words had not has the most dominance for Adv (just gained balance with SJ & mixed recently)

Plot: distribution over origin



Analysis: distribution over origin

- Axes
  - x = activity year
  - y = frequency (conditioned on POS and Origin) / frequency (conditioned on Origin) [proportion]
- The use of pure foreign words is almost only in Noun (not including Verb, Adj), indicating there is no use of foreign words as Adv
- The use of native words is balanced and stable through POS
- The use of pure SJ (not including Verb, Adj) increases in Noun and decreases in Adv
- The use of mixed words oscillates significantly, so we could only interpret that there is varying usage of mixed words across all POS

Project Workload

- Core code: 500+ lines
- Peripheral code: 200+ lines
- 12 hours of fine-tuning on CHTC
- 4 hours of MeCab analysis on my laptop

Future Works / Directions for Expansion

- Run and analyze this corpus on Juman++ (need a lot of time to run with model, much longer than MeCab, so not included in this project)
- Use a larger model / more epochs for a better accuracy
- Split mixed to SJ and foreign (for Verb, Adj) to better analyze distribution over origin (by using a more accurate dictionary?)
- A better method to analyze post-1950 works with fewer data points (with higher granularity? a better data source containing modern texts?)
- Also, the Hasegawa reading does not mention the portion of SJ, native, and foreign words in every additional batch of words, so we can also find the distribution of these words conditioned the top 5000, 7000, and 10000 words (based on Hasegawa's finding on 10000 words to cover 87% of text) [How many words do you want to learn to read a book?]

Credits

- Used CHTC for analysis and fine-tuning.
- aozora_json_scrape: https://github.com/takahashim/aozora_json_scrape
- AozoraTxt: https://github.com/levelevel/AozoraTxt
- Juman++: https://github.com/ku-nlp/jumanpp, https://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN%2B%2B
- rhoknp: https://github.com/ku-nlp/rhoknp
- deberta-v2-base-japanese: https://huggingface.co/ku-nlp/deberta-v2-base-japanese
- 現代書き言葉UniDic: https://clrd.ninjal.ac.jp/unidic/back_number.html
- unidic-py: https://github.com/polm/unidic-py
- mecab-python3: https://pypi.org/project/mecab-python3/