Analysis of Post-Meiji Word Origins in Japanese Literature

An approach in computational linguistics
ASIAN 434, Fall 2023

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

16 December 2023
Version 1

Contents

1 Introduction

In the week 4 Wednesday reading of Hasegawa (2015), pp. 61-74, there are statistics of distribution of SJ, native, and foreign words on magazines over time, and we can reproduce that in the specialized literature area.

By the wide availability of pre-modern public domain literature, we further investigate older texts (before 1956) and a continuum of texts (with data points from almost every year instead of two distinct years in Hasegawa).

One can make such an 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?

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). In one sentence, how many words do you want to learn to read a book? This is an extended question not investigated in this paper, but it can be implemented easily by the code we have so far.

2 Terminologies and Tools

morpheme analyzer splits a line to morphemes, for example, the sentence "日本語を勉強する" should be splitted to "[日本語][を][勉強する]", rather than "[日本][語][を][勉強][する]". we have used MeCab (Kudou n.d.) and Juman++ (Tolmachev, Kawahara, and Kurohashi 2018, Version 2.0.0 RC4) for this task. MeCab uses a dictionary (UniDic (DEN et al. 2007)) 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 encoded in the neural network), but does not have built-in origin data (need to train model to predict origin, or at least use a dictionary). According to 築地 and Shinnou (2021), Juman++ has better accuracy in tokenization task (used here for morpheme segmentation).

POS is the abbrevation used in this paper for part of speech, for example, 名詞 (Noun), 動詞 (Verb), 形容詞 (Adj), and 副詞 (Adv) are used POS in this paper.

Aozora Bunko is an online library for public domain literature. According to its own description (Aozora Bunko n.d.b),

> “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

The **model** we used in this paper does the following job: given $(\mathbf{x}, y)$, where $\mathbf{x}$ is a (tokenized) morpheme and y is the origin of the morpheme (in “和”, “固”, “漢”, “外”, “混”, “記号”, and “NA”; are referred in this paper as “native”, “fixed”, “SJ (Sino-Japanese)”, “foreign”, and “mixed”) as in UniDic; the task predicts $\hat{y}$ (predicted origin of morpheme) from $\mathbf{x}$ which we might not have seen. The model is a function $f : \mathbf{x} \mapsto \hat{y}$. In one sentence, input a morpheme, and output its origin. For example, “会議（かいぎ)” → “漢” (SJ), “戯（たわむ）れる” → “和” (native), and “アップル” → “外” (foreign).

Fine-tuning is the process we modify, instead of use, the model. 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. The following Figure 1 shows the idea of fine-tuning to correct a wrong prediction by exposing the model to a close, correct example.

We can see an example of morpheme analysis for the first sentence of the famous novel *Yukiguni* (Kawabata 1976): “国境の長いトンネルを抜けると雪国であった。” in the following Figure 2 and Figure 3. We can see morphemes and the corresponding part of speech and origin

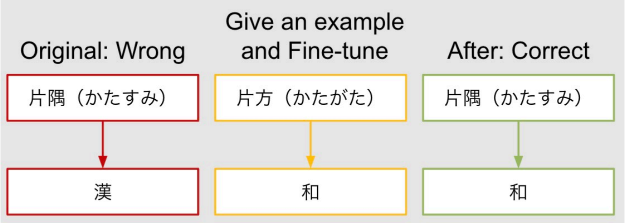
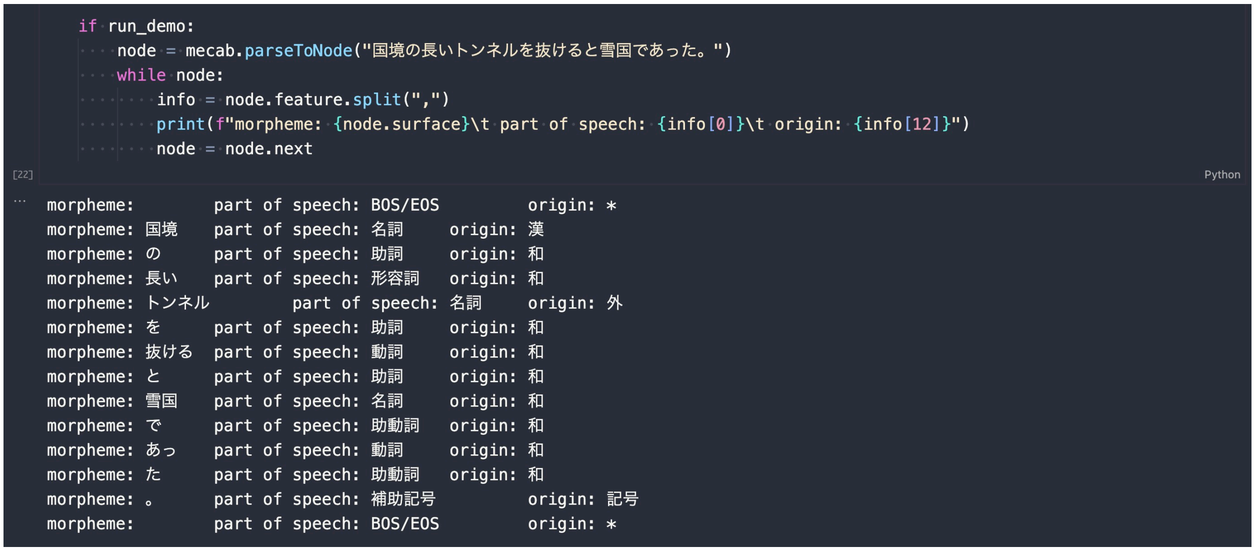


Figure 1: The wrong prediction to SJ is corrected to native

information. The example also represents the better capability of Juman++ in splitting by morphemes instead of words. And from the model, it can even predict “雪国” as fixed word, which is true (better than the dictionary), since it refers to Yuzawa in Niigata Prefecture across the Shimizu tunnel.



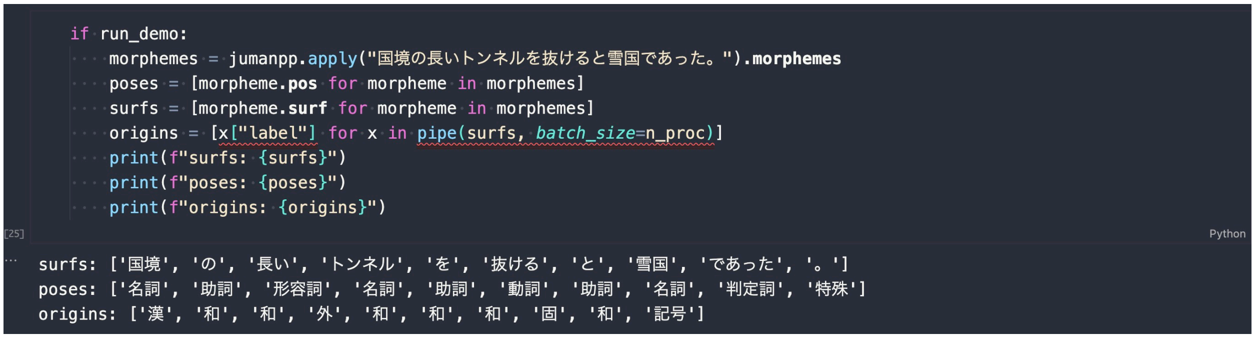
```python
if run_demo:
    node = mecab.parseToNode("国境の長いトンネルを抜けると雪国であった。")
    while node:
        info = node.feature.split(",")
        print(f"morpheme: {node.surface}\t part of speech: {info[0]}\t origin: {info[12]}")
        node = node.next
```

```
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: *
```

Figure 2: MeCab with UniDic, Splitted “であった”

However, the model is not perfect, and it still has inconsistencies. For example, as in Figure 4, the model predicts “簡単な” and “簡単に” as native, but they should be of either SJ or mixed (if count the suffix).

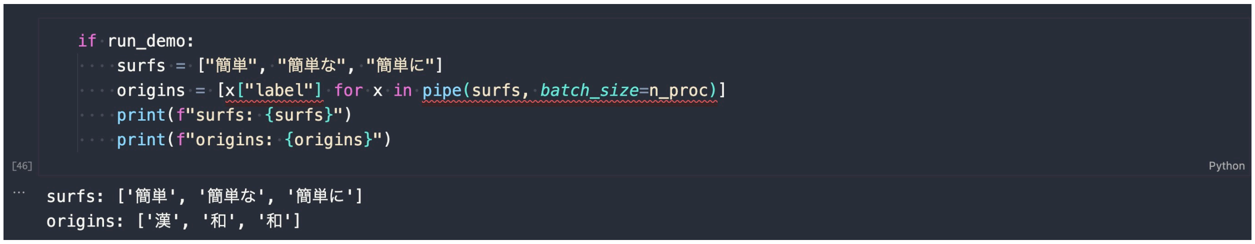
For this paper, the core code is over 500 lines, and the peripheral code is over 200 lines. The code is available upon request.



```
    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]                                                                                  Python
```

```
... 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']
```

Figure 3: Juman++ with Model, Cleverly pick “であった” as a whole



```
    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]                                                                                  Python
```

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

Figure 4: Inconsistency in “簡単”

3 Procedures

3.1 Analysis

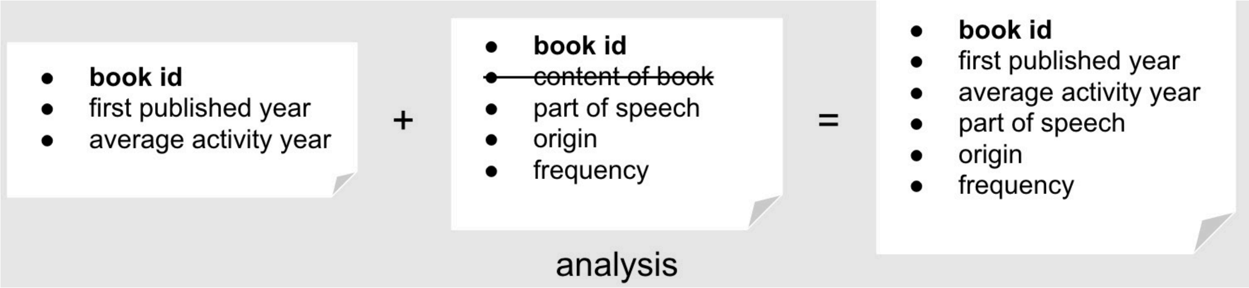


Figure 5: Dataframes in Analysis

As in Figure 5, we mainly run analysis on dataframes for the ease of management. The dataframes are created by the following steps:

1. Create a dataset from `aozora_json_scrape/card.json` (Takahashi 2022), containing the book ID, the book published year, the author IDs, and the average activity year of the authors.

2. Create another dataset from `AozoraTxt/person_utf8/` (EMURA 2023), containing the book ID and the content of the book (by the author ID).
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. Create statistics for content (book ID, part of speech, origin, and frequency).
4. Merge the two datasets by the book ID. Plot and analyze the data by the average activity year of the authors.

For the ease and precision of analysis, we intentionally pick books by the average activity years. The average activity year is just the average of birth and death years. We take only books with average activity year ≥ 1868 (starting of Meiji period), since Japanese has not gone over 言文一致運動 (which phases out the classical grammar), and the classical grammar could cause inaccuracies in models trained and tools developed only on modern grammar. For the authors less than 120 years old without a death year ($2023 -$ birth year ≤ 120 years), we estimate activity year = birth year + 60 years.

We are not using first published year here, because there can be multiple publications of the same book, and the book input in the system might be use the first version, so it can underestimate the age of the book, but if it is correct, it should be a better estimation than our heuristic on average activity year, which we could investigate in the future.

For use of morpheme analyzer, we have the following methods:

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.

The first method needs a lot of time to run Juman++ with model, much longer than MeCab, by the large number of parameters of model and incapability in multiprocessing of the library (Kiyomaru and Ueda 2023) we have used to run Juman++ to fully use CPU. Even with the second method, I run it on my laptop (MacBook Pro 2021, M1 Max CPU) for 4 hours to complete the analysis on the filtered (post-Meiji) dataset.

3.2 Fine-tuning

We are able to fine-tune a classifier to predict the origin of morphemes of (after-train) evaluation accuracy be 0.9162760461217367 out of 87681 samples, by using the output model (checkpoint) at step 21000. The procedure for fine-tuning follows:

1. Create a training/validation/test split dataset (train 80%, validation (in training) 10%, evaluation (after training) 10%) for categorizing morphemes, from UniDic with 876803 items.
2. Fine-tune DeBERTa-v2-base-Japanese (Language Media Processing Lab at Kyoto University 2023) model on the dataset (110 million parameters). The original parameters are pre-trained in masking task. We then adjust, continue to train the parameters to this task of sequence classification for morpheme origins by providing examples in training set. 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).

We have used four NVIDIA A100 40GB GPUs at CHTC for 12 hours to fine-tune the model. The hyperparemeters we used follows:

- train epochs: 3
- batch size: 16
- 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

For some technical details, in Figure 6, we can see the slow convergence of the loss function (on train set) to about 0.15, and the curve starts to oscillate which could indicate overfitting, so we might increase the learning rate, use a larger model, or change our train settings. For example, we can find in-context samples for morphemes (input become "国境の長いトンネルを [抜ける] と雪国であった。" instead of "抜ける" to use the context and conjugations for better prediction). Also, the model could use the output on POS and more from Juman++ which might help the prediction.

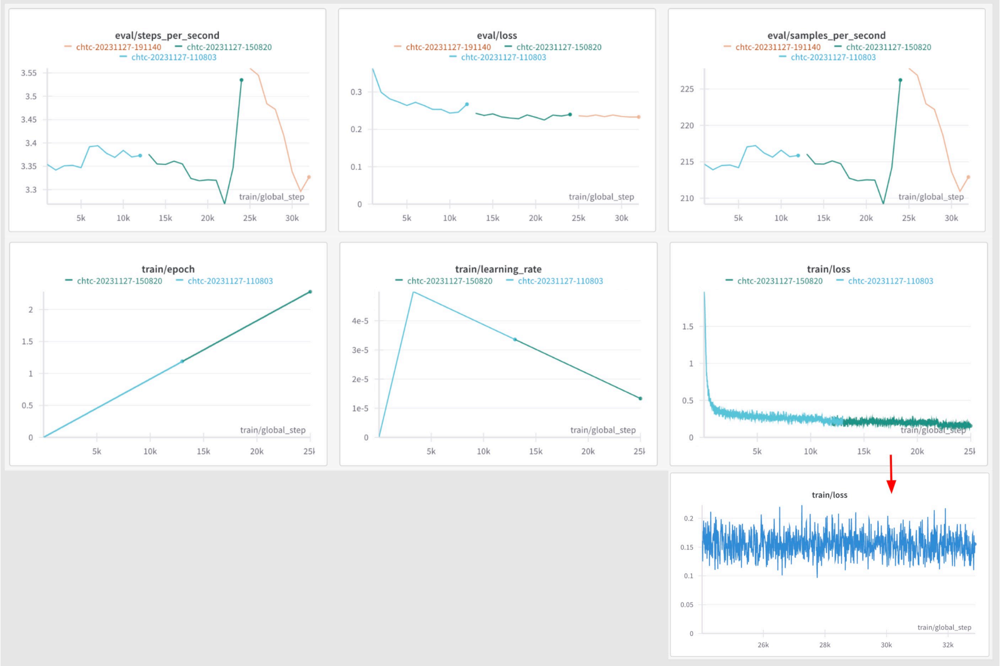


Figure 6: Plots in the fine-tuning process

4 Plots and Analysis

By applying MeCab and UniDic on the dataset, we have obtained the following plots, and we analyze them in the following subsections.

4.1 Frequency

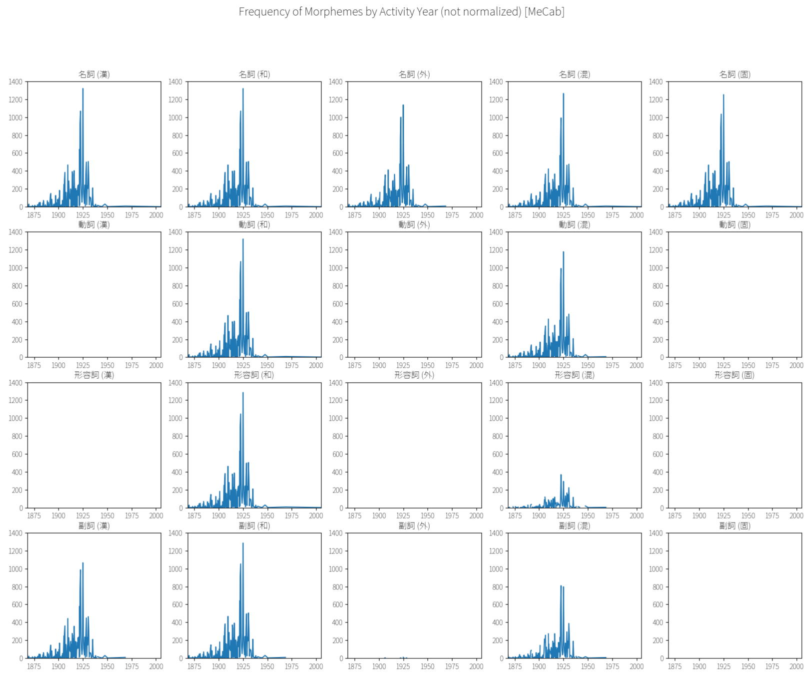


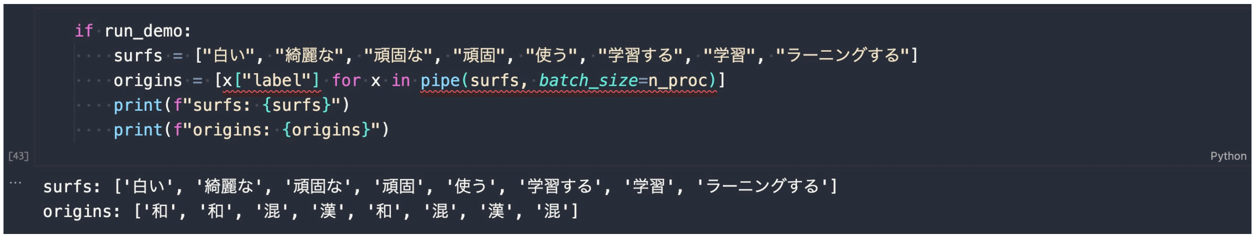
Figure 7: frequency

In Figure 7, the horizontal axis is activity year, and the vertical axis is frequency (unconditioned). We found the low data availability after 1950, and this could be caused by time to digitize books and the copyright protection for post-1968 books, as described (Aozora Bunko n.d.a):

> “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

As a future task, there could be a better method to analyze post-1950 works with fewer data points. For example, with higher granularity, or a better data source containing more modern texts in Heisei period.

Verb (mixed), Adj (mixed) are likely to be caused by the suffixes (like する/な), but we can treat them as SJ and foreign. This could be reflected even in the model trained on the data, which might also misleads the model prediction, as in Figure 8. To better analyze the distribution, we should split mixed to SJ and foreign (for Verb and Adj), which can also be from using a more accurate dictionary.



```
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}")
```

```
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']
```

Figure 8: Once we add the suffix, the prediction becomes mixed.

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 (for Verb) and significantly (for Adj and Adv) lower frequencies. We will see this more significantly in later plots.

4.2 Distribution over All Morphemes

In Figure 9, the horizontal axis is activity year, and the vertical axis is frequency (conditioned on POS and Origin) / frequency (unconditioned). To better see the trends, we plot the proportions to alleviate the effect of low data.

Also in the figure, we can see that 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.

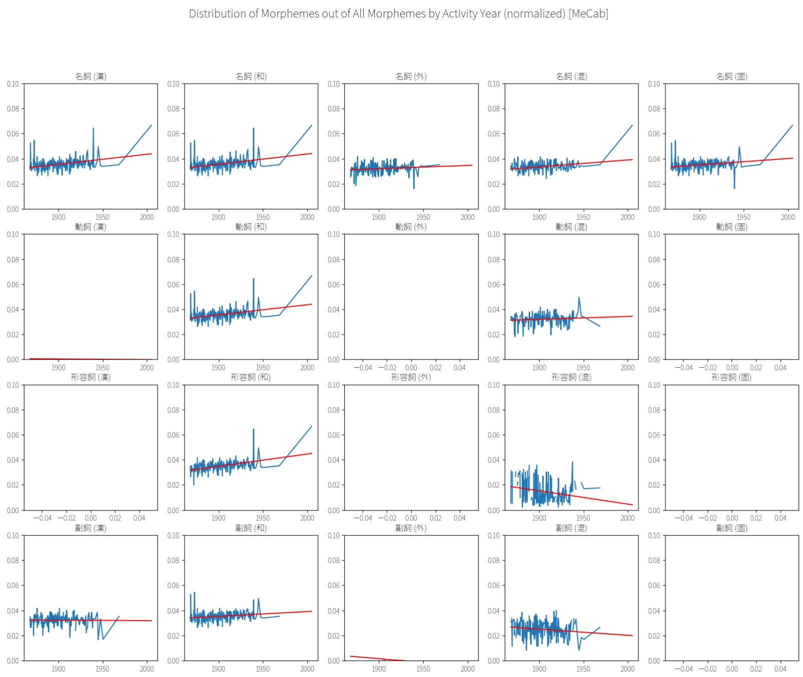


Figure 9: distribution over all morphemes

One might think, despite the two decreases, the remaining blocks are all increasing, which does not reveal much information.

4.3 Distribution over POS

In Figure 10, the horizontal axis is activity year, and the vertical axis is frequency (conditioned on POS and Origin) / frequency (conditioned on POS). 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. In this way, we should treat every row as a group (aggregated to 1 within a group).

If we see Noun itself, then the distribution is almost constant, so the increase we have just seen in Figure 9 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

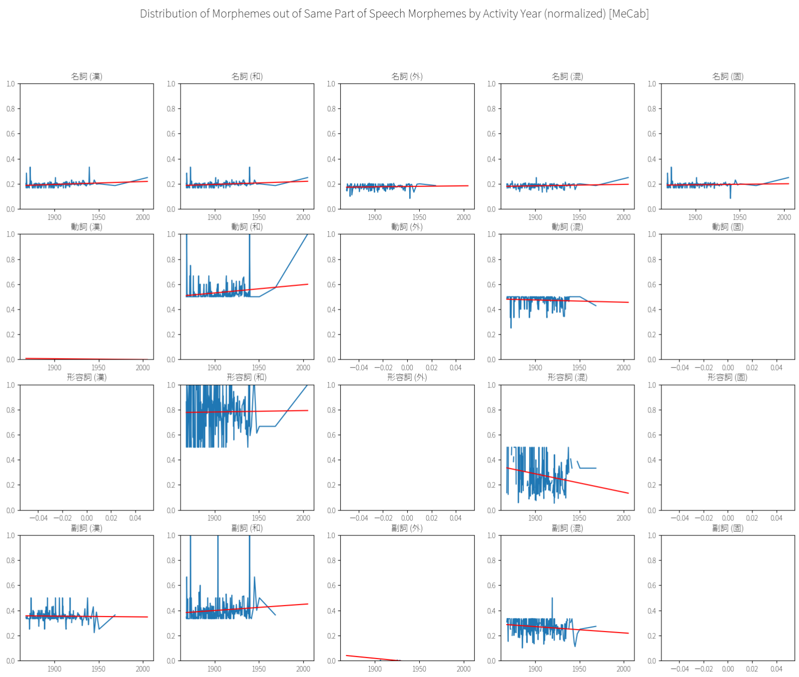


Figure 10: distribution over POS

for the total of SJ and mixed words, with the more dominance of native words.

Similar trends are observed for Adj and Adv, but native words had not has the most dominance for Adv (it just started to gain the balance with SJ & mixed recently).

4.4 Distribution over Origins

In Figure 11, the horizontal axis is activity year, and the vertical axis is frequency (conditioned on POS and Origin) / frequency (conditioned on Origin). In this way, we should treat every column as a group.

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.

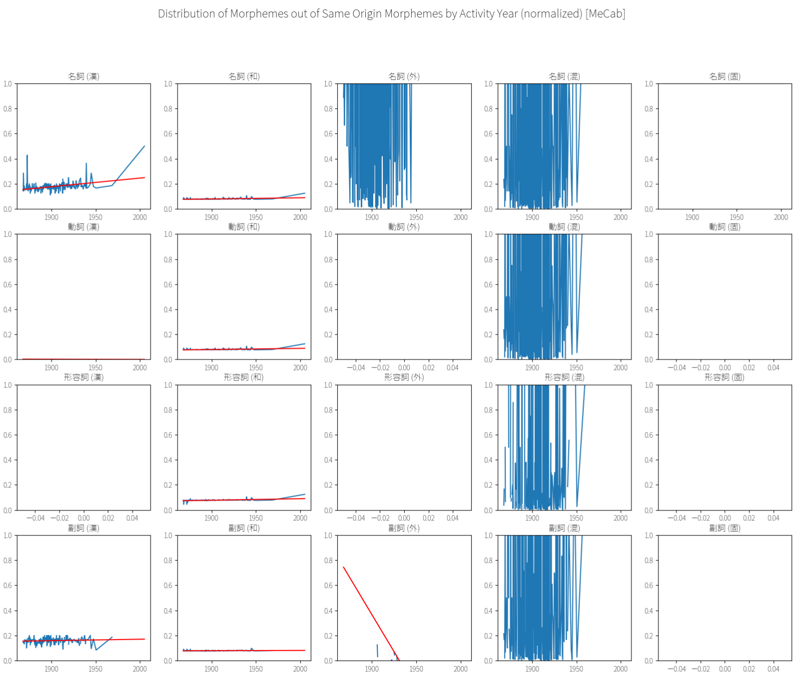


Figure 11: distribution over origins

5 Conclusion

In this paper, we have analyzed the distribution of origins of morphemes, and proposed two methods for such analysis. We back to the assumption in the introduction that the usage of native and SJ words decrease over year, but from the findings, the answer is No, for native words, and we can even see some preferences toward native words. This might be caused the lack of data as we have discussed in the analysis section, but since we put the same weight on every year, the conclusion should still be valid, while it still left to us to investigate further in the “mixed” cattegory.

6 References

Used CHTC at University of Wisconsin–Madison for analysis and fine-tuning.

Aozora Bunko. n.d.a. “青空文庫　著作権保護期間延長になった作家名一覧.” Accessed

December 14, 2023. https://www.aozora.gr.jp/shiryo_pdlocked.html.

———. n.d.b. “青空文庫編青空文庫早わかり.” Accessed December 14, 2023. https://www.aozora.gr.jp/guide/aozora_bunko_hayawakari.html.

DEN, Yasuharu, Toshinobu OGISO, Hideki OGURA, Atsushi YAMADA, Nobuaki MINEMATSU, Kiyotaka UCHIMOTO, and Hanae KOISO. 2007. “コーパス日本語学のための言語資源: 形態素解析用電子化辞書の開発とその応用.”国書刊行会. https://doi.org/10.15084/00002185.

EMURA, Hideyuki. 2023. “Levelevel/AozoraTxt: 青空文庫のテキストファイル.” https://github.com/levelevel/AozoraTxt/tree/master.

Hasegawa, Yoko, 1950- author. 2015. *Japanese : A Linguistic Introduction.* Cambridge : Cambridge University Press, 2015. https://search.library.wisc.edu/catalog/9913899080302121.

Kawabata, Yasunari. 1976. *Yukiguni* 雪国. Iwanami Bunko 岩波文庫. Iwanami Shoten 岩波書店. https://www.google.com/books/edition/%E9%9B%AA%E5%9B%BD/JYRAEWsc16UC.

Kiyomaru, Hirokazu, and Nobuhiro Ueda. 2023. “Rhoknp: Yet Another Python Binding for Juman++/KNP/KWJA.” https://github.com/ku-nlp/rhoknp.

Kudou, Taku. n.d. “MeCab: Yet Another Part-of-Speech and Morphological Analyzer.” Accessed December 14, 2023. https://taku910.github.io/mecab/.

Language Media Processing Lab at Kyoto University. 2023. “Ku-Nlp/Deberta-V2-Large-Japanese ·Hugging Face.” https://huggingface.co/ku-nlp/deberta-v2-large-japanese.

Takahashi, Masayoshi. 2022. “Takahashim/Aozora_json_scrape.” https://github.com/takahashim/aozora_json_scrape.

Tolmachev, Arseny, Daisuke Kawahara, and Sadao Kurohashi. 2018. “Juman++: A Morphological Analysis Toolkit for Scriptio Continua.” In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, edited by Eduardo Blanco and Wei Lu, 54–59. Brussels, Belgium: Association for Computational

Linguistics. https://doi.org/10.18653/v1/D18-2010.

築地俊平, and Hiroyuki Shinnou. 2021. “Tokenizerの違いによる日本語BERTモデルの性能評価.” In *言語処理学会第27回年次大会発表論文集*.