Just One is Enough: An Existence-based Alignment Check for Robust Japanese Pronunciation Estimation

Hayate Nakano and Nobuhiro Kaji
LY Corporation
{hanakano,nkaji}@lycorp.co.jp

Abstract

Neural models for Japanese pronunciation estimation often suffer from errors such as hallucinations (generating pronunciations that are not grounded in the input) and omissions (skipping parts of the input). Although attention-based alignment has been used to detect such errors, selecting reliable attention heads is difficult, and developing methods that can both detect and correct these errors remains challenging.

In this paper, we propose a simple method called *existence-based alignment check*. In this approach, we consider alignment candidates independently extracted from all attention heads, and check whether at least one of these candidates satisfies two conditions derived from the linguistic properties of Japanese pronunciation: monotonicity and pronunciation length per character. We generate multiple hypotheses using beam search and use the alignment check as a filtering mechanism to correct hallucinations and omissions.

We apply this method to a dataset of Japanese facility names and demonstrate that it improves pronunciation estimation accuracy by over 2.5%.

1 Introduction

The task of estimating the pronunciation for a given text is crucial in Japanese NLP. This task is commonly formulated as translating text into phonetic *kana* character strings. This is known as a challenging task because the Japanese writing system intermixes ideographic kanji, alphanumeric characters, and phonetic kana characters (Hatori and Suzuki, 2011). Accurate pronunciation estimation is important in various Japanese NLP applications including information retrieval and text-to-speech synthesis. Hereafter, pronunciation refers to its representation in kana.

Neural models, notably Transformers (Vaswani et al., 2017), are now widely used for this task (Jones et al., 2023). However, neural machine translation is known to suffer from problems such as *hallucination* (generating content unrelated to the source text) and *omission* (failing to generate content for parts of the source text) (Tu et al., 2016; Ji et al., 2023). These problems are particularly prevalent when training data resources are insufficient or contain significant noise (Raunak et al., 2021).

A straightforward way to spot such an error is to inspect the correspondence between source and target tokens, i.e., the alignment. Because Transformer-based sequence-to-sequence models employ cross-attention, interpreting the attention weights as a *soft* alignment and using them for detection seems appealing. This strategy, however, faces two fundamental challenges.

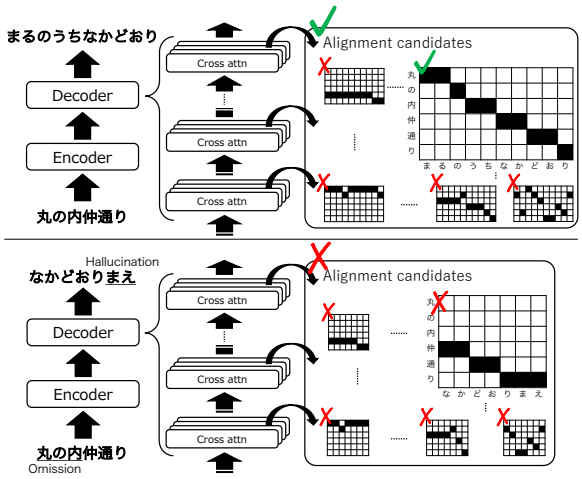


Figure 1: Illustration of pronunciation estimation for the Japanese phrase 丸の内仲通り read as まる/の/うち/なか/どお/り *Marunouchi Nakadori*. When the model correctly predicts the pronunciation (top), at least one valid alignment exists among the alignment candidates (indicated by a ✓). In contrast, if no such valid candidate is found (bottom), the output is judged to contain hallucinations or omissions.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 584–594
November 4-9, 2025

First, selecting which attention head to inspect is non-trivial. Alignment information is known to concentrate in only a few heads within the Transformer's multi-layer, multi-head attention (Kobayashi et al., 2020). Determining where those heads are typically requires extra reference data or computational overhead (Ferrando and Costa-jussà, 2021; Wang et al., 2024). Second, although many prior studies have investigated error detection based on attention weights (Lee et al., 2018; Berard et al., 2019; Raunak et al., 2021; Ferrando et al., 2022a; Guerreiro et al., 2023b), few have explored methods for performing actual correction (Guerreiro et al., 2023c; Dale et al., 2023a).

In this paper, we propose a simple and training-free method that detects and corrects hallucinations and omissions in Japanese pronunciation estimation, using only attention weights.

The key idea is to formulate a concise yet strict condition for *the valid alignment*, which would be obtained when the model output is free of hallucinations and omissions, based on the linguistic properties of Japanese pronunciation. Based on this idea, we check whether the alignment candidate derived from the weights of each attention head satisfies the validity condition. If and only if at least one candidate satisfies the condition, the model output is considered to be free of hallucinations and omissions, as illustrated in fig. 1.

In this approach, we consider all heads simultaneously, so we do not need to identify specific heads in advance. Moreover, by using the existence check as a filter, we are able to correct hallucinations and omissions.

We empirically investigated the effectiveness of our method with its application to a map app provided by our company in mind. Specifically, we created a benchmark dataset for pronunciation of facility names registered in our in-house database, and evaluated the accuracy of the pronunciation estimation task. The experimental results demonstrated that our method successfully improved estimation accuracy by more than 2.5%.

The remainder of this paper is organized as follows. Section 2 provides background on Japanese pronunciation estimation. Section 3 introduces our proposed method to detect and correct hallucinations and omissions. Sections 4 and 5 detail our experimental setup and present an in-depth analysis of the results. Section 6 discusses related work, and Section 7 concludes the paper with suggestions for future work. Section 8 discusses the limitations of our work.

2 Characters and Pronunciations in Japanese

The primary characters used in Japanese are kana and kanji. In addition, alphanumeric characters and various typographical symbols are also used.

Most kanji characters have multiple pronunciations expressed in kana, and the specific pronunciation used depends on the context. Although kanji characters and their pronunciations are numerous and diverse, we focus on one property common to them all: the length of each pronunciation. For an individual kanji, the length typically ranges from one to four. Our survey of entries in UniDic (Den et al., 2007) shows that pronunciations longer than four kana occur in only about 0.1% of cases, and most of these belong to infrequently used words.

Alphanumeric characters are often intermixed within Japanese text. These non-Japanese characters are also frequently assigned kana-based pronunciations through transliteration (Knight and Graehl, 1998), adapting their original pronunciations to the Japanese phonetic system. Unlike kanji, lengths of pronunciations for these cases cannot be defined per character, but the ratio of word length to kana length can be measured. This ratio typically falls between zero and five. An analysis on the English-to-Katakana dataset by Merhav and Ash (2018) showed that fewer than 0.001% of the entries violated this condition.

3 Method

This section presents the proposed method. In section 3.1, we briefly summarize Transformer models for pronunciation estimation. Section 3.2 explains how to extract alignment candidates from attention weights, and section 3.3 presents the proposed existence-based alignment check. Section 3.4 discusses a filtering-based approach to correcting hallucinations and omissions by using the alignment check. Section 3.5 discusses alternative approaches to extract alignment candidates.

3.1 Pronunciation Estimation Model

We define pronunciation estimation as a sequence-to-sequence generation task: transducing Japanese character sequences into kana sequences. For this task, we employ a Transformer model (Vaswani et al., 2017).

We explore character-based, rather than subword-based, Transformer models because our method utilizes conditions of valid alignment (Section 3.3), which are naturally defined at the character level (not at subword level). Let s and t denote the lengths of the source and target sequences, respectively. The token indices for the source and target satisfy $i \in \{1, \ldots, s\}$ and $j \in \{0, \ldots, t+1\}$, where $j = 0$ and $j = t+1$ correspond to the `BOS` and `EOS` tokens, respectively.

The decoding process is autoregressive. At the j-th step, the decoder takes the $(j-1)$-th target token as input to predict the j-th target token. Let $l \in \{1, \ldots, L\}$ and $h \in \{1, \ldots, H\}$ be the indices for the decoder layer and head. The cross-attention weight is denoted by $\alpha_{ji}^{(l,h)}$, which represents the contribution of the source token at position i to the prediction of the target token at position j.

3.2 Alignment Candidates from Attention Weight

We define an alignment as a sequence $\mathbf{a} = (a_1, ..., a_t)$, where each $a_j \in \{1, \ldots, s\}$ represents the source position that contributes most to the j-th target token. Based on the cross-attention weights, we obtain an alignment candidate from each layer l and head h as follows:

$$a_{j-1}^{(l,h)} = \arg\max_i \alpha_{ji}^{(l,h)}. \quad (1)$$

The subscript $j-1$ on the left-hand side signifies that the alignment is calculated with a focus on the *input* token for the j-th decoding step. Therefore, this method is referred to as **Alignment with Input (AWI)**. In contrast, when the left-hand side of eq. (1) is changed to $a_j^{(l,h)}$, it corresponds to the *output* token and the method is called **Alignment with Output (AWO)**.

3.3 Validity Condition and Existence-Based Alignment Check

Next, we formulate the validity condition $C(\mathbf{a})$ by focusing on the linguistic properties of pronunciations. Specifically, we impose two key constraints: **monotonicity** and **length**.

The monotonicity constraint refers to the property that the source character sequence is read sequentially. This is given by:

$$M(\mathbf{a}) : a_1 \leq \cdots \leq a_t.$$

The length constraint refers to the possible length of target (i.e., kana) character sequence corresponding to a single source character. Let $c_i \in \{\texttt{kanji}, \texttt{kana}, \ldots\}$ be the character type at source position i and $Y_c \subset \mathbb{N}$ be a set of possible lengths of target character sequence corresponding to a source character of type c. Based on the discussion in section 2, we adopt the sets for Y_c in table 1. Then, the length constraint is given by:

c	Y_c
Kana	$\{1\}$
Kanji	$\{1, 2, 3, 4\}$
Alphanumeric	$\{0, 1, 2, 3, 4, 5\}$
Typographical Symbols	$\{0\}$

Table 1: Pronunciation length of Japanese characters.

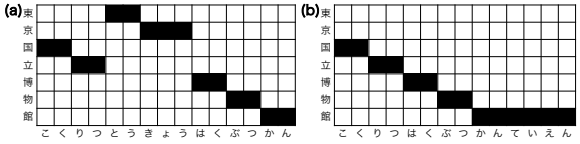


Figure 2: Invalid alignment examples for the Japanese facility name 東京国立博物館 (Tokyo National Museum) read as とう/きょう/こく/りつ/はく/ぶつ/かん *Tokyo Kokuritsu Hakubutsukan*. (a) violates the monotonicity condition M. (b) violates the length condition R. No pronunciations for "国立" indicate an omission, and excessively long pronunciation for "館" indicate a hallucination.

$$R(\mathbf{a}) : \forall i,\ \#\{j \mid a_j = i\} \in Y_{c_i}.$$

In this work, we use the validity condition defined by the logical AND of the two constraints:

$$C(\mathbf{a}) = M(\mathbf{a}) \wedge R(\mathbf{a}). \quad (2)$$

Figure 2 provides illustrative examples demonstrating that outputs affected by hallucination or omission errors violate the constraints.

Now, we define the set of all alignment candidates $\mathcal{A}$ as:

$$\mathcal{A} = \left\{ \mathbf{a}^{(l,h)} \middle| l \in \{1, \ldots, L\}, h \in \{1, \ldots, H\} \right\}.$$

Our core proposal is to perform an **existence-based alignment check**: we judge the model output to be valid if there exists at least one candidate in $\mathcal{A}$ that satisfies the validity condition:

$$\exists \mathbf{a} \in \mathcal{A} \text{ s.t. } C(\mathbf{a}) \\ \Rightarrow \text{Free of Hallucinations and Omissions.} \quad (3)$$

3.4 Correction of Hallucinations and Omissions

Based on the criteria defined in eq. (3), we propose a filtering-based method for correcting hallucinations and omissions.

1. Output k pronunciation candidates using beam search.
2. For each pronunciation candidate, check whether it satisfies the criteria.
3. Among the valid candidates, select the one with the highest model score as the output.
4. If none of the candidates are valid, select the original output of the model.

This fallback step (Step 4) is designed to avoid a loss in performance when our filtering process fails to find any valid candidates.

3.5 Alternative Approaches of Alignment Candidate Extraction

In this study, alignment candidates are defined as in eq. (1); however, various other definitions have been proposed in prior work. In our experiments, we also evaluated several alternative definitions, as described below.

Weight-based vs. Norm-based

Excluding bias terms, the cross-attention output $\mathbf{y}_j$ can be expressed as $\mathbf{y}_j = \sum_{i,h} \alpha_{ji}^{(l,h)} \mathbf{f}_i$ using the attention weights $\alpha_{ji}^{(l,h)}$ and the transformed vectors $\mathbf{f}_i$. Kobayashi et al. (2020) proposed a method to obtain the alignment candidate using the norm of the weighted vector $\|\alpha_{ji}^{(l,h)} \mathbf{f}_i\|$, instead of the raw attention weight α_{ji}. While the original method is referred to as the **weight-based** approach, this method is called the **norm-based** approach.

Head-wise vs. Integration-per-Layer

The method described in section 3.2, which generates candidates for each head, is a **head-wise** approach. An alternative is to integrate the contributions from all heads within the same layer to obtain a single candidate per layer, which we call an **integration-per-layer** approach. For the weight-based method, this can be done by summing the weights $\sum_h \alpha_{ji}^{(l,h)}$[1]. For the norm-based method, by taking the norm of the sum of weighted vectors $\| \sum_h \alpha_{ji}^{(l,h)} \mathbf{f}_i^{(l,h)} \|$ (Li et al., 2019).

[1]This is the same as the averaged attention weights that PyTorch's `MultiHeadAttention` module outputs by default, except for a constant factor.

4 Experimental Setting

4.1 Data

We manage information on geographic entities—such as names, addresses, and coordinates—for use in our services. From this, pairs of facility names and their pronunciations were obtained and used as training data. The average length of the facility names in the training data was 10 characters, and the average pronunciation length was 15 kana characters. The dataset was divided into training (~5 million items), validation (~10,000 items), and test sets (~1,000 items). We manually re-annotated the pronunciations for the test data to ensure the data quality.

4.2 Model

As described in section 3, we employ a standard encoder-decoder Transformer model in conjunction with a character-level tokenizer.

The number of alignment candidates obtained from attention weights is $L \times H$, where L and H denote the number of decoder layers and attention heads, respectively. To investigate the relationship between the number of alignment candidates and accuracy of our method, we explored various configurations of L and H while fixing the total number of layers (encoder + decoder) to 12 and the embedding dimension d to 512 (thus keeping the total number of parameters approximately constant). Specifically, we varied the number of decoder layers $L \in \{1, 2, 3, 6\}$ and the number of attention heads $H \in \{4, 8, 16\}$. To ensure robustness of our results, each model configuration was trained three times with different random seeds. For other parameters, please refer to appendix A.

5 Results and Discussion

5.1 Accuracy Gains with Filtering

Table 2 shows the accuracy and mean F-score (Chen et al., 2018), obtained after applying our filtering-based correction method, across various combinations of decoder layers L, attention heads H and the number of candidates k. Our method consistently improves accuracy in most configurations. The most significant gain is observed when $L = 3, H = 16$, and $k = 16$, where the accuracy improves by over 2.5% compared to the baseline

		L											
		1				2				3				6			
H	k	Acc	Δ	F	Δ	Acc	Δ	F	Δ	Acc	Δ	F	Δ	Acc	Δ	F	Δ
4	4	79.63	-1.34	96.27	-0.06	83.37	+1.30	97.03	+0.44	83.40	+1.30	97.02	+0.45	83.27	+1.54	96.97	+0.49
	16	77.73	-3.24	96.10	-0.23	83.93	+1.86	97.16	+0.57	83.57	+1.47	97.10	+0.53	83.67	+1.94	97.10	+0.62
	baseline	80.97		96.33		82.07		96.59		82.10		96.57		81.73		96.48	
8	4	82.60	+1.00	96.86	+0.41	83.20	+1.03	97.00	+0.34	83.43	+1.76	97.07	+0.47	83.43	+1.80	97.05	+0.52
	16	82.67	+1.07	96.94	+0.49	83.93	+1.76	97.17	+0.51	84.13	+2.46	97.24	+0.64	84.23	**+2.60**	97.22	**+0.69**
	baseline	81.60		96.45		82.17		96.66		81.67		96.60		81.63		96.53	
16	4	82.10	+0.70	96.83	+0.31	83.13	+1.46	97.04	+0.47	83.70	+1.53	97.12	+0.44	83.60	+1.83	97.03	+0.53
	16	82.03	+0.63	96.88	+0.36	83.63	+1.96	97.18	+0.61	**84.70**	+2.53	**97.34**	+0.66	84.13	+2.36	97.14	+0.64
	baseline	81.40		96.52		81.67		96.57		82.17		96.68		81.77		96.50	

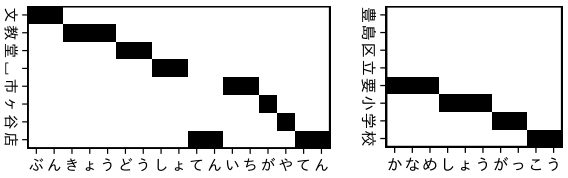
Table 2: Results after correction. The "k" column indicates the number of candidates, as defined in section 3.4. The "baseline" row shows the original output of the model (top-1), without any correction. The "Acc" and "F" columns indicate accuracy and mean F-score, respectively. The "Δ" column shows the difference from the base performance.

(no correction), achieving the highest final score among all settings. This configuration was used in all subsequent analyses.

In our method, the quality of output candidates is a critical factor for accuracy improvement. We estimate the theoretical upper bound of improvement by computing the difference between top-k accuracy and top-1 accuracy. For $k = 4$, this value is 10.90%; for $k = 16$, it is 13.96%, indicating a difference of over 3%. As our method specifically targets hallucinations and omissions, the actual gain is typically limited to about 15–20% of the theoretical maximum. Additional top-k accuracy values are provided in appendix B for reference.

One potential drawback of our filtering approach lies in cases where the top-1 output is correct but fails to satisfy the criteria, while a lower-ranked one does satisfy it. In such cases, filtering degrades the output quality. Although the frequency of degradation cases increases with the candidate size k, the growth remains sufficiently small: 0.23% of samples for $k = 4$, and 0.37% for $k = 16$. The increase in improvements with k surpasses the corresponding rise in degradation: 1.77% of samples for $k = 4$, and 2.90% of samples for $k = 16$.

We present a qualitative analysis of the output changes for a model with a specific random seed under the $L = 3$, $H = 16$, and $k = 16$ setting. First, it is important to note that about 18% of the errors in the baseline output (i.e., the top-1 output of the model) include not only hallucinations and omissions but also transliteration errors, failures in disambiguating kanji pronunciation, and combinations of these. Because the contextual information available from facility names is limited,



(a) Hallucination (b) Omission

Figure 3: Examples of alignment candidates in the presence of a hallucination and an omission.

disambiguation failures are common. These errors are inherently difficult to resolve without introducing some form of external knowledge. For example, the kanji sequence 山和 in a facility name can be read as either さんわ *Sanwa* or やまわ *Yamawa*, and it is extremely difficult to determine the correct reading of 山和 for an unknown facility name.

In the outputs we analyzed, our filtering method achieved a 1.4% accuracy improvement by correcting hallucinations and a 1.5% improvement by correcting omissions. Conversely, filtering degraded performance in 0.4% of the samples. Overall, this resulted in a net improvement of 2.5%.

We present illustrative examples of improvements below. Owing to contractual restrictions, the examples are not taken directly from the original dataset. Instead, they were created by referencing publicly available facility names on the web that resemble the confirmed cases.

As an example of a corrected hallucination, the baseline model predicted the pronunciation ぶんきょうどうしょてんいちがやてん for the facility name 文教堂 市ヶ谷店 correctly read as ぶんきょうどういちがやてん *Bunkyodo Ichigayaten*. Figure 3a shows one of the alignment candidates for

Method	Weight-based		Norm-based	
	Acc	F	Acc	F
Head-wise				
$\mathcal{A}^{\text{AWI}}$	**84.70**	**97.34**	**84.73**	**97.34**
$\mathcal{A}^{\text{AWO}}$	82.00	96.95	82.03	96.99
$\mathcal{A}^{\text{AWI}} \cup \mathcal{A}^{\text{AWO}}$	**84.70**	**97.34**	84.67	**97.34**
Integrated-per-Layer				
$\mathcal{A}^{\text{AWI}}$	80.90	96.69	77.63	96.04
$\mathcal{A}^{\text{AWO}}$	81.97	96.89	80.53	96.56
$\mathcal{A}^{\text{AWI}} \cup \mathcal{A}^{\text{AWO}}$	83.67	97.18	81.83	96.84
ALTI	79.73	96.40		
baseline	82.17	96.68		

Table 3: Comparison of alignment candidate extraction methods. The "Acc" and "F" columns indicate accuracy and mean F-score after filtering, respectively.

this incorrect output. Although the alignment is correct and satisfies the validity condition $C(\mathbf{a})$ except for the しょてん part, this hallucinated segment violates the condition. Thus, due to the influence of the hallucinated string, none of the alignment candidates could satisfy condition. A complete list of the obtained alignment candidates is provided in appendix C.

As an example of a corrected omission, the baseline model predicted the pronunciation かなめしょうがっこう for the facility name 豊島区立要小学校 correctly read as としまくりつかなめしょうがっこう *Toshimakuritsu Kanameshogakko*, omitting the first part of the pronunciation. Figure 3b shows one of the alignment candidates for this incorrect output, demonstrating that the validity condition $C(\mathbf{a})$ is violated because there is no output corresponding to the 豊島区立 part. A complete list of the obtained alignment candidates is also provided in appendix C.

5.2 Analysis of Alignment Candidate Extraction

While the definition of alignment candidates in this paper follows eq. (1), as discussed in sections 3.2 and 3.5, various alternative definitions have been proposed in prior work. We also conducted experiments to evaluate the accuracy when using these alternative extraction methods.

Let $\mathcal{A}^{\text{AWI}}$ and $\mathcal{A}^{\text{AWO}}$ denote the sets of alignment candidates obtained using the AWI and AWO approaches defined in section 3.2, respectively. We consider three candidate sets: $\mathcal{A}^{\text{AWI}}$, $\mathcal{A}^{\text{AWO}}$, and their union $\mathcal{A}^{\text{AWI}} \cup \mathcal{A}^{\text{AWO}}$. Following the discussion in section 3.5, we also evaluate four combinations of candidate extraction methods: using either the weight-based or norm-based approach, and applying either head-wise extraction or integration across heads within each layer.

Table 3 compares the performance. The most significant finding is that head-wise extraction of candidates using the AWI setting yields a substantial improvement in accuracy. This corresponds to the definition given in eq. (1). Using $\mathcal{A}^{\text{AWO}}$ alone resulted in degraded accuracy, and $\mathcal{A}^{\text{AWI}} \cup \mathcal{A}^{\text{AWO}}$ showed no improvement over using $\mathcal{A}^{\text{AWI}}$ alone. Finally, extraction from integrated-per-layer setting resulted in accuracy significantly lower than head-wise extraction.

While Kobayashi et al. (2020) reported the superiority of the norm-based method, our experiments found no significant difference between it and the weight-based method. We attribute this discrepancy to the absence of a source-side `EOS` token in our setting. The primary advantage of norm-based analysis is its ability to mitigate the over-concentration of attention weights on the source-side `EOS` token, which is sometimes introduced to handle target tokens without a corresponding source. However, in our pronunciation estimation task, every target token is expected to align with some source token by design. Therefore, there is no need to insert a source-side `EOS` token, effectively nullifying the main advantage of the norm-based method. In this context, the simpler weights-based method is sufficient.

ALTI (Ferrando et al., 2022b,a; Dale et al., 2023a; Ferrando et al., 2023) is a method that aggregates contributions from all attention modules in both the encoder and decoder to quantify how much each source token contributes to each target token. Recent studies have used the aggregated contribution per source or target token as an anomaly score to detect hallucinations and omissions, making ALTI a common baseline for such tasks (Dale et al., 2023a,b; Guerreiro et al., 2023a,b). Table 3 also reports the results of applying our method to hard alignments obtained from ALTI's aggregated weights. In this setting, the output quality degraded compared to the original model. We attribute this to a mismatch between the soft, continuous nature of ALTI's aggregated weights and our filtering criteria, which require hard, discrete alignments.

6 Related Work

6.1 Speech Processing

Methods focusing on the monotonicity between source and target have been used in speech processing. A typical approach is to restrict attention to be inherently monotonic by changing model architecture (Chiu and Raffel, 2018).

Wang et al. (2024) propose a decoding-time constraint method that prevents omission by focusing on specific attention heads containing alignment information. Unlike our method, their approach requires reference data with gold alignments to identify the relevant attention heads.

6.2 Detection of Hallucinations and Omissions

Various methods have been proposed for detecting hallucinations and omissions. Dale et al. (2023b) categorize these into *internal* methods, which focus on the internal behavior of the model, such as the attention weight, and *external* methods, which make judgments using external models related to sentence similarity or translation quality.

Approaches focusing on attention weights include methods that track the total weight assigned to the source-side `EOS` token (Berard et al., 2019), as well as entropy-based methods that detect over-concentration on specific tokens (Lee et al., 2018). In addition to these, log-likelihood-based methods have also been proposed for hallucination detection (Guerreiro et al., 2023c).

7 Conclusions and Future Work

We formulated alignment conditions that are free of hallucinations and omissions, based on the linguistic properties of Japanese pronunciation. Furthermore, we proposed an existence-based alignment check and a corresponding filtering method to correct hallucinations and omissions. Through experiments, we demonstrated that our method effectively reduces hallucinations and omissions, thereby improving the accuracy of pronunciation estimation. Although the absolute improvement of 2.5% may appear modest, it corresponds to the correction of hallucinations and omissions, which are infrequent but critical errors. Considering that this method has already been deployed in real-world products, the improvement has substantial practical impact.

Currently, our method focuses solely on the character type on the source side, without leveraging specific pronunciation patterns for individual characters. Incorporating such information to define stronger alignment conditions is a promising direction for future research. However, applying stricter conditions directly as a filter may increase false positives and ultimately degrade overall accuracy. One potential solution is to treat the detection results as features and use them in a reranking model, rather than applying them as a hard filter.

8 Limitations

Our method fundamentally relies on the existence of hard, character-level alignments, particularly assuming a one-to-many correspondence between source and target tokens. This assumption holds well for most kanji and kana characters. However, this assumption breaks down for alphanumeric characters. In such cases, it is not uncommon for characters to contribute nothing to the output, leading to a pronunciation length of zero. As a result, our method may fail to detect omission errors for these characters, since zero-length alignments are allowed by design.

Furthermore, Japanese contains a linguistic phenomenon known as *Jukujikun*, in which a single word composed of multiple kanji characters is assigned a unique pronunciation that cannot be decomposed into pronunciations for individual characters. In most cases, the length of the pronunciation equals or exceeds that of the kanji string, which enables our method to forcibly assign plausible—though incorrect—character-level alignments without triggering error detection. However, in rarer cases such as 百舌鳥 read as もず *Mozu* or 再従兄弟 read as はとこ *Hatoko*, the kana-based pronunciation is shorter than the kanji representation, breaking our underlying assumption. In such cases, our current framework fails to produce a valid alignment, and ad-hoc handling becomes necessary.

Finally, our experiments were limited to short text strings and facility names. However, this setup directly reflects the motivation of solving practical business challenges, which guided the design of this study. Of course, evaluating the method's effectiveness on longer texts remains an important question for future investigation.

References

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages

526–532, Florence, Italy. Association for Computational Linguistics.

Nancy Chen, Rafael E. Banchs, Min Zhang, Xiangyu Duan, and Haizhou Li. 2018. Report of NEWS 2018 named entity transliteration shared task. In *Proceedings of the Seventh Named Entities Workshop*, pages 55–73, Melbourne, Australia. Association for Computational Linguistics.

Chung-Cheng Chiu and Colin Raffel. 2018. Monotonic chunkwise attention. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

David Dale, Elena Voita, Loic Barrault, and Marta R. Costa-jussà. 2023a. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–50, Toronto, Canada. Association for Computational Linguistics.

David Dale, Elena Voita, Janice Lam, Prangthip Hansanti, Christophe Ropers, Elahe Kalbassi, Cynthia Gao, Loic Barrault, and Marta Costa-jussà. 2023b. HalOmi: A manually annotated benchmark for multilingual hallucination and omission detection in machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 638–653, Singapore. Association for Computational Linguistics.

Yasuharu Den, Toshinobu Ogiso, Hideki Ogura, Atsushi Yamada, Nobuaki Menematsu, Kiyotaka Uchimoto, and Hanae Koiso. 2007. The development of an electronic dictionary for morphological analysis and its application to japanese corpus linguistics. *Japanese Linguistics*, 22:101–123.

Javier Ferrando and Marta R. Costa-jussà. 2021. Attention weights in transformer NMT fail aligning words between sequences but largely explain model predictions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 434–443, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022a. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8756–8769, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022b. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Nuno M. Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023a. Hallucinations in large multilingual translation models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.

Nuno M. Guerreiro, Pierre Colombo, Pablo Piantanida, and André Martins. 2023b. Optimal transport for unsupervised hallucination detection in neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13766–13784, Toronto, Canada. Association for Computational Linguistics.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023c. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Jun Hatori and Hisami Suzuki. 2011. Japanese pronunciation prediction as phrasal statistical machine translation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 120–128, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Llion Jones, Richard Sproat, Haruko Ishikawa, and Alexander Gutkin. 2023. Helpful neighbors: Leveraging neighbors in geographic feature pronunciation. *Transactions of the Association for Computational Linguistics*, 11:85–101.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. OpenReview, submitted to ICLR2019.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Yuval Merhav and Stephen Ash. 2018. Design challenges in named entity transliteration. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 630–640, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hankun Wang, Chenpeng Du, Yiwei Guo, Shuai Wang, Xie Chen, and Kai Yu. 2024. Attention-constrained inference for robust decoder-only text-to-speech. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 630–637.

A Model Parameters

Table 4 presents the detailed hyperparameters for training.

B Performance of Baseline Model

Table 5 shows the baseline model performance. We report both top-1 accuracy (Acc) and top-k accuracy. The difference between the Acc and the top-k accuracy indicates the theoretical upper bound for the accuracy improvement from our filtering method.

C Examples of Alignment Candidates

Figures 4 to 7 denote the examples of extracted alignment candidates.

embed dim	512
ffn embed dim	2048
layer norm ϵ	1e-6
norm first	True
activation function	ReLU
loss type	cross entropy
label smoothing	0.1
optimizer	Adam
Adam β_1	0.9
Adam β_2	0.98
Adam ϵ	1e-9
lr scheduler	linear warmup & inverse square decay
warmup steps	5000
batch size	1024
max epoch	20
drop out	0.1
number of GPUs used	4

Table 4: Hyperparameters of the model.

		L			
H		1	2	3	6
4	Acc	80.97	82.07	82.10	81.73
	Top-4 Acc	92.13	92.70	92.20	92.13
	Top-8 Acc	93.90	94.63	94.47	94.37
	Top-12 Acc	94.83	95.53	95.17	95.17
	Top-16 Acc	95.50	95.83	95.60	95.73
8	Acc	81.60	82.17	81.67	81.63
	Top-4 Acc	92.27	92.43	93.00	92.17
	Top-8 Acc	94.37	94.63	94.70	94.27
	Top-12 Acc	94.93	95.40	95.50	95.50
	Top-16 Acc	95.77	95.97	96.07	96.00
16	Acc	81.40	81.67	82.17	81.77
	Top-4 Acc	92.33	92.70	93.07	92.50
	Top-8 Acc	94.03	94.33	94.83	94.60
	Top-12 Acc	94.90	95.33	95.57	95.30
	Top-16 Acc	95.30	95.70	96.13	95.80

Table 5: Top-k accuracy for various L and H configurations. Beam width is equal to k.

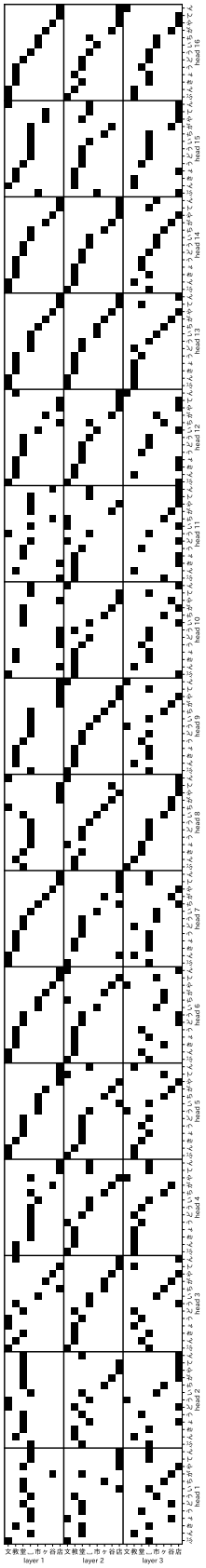


Figure 4: Alignment candidates between the Japanese phrase 文教堂 市ヶ谷店 and the correct pronunciation ぶんきょうどういちがやてん

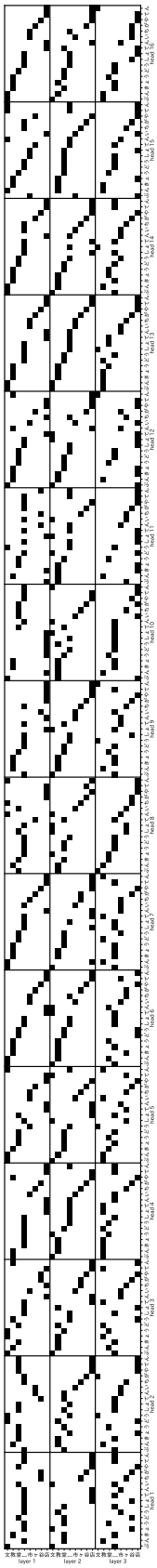


Figure 5: Alignment candidates between the Japanese phrase 文教堂 市ヶ谷店 and the incorrect pronunciation with hallucination ぶんきょうどうしょてんいちがやてん

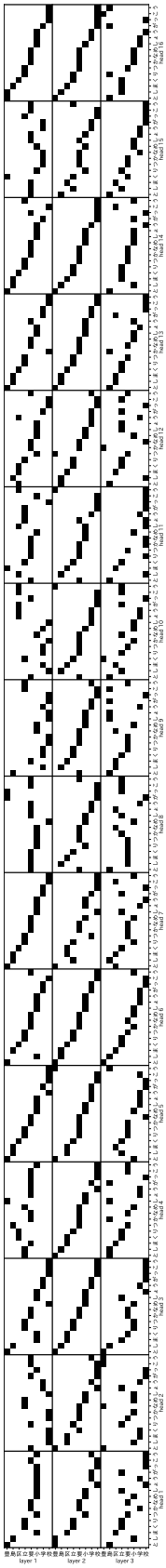


Figure 6: Alignment candidates between the Japanese phrase 豊島区立要小学校 and the correct pronunciation としまくりつかなめしょうがっこう

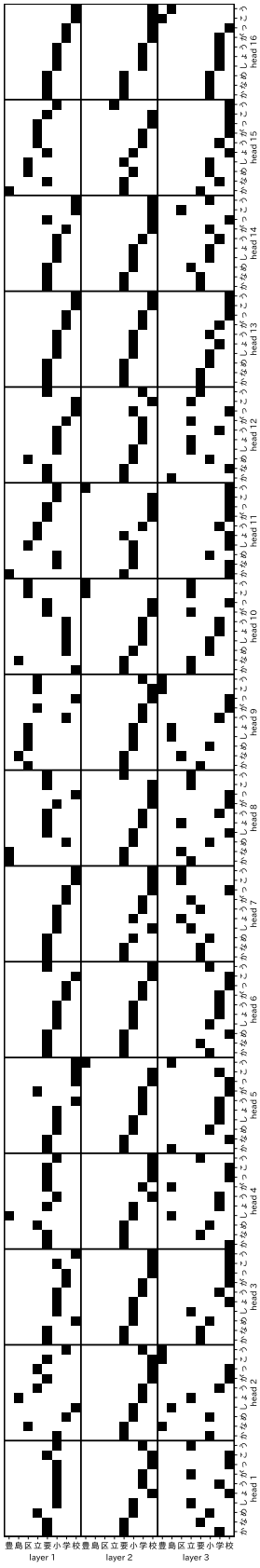


Figure 7: Alignment candidates between the Japanese phrase 豊島区立要小学校 and the incorrect pronunciation with omission かなめしょうがっこう