Pronunciation Ambiguities in Japanese *Kanji*

Wen Zhang
The Graduate Center, City University of New York
wenzhang0222@gmail.com

Abstract

Japanese writing is a complex system, and a large part of the complexity resides in the use of *kanji*. A single *kanji* character in modern Japanese may have multiple pronunciations, either as native vocabulary or as words borrowed from Chinese. This causes a problem for text-to-speech synthesis (TTS) because the system has to predict which pronunciation of each *kanji* character is appropriate in the context. The problem is called homograph disambiguation. To solve the problem, this research provides a new annotated Japanese single *kanji* character pronunciation data set and describes an experiment using the logistic regression (LR) classifier. A baseline is computed to compare with the LR classifier accuracy. This experiment provides the first experimental research in Japanese single *kanji* homograph disambiguation. The annotated Japanese data is freely released to the public to support further work.

1 Introduction

Japanese uses a mixed writing system with three distinct scripts and one romanization. *Kanji* 漢字 is the writing script that borrows directly from Chinese characters which were introduced in Japan from China through Korea from the third century CE. There are 2,136 commonly used *kanji* characters termed *Joyo kanji* in present-day Japanese.[1] A single *kanji* character in modern Japanese may have multiple pronunciations derived from the linguistic history of the *kanji* characters as either native vocabulary words or as terms borrowed from Chinese. For instance, the *kanji* character 山 'mountain' can be read as either the native Japanese word *yama* or the Chinese-derived term *san*. The native Japanese pronunciations of the *kanji* character 文 'letter, sentence, writings' are *humi*, *aya*, and *kaza*, while Chinese borrowed pronunciations are *bun* and *mon*. Because a *kanji* character has multiple pronunciations, to predict the appropriate pronunciation for each *kanji* character, a text-to-speech synthesis engine must select the appropriate reading. This is a form of homograph disambiguation.

This research is a computational study of Japanese *kanji* homograph disambiguation. Recent research in homograph disambiguation in Japanese is limited because of the lack of extensive data sets that include comprehensive pronunciations for the most commonly used *kanji* characters. The goal of this research is to fill this void, make new data sets to conduct the analysis of *kanji* characters with multiple pronunciations, and use the computational methodology to test the data set to lay a foundation for computational research on Japanese *kanji* homographs in the future.

1.1 Japanese writing scripts

The Japanese writing system uses three different scripts, Chinese characters (*kanji*), and two *kana* systems: *hiragana* and *katakana*, which are derivatives of Chinese characters. *Hiragana* resulted from the cursive style of writing Chinese characters, while *katakana* developed from the abbreviation of Chinese characters. Roughly speaking, *kanji* are used for content words such as nouns, stems of adjectives，and verbs, whereas *hiragana* is used for writing grammatical words

[1] https://kanji.jitenon.jp/cat/joyo.html

Proceedings of the Workshop on Computation and Written Language (CAWL 2023), pages 50–60

(case markers and other 'small' words). *Katakana* is almost exclusively used today to write foreign words and names such as Tennessee テネシー *teneshii* (Sproat, 2009: 47).

In addition to the three writing scripts that originate in Chinese characters, *romaji* is another phonetic writing script using the Roman alphabets. Thus, Japanese essentially has four ways to write the language. For instance, the word for 'mountain' can be written as 山 in *kanji*, as やま in *hiragana*, as ヤマ in *katakana*, and *yama* in *romaji*. For more details, see Zhang (2023: 4f.).

1.2 *Kanji*

The following sections introduce the *kanji* pronunciation ambiguities.

1.2.1 *On* readings and *kun* readings

Over time as Chinese characters were adapted to Japanese, the characters came to be associated with native Japanese words as well. For instance, the Chinese character for *shān* 山 was borrowed and used to write the newly created Japanese morpheme /*san*/. However, the Japanese already had the word *yama* 'mountain'. The character 山 was also used to write the native word *yama*. Present-day Japanese has kept both terms for 'mountain' but uses them in different contexts. For instance, by itself, the signifier 'mountain' is usually referred to as *yama*, but Mt. Fuji is *Fujisan*. The *kanji* character 人 'people, person' has two borrowed pronunciations: *nin* and *jin*, and one native Japanese pronunciation *hito*; the *kanji* character 者 'person' has one borrowed pronunciation *sha*, and one native pronunciation *mono*. A majority of *kanji* characters have one or more Chinese-derived readings, and one or more native readings (Sproat et al., 2021). [2] The borrowed readings and native ones are known as *on* readings and *kun* readings, respectively. The readings in speech are not a problem, however, when given the written form first, for instance, both *san* and *yama* are written as 山, one must decide, depending on the context of each occasion, whether the character should be pronounced as *san* or *yama*. The multiple context-based pronunciations of a single *kanji* make Japanese text a challenge.

1.2.2 Multiple *on* readings

In general, a given *kanji* character may have several different Sino-Japanese readings reflecting the different stages at which the *kanji* character was borrowed from Chinese (Sproat, 2009: 47; Olinsky, 2000). Many Chinese words were assimilated into Japanese along with their characters and sounds during three unique historical periods. Each of these three periods of linguistic exchange are marked by a specific system of pronunciation. The three systems are *Go'on* 'Go pronunciations', *Kan'on* 'Kan pronunciations', and *To'on* 'To pronunciations'. For instance, the *kanji* character 行 has several *on* readings: *gyo*, *ko*, and *an*. For more details, see Zhang (2023: 7f.).

1.2.3 Multiple *kun* readings

The Chinese character borrowing has experienced at least three booms, and the cycles of *kanji* borrowing led to multiple usages for each single *kanji* character. In other words, one *kanji* character can hold multiple Japanese native readings with disparate associated meanings. For instance, the *kanji* character 生 has several *kun* readings each with a different meaning: *iki* 'live, exist', *hae* 'grass grows', *nama* 'raw', and *u* 'to produce, give birth to'; 生 can also be read as *ha*, *o*, *ki*, *inochi*, *ubu*, and *na*.

1.2.4 Personal name readings

Personal name readings are a reading category different from *on* readings and *kun* readings. A *kanji* character has diverse readings in personal names which are different from its *on* readings and *kun* readings. For instance, the *kanji* character 一 has several personal name readings: *i*, *osamu*, *ka*, *kazu*, and *katsu*. For more details, see Zhang (2023: 9f.).

1.2.5 Reading ambiguities

Each single *kanji* character, generally, has at least one *on* and at least one *kun* reading. Because a *kanji* character has multiple readings, and each reading is used in different senses, when encountering a *kanji* character, one needs to figure out an appropriate contextual reading for the *kanji* character. For instance, the *kanji* character 行 is

[2] There is a *kanji* category called 和製漢語 *wasei kango* 'Japanese-made Chinese-character-based words'. *Wasei kango* are words that are composed of Chinese morphemes but were made by the Japanese rather than borrowed from Chinese. The items have *kanji* forms and most of them are read based on the *kun* reading rules, e.g., 蛯 *ebi* 'shrimp', 躾 *shitsuke* 'upbringing', and 凧 *tako* 'kite'. A few of them have both *on* reading and *kun* reading, e.g., 雫 *da* (*on* reading), *shizuku* (*kun* reading) 'driblet', and 鱈 *setsu* (*on* reading), *tara* (*kun* reading) 'cod'.

pronounced *gyo* in 修行 *shugyo* ‘ascetic practices, training’, *ko* in 行動 *kodo* ‘action’, and *an* in 行脚 *angya* ‘pilgrimage’.

1.3 TTS and TTS approaches

Text-to-speech synthesis (TTS) is a technology that allows written text to be output as speech. Because people are in fact very sensitive to both the words and the way they are spoken, the goals in building a high-quality TTS system should clearly get across the message and use a human-like voice. These two goals of TTS are called intelligibility and naturalness (Taylor, 2009: 2-3).

The TTS problem is traditionally split into front-end and back-end systems. As one of the front-end system problems, the TTS system must predict the pronunciations of the words. For the in-vocabulary words with a single pronunciation, this requires only dictionary lookup. But for other types of words, for instance, homographs, because polysemous words are pronounced differently depending on the intended sense, one must analyze the context in which a *kanji* character occurs to select a contextually appropriate pronunciation (Gorman et al., 2018). This problem has been studied as homograph disambiguation, e.g., in English and a few other languages. A number of methods have been tried for several disambiguation tasks in NLP, including part of speech (POS) tagging and decision lists. Sproat et al. (1992) propose statistics of POS bigram or trigram to solve the problem and improve the disambiguation performance with words that have different POS taggers. Yarowsky (1994, 1997) presents decision list algorithms that combine the strengths of n-gram taggers, Bayesian classifiers, and decision trees in a highly effective general-purpose decision procedure for lexical ambiguity resolution. Gorman et al. (2018) select a set of 163 homographs for the US English experiment and find that hybrid systems (making use of both rules and machine learning) are significantly more accurate than either hand-written rules or machine learning alone.

1.4 Japanese TTS homograph ambiguities

Japanese writing is a complex system, and a large part of the complexity resides in the reading of *kanji* characters. The trick in any case is to know which is the right reading, which makes reading Japanese text a challenge for the TTS system (Sproat, 2009: 47). As discussed in section 1.2.1, the *kanji* character 山 ‘mountain’ could be pronounced either *san* or *yama*. The two pronunciations share the same meaning. However, the TTS system must do homograph disambiguation to find an appropriate pronunciation based on the contextual information of the *kanji* character.

Two features are also related to the Japanese homograph disambiguation performance: word boundaries and formality. Because there is no word-boundary delimiter in Japanese, it is hard to identify a word. In the word segmentation process, if word boundaries cannot be identified correctly, it may lead TTS to incorrectly pronounce a string (Olinsky, 2000; Ooyama et al., 1987; Tesprasit et al., 2003). Therefore, problems of word boundary ambiguity and homograph ambiguity always occur together. Additionally, because Japanese writing is a combination of different scripts, *kanji* is used primarily for stems and *hiragana* is used for most inflectional endings and grammatical devices, word boundary discrimination can be simplified by detecting the transitions of the two different scripts. For instance, the string “現代の行政区分 ” ‘modern administrative divisions’ can be segmented at least into three tokens with the intervention of the *hiragana* の ‘possessive particle’ between the two *kanji* clusters: “現代 ” ‘modern time’ and “行政区分” ‘administrative divisions’. *Katakana*, on the other hand, is used primarily for phonetic renderings of foreign words, further reducing ambiguity. Thus, *kanji* can be considered the “harder” case for word segmentation (Olinsky, 2000). For instance, the *kanji* string “米国産業界 ” can be segmented into two separate ways. The first *kanji* character holds different meanings and pronunciations based on the different segmentations. The first segmentation is 米国 ‘America’ 産業 ‘Industry’ 界 ‘Realm’, in which the first *kanji* character 米 is pronounced *Bei*; the second segmentation is 米 ‘Rice’ 国産 ‘Domestic production’ 業界 ‘Industry’, in which the *kanji* character 米 is pronounced *Kome*. Taylor (2009: 46) discusses the homograph syntactic ambiguity using English sentence “Police help dog bite victim” which has at least two different possible syntactic patterns: (Police help dog) bite victim; and Police help (dog bite victim). The homograph syntactic ambiguities also exist in Japanese sentences. If not processed appropriately, it can hurt the performance of one of the TTS goals—intelligibility.

Japanese is famous for its politeness and formality. Some Japanese words have both informal and formal forms. Formal Japanese forms can additionally be divided into three categories: 丁寧語 *teinei-go* 'polite form', 尊敬語 *sonkei-go* 'honorific form', and 謙譲語 *kenjo-go* 'humble form'. The *kanji* word 今日 'today' has two pronunciations, the informal pronunciation is *kyo*, and the formal pronunciation is *konnichi*. Therefore, given the *kanji* word 今日, the system needs to analyze the formality based on the contextual information and select an appropriate pronunciation accordingly. Whether the system can do the contextual formality analysis perfectly or not will affect the achievement of the other TTS goal—naturalness.

In addition to its importance in TTS applications, homograph disambiguation is relevant to automatic speech recognition (ASR) and is also a subset of word sense disambiguation (WSD) (Seale, 2021). Therefore, the task of *kanji* homograph disambiguation is not only important in improving the Japanese TTS performance but is also a crucial part of gaining high ASR and WSD accuracy.

However, Japanese *kanji* homograph disambiguation, to the best of the author's knowledge, is not currently attested to in peer-reviewed literature or otherwise published online. Also, there does not exist a well-developed data set that can support the research.

1.5 Labeling in Japanese *kanji* homographs

In TTS synthesis, the selection of the correct pronunciation of a text string occurs when a homograph is encountered (Seale, 2021). Homographs are pronounced differently depending on the intended sense, and the context provides enough clues for a homograph to select a contextually appropriate pronunciation (Gorman et al., 2018; Hearst, 1991). Yarowsky (1997) describes the techniques of English homograph disambiguation where each homograph is labeled originally by hand and a collection of features such as nearby content words. We manually label the pronunciations for each *kanji* homograph given a context and use machine learning methods to test the performance of the Japanese *kanji* homograph disambiguation. The data is released to the public for further experimentation by the NLP research community.

1.6 Research contributions

This research serves as the first academic work focused on Japanese *kanji* homograph disambiguation. Its contributions include publicizing the first single *kanji* pronunciation annotated data set, a typology of homographs with implications for both labeling and modeling and offering substantial language-specific resources to do Japanese homograph disambiguation. The data is released to the public for further experimentation by the NLP research community.

2 Data-driven *kanji* homograph research

This chapter introduces the Japanese *kanji* homograph data collecting, labeling, and modeling. Although very well respected at the current time, this research determined that the part of speech (POS) method is not compatible with Japanese homograph disambiguation. An explanation will be presented at the beginning of the chapter.

2.1 The reason for not using POS

With the increasing availability of annotated language data, several statistical part of speech (POS) tags have been developed which achieve high accuracy. Many prior work (Asahara et al., 2000; Brants, 2000; Denis, 2009; Gorman et al., 2018; Manning, 2011; Ratnaparkhi, 1997; Seale, 2021; Toutanova et al., 2000) uses POS features for disambiguating words. However, unlike homographs in some languages, the readings of *kanji* are generally not disambiguated by POS tags: firstly, most readings of *kanji* characters correspond to the same parts of speech, for instance, the *kanji* character 山 is a noun whether it is read as *san* or *yama*; secondly, some *kanji* characters cannot be assigned part of speech, for instance, the *kanji* character 文 is a component of the nouns 文化 'culture', 文法 'grammar', and 文学 'literature', and the part of speech for 文 itself cannot be defined. Therefore, POS annotation was not adopted as an analyzer in this research. For more details, see Zhang (2023: 16f.).

2.2 *Kanji* homograph data

The goal of this research is to construct a way of determining an appropriate pronunciation for Japanese *kanji* homographs. For this purpose, a data set labeled the pronunciation of commonly used single *kanji* characters was constructed.

2.2.1 Data collection

The labeled single *kanji* homograph data set is constructed using the following data: Japanese dictionary Jiten [3] and Universal Dependencies (UD) Japanese-GSD.[4] Jiten is an online Japanese dictionary. According to Jiten, as of June 2023, the number of recorded *kanji* characters is 27,693, and the total number of commonly used *kanji* characters is 2,136. This research collected each commonly used *kanji* character readings, including *on* readings, *kun* readings, and personal name readings. In addition, sentences from UD Japanese-GSD were combed for the context of commonly used *kanji* characters. These *kanji* characters and their pronunciations were combined with the Jiten *kanji* characters. The UD Japanese-GSD resource consists of sentences from Wikipedia and sentences which have been automatically split into words by IBM's word segmenter (Asahara et al., 2018). The data set is segmented into 193,654 tokens and 8,100 sentences, and divided into training, development, and test sets. This research ignored the original splits for data collection.

2.2.2 *Kanji* homograph extractions

This research used Python libraries, data classes, and collections to extract the *kanji* homographs. The top 100 most commonly used *kanji* homographs in the combined UD Japanese-GSD data set were extracted. After this process, some kanji homographs were excluded. Firstly, it was determined that a *kanji* homograph with a frequency of 50 or more occurrences was optimal, because if there is not enough data for a given *kanji* homograph, we cannot build a good classifier. This resulted in 86 *kanji* homographs. Secondly, 18 semiotic classes were excluded. There are some semiotic classes, for instance, computer languages, email addresses, dates, times, telephone numbers, and postal addresses are much simpler than natural language, and problems will arise when we mix the natural language and those semiotic systems in the same signal and using the same characters to do so (Taylor, 2009: 33-34). In Japanese, the reading of a *kanji* character inside a number or date expression is different from reading a *kanji* character that is not a part of one of those expressions. For instance, the *kanji* homograph 一 'one' has multiple readings. When 一 stands alone, the reading is *ichi*; when 一 is one part of the semiotic classes, for example, *ichimai* 'one piece of', *ikko* 'one', and *hitotsu* 'one', the reading will be *ichi*, *itsu*, and *hito*, respectively, depending on the following characters. The third exclusion ensures that each pronunciation must occur more than once and be at least 2% overall of the annotated data. Therefore, *kanji* homographs with only one pronunciation were removed so as not to bolster model scores, as the models would correctly pick the only pronunciation class available, and 36 *kanji* homographs removed due to pronunciation invariance. Thus, there were 32 *kanji* homographs retained. 1 *kanji* homograph was excluded due to Japanese formality. The *kanji* character 私 is a first-person singular pronoun that has two pronunciations: *watashi* and *watakushi*. There is only a slight difference between the two pronunciations, and distinguishing the two pronunciations requires subtle context. Therefore, it was excluded to avoid confusing the system. 2 *kanji* homographs were removed due to automatic *kanji* homograph extraction errors.[5] Data for a further 3 *kanji* homographs were removed because the number of examples that can be labeled was very small.[6] For more details, see Zhang (2023: 18f.). After removing 74 *kanji* homographs, the remaining UD Japanese-GSD homograph data contains 26 unique homographs, and the 26 *kanji* homographs were modeled.

2.2.3 *Kanji* homograph pronunciation class size

Each *kanji* homograph in the data set has at least two pronunciations. 77% of the *kanji* homographs have two pronunciations, and 23% have three or four pronunciations. 70% of the *kanji* homographs with two pronunciations have one commonly used pronunciation, and the commonly used pronunciation is greater than or equal to 40% of the available data. One *kanji* homograph has the largest difference in pronunciation class size, with a ratio of 4:96, and one has a ratio of 49:51. On the other hand, four *kanji* homographs with more than two pronunciations have two commonly used pronunciations. Those two pronunciations accounted for around 90% of the available data and the ratio of the two is very close.

[3] https://jitenon.com/cat/common_kanji.php

[4] The treebank is licensed under the Creative Commons License Attribution-ShareAlike 4.0 International.

[5] The *kanji* homographs are 見 and 出.

[6] The *kanji* homographs are 名, 位, and 次.

いつ	も	、	人	が	沢山	い	ます。
$t-2$	$t-1$		t	$t+1$	$t+2$		

Table 1: Example of n-gram features for the ambiguous *kanji* character 人 'people, person'.

2.2.4 Data split redistribution

The total number of examples for the 26 *kanji* homographs in the data set is 1,903. These samples were split into 80% train, 10% dev, and 10% test. Stratified sampling was used as the default to maintain pronunciation class distribution among the splits. The stratified sampling method could be advantageous to sample each *kanji* homograph pronunciation category independently.

2.2.5 Data release

The new data sets were released for general use with the hope of helping to advance future research in Japanese and cross-lingual homographs. The data sets were released in two parts. First, 2,136 commonly used *kanji* homographs with their readings and reading types were released in a tab-separated values (TSV) file.[7] Readings for each *kanji* homograph include *on* readings, *kun* readings, and personal name readings. An annotated UD Japanese-GSD—a data set of *kanji* homograph readings in context was also released.[8] It includes sentences in which the target *kanji* homograph has been found.

2.3 Modeling

As the task of homograph disambiguation is to select a contextually appropriate pronunciation for a homograph, a logistic regression classifier was developed to make pronunciation predictions, and a baseline was computed to compare it against. While a baseline makes predictions that ignore the input features, the logistic regression classifier derives the input features from the homographs and the context surrounding them.

2.3.1 Baseline

A baseline was computed using the most frequent class label for each homograph, and then it was compared to the logistic regression classifier accuracy.

2.3.2 Logistic regression classifier

As the main task of homograph disambiguation is to select an appropriate pronunciation for a homograph given the context, a logistic regression (henceforth, LR) classifier was developed which considers the contextual features. In the development of the LR classifier, one LR classifier per-*kanji* was trained with the following n-gram features: tokens indexed one and two before and behind the homograph token, bigrams indexed immediately before and after the homograph token, and a skip-gram, the constituents of which surround the homograph token.[9] Table 1 displays one example of the *kanji* homograph n-gram features: t shows the position of the target *kanji* homograph; t − 2 "も" and t − 1 "、" are the left two tokens, while t + 1 "が" and t + 2 "沢山" are the right two tokens. Each unigram t − 2, t − 1, t + 1, and t + 2; the previous bigram t − 2 and t − 1; the following bigram t + 1 and t + 2; and the skip-gram bigram t − 1 and t + 1 were extracted as the target token features.

2.4 Evaluation and analysis

Per-class accuracy of one of the models from the most performant model type is reviewed, and error analysis is done for all models.

[7] https://github.com/wenzhang0222/thesis
[8] https://github.com/wenzhang0222/thesis

[9] Writing script categories (*kanji*, *hiragana*, and *katakana*) for n-grams were also checked but not selected as one of the features because they did not help the overall performance.

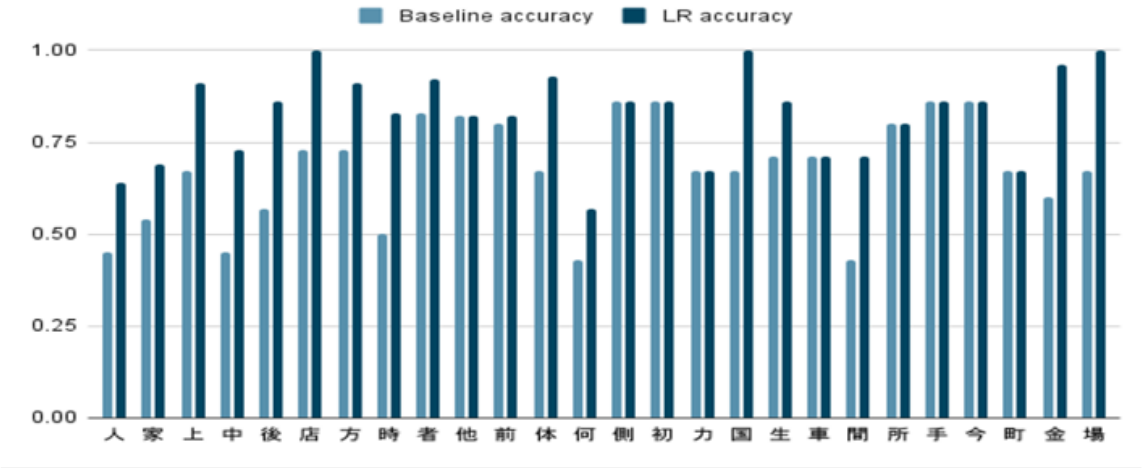

Figure 1: Baseline and LR accuracy for each *kanji* homograph.

Model	Micro acc.	Macro acc.
baseline	.67	.67
LR	.83	.83

Table 2: 26 *kanji* models' micro and macro accuracy.

2.4.1 Evaluation procedures

A baseline and LR classifier were trained and evaluated on the test set. Randomness can play a major role in the outcome of experiments and common sequence tagging tasks, the seed value for the random number generator can result in statistically significant differences for state-of-the-art systems (Reimers et al., 2017). Because randomness is inherent in the model, different results will be obtained if it is run multiple times. To protect against the human selection of metrics from a particularly good run, the reported metrics were taken from the model with the median balanced accuracy from each set of five models' performances. Hyperparameters were adjusted during training: L_1 regularization, LIBLINEAR solver, and hyperparameter C = 10. Separate models were trained for each *kanji* homograph. In addition, development of the models was done using the train and dev splits, and metrics were reported on the test split.

2.4.2 Metrics

The accuracy of a model provides a measure of its predictions' proximity to the correct values. Accuracy is determined in the range between 0 and 1. The performance of the models was evaluated using micro accuracy and macro accuracy.

2.4.3 Model performance

The micro and macro accuracies for the baseline and the LR classifier trained in this research are recorded in Table 2 — they show, on average, an increase between baseline and the non-baseline accuracies. The reasons are, firstly, logistic regression is famous for handling classification problems; secondly, L_1 regularization was applied in the LR classifier, and it can handle both dense and sparse input.

2.4.4 Per-*kanji* homograph performance

Figure 1 shows the accuracies for each *kanji* homograph: the baseline accuracy is represented by light blue histograms and the LR accuracy is shown in dark blue histograms. While the baseline accuracies range from .43 to .86, the LR accuracies range from 0.57 to 1.00. Overall, the LR accuracies outperform baseline accuracies. For more details, see Zhang (2023: 29f.).

2.4.5 Error analysis

The following sections report error analysis based on the errors made by the baseline and the LR classifier.

Kanji geographical feature error analysis

Reading a *kanji* character in some place names is a problem due to many *kanji* characters in those place names do not follow the default or general reading rules (Jones et al., 2022). Most parts of

Japan have their own dialects that can be used for colloquial interactions. As a result, *kanji* reading in place names are sometime following local traditions or dialects. In this research, there is one *kanji* character that reflects the Japanese *kanji* geographical features. The *kanji* character is 町 'town'. 町 has two pronunciations: the *on* reading *cho* and the *kun* reading *machi*. The reading of 町 seems casual and it is largely influenced by its geographical location. Both 町 's baseline and LR accuracy are .67. For more details, see Zhang (2023: 31f.).

Fixed expression error analysis

Most *kanji* characters have a commonly used *on* reading and a commonly used *kun* reading. Generally, when *on* reading and *kun* reading share the same meaning, the context will select a reading based on the n-gram features and sentential formality. While *kun* reading is casual, *on* reading is more formal. However, in some fixed expressions, due to history and/or geographical reasons, *on* reading and *kun* reading will no longer be distinguished, and there is only one reading. For instance, the *kanji* character 生 has two commonly used pronunciations: one is the *on* reading *sei*, the other is the *kun* reading *nama*. When 生 means 'live', it can only be read as *nama* instead of *sei*. The LR classifier predicts 50% incorrectly in this case. Other examples can be found in Zhang (2023: 34f.).

Formality error analysis

The Japanese honorific system is well-developed, ranging from pronouncing a single *kanji* character in a specific context to choosing sentence patterns and expressions. We retained a mild formality *kanji* homograph to test whether the model can learn, and to what extent formality can be learned during training. The *kanji* character is 他 'others'. 他 can combine with the *hiragana* phrase その 'that' to make a fixed expression その他 'the others', and it has two pronunciations in the combination: *hoka* and *ta*. The only difference between the pronunciations *sonohoka* and *sonota* is that *sonohoka* sounds more casual, which can be used in daily life conversations; *sonata* is formal, can be found in business expressions and official documents. The LR accuracy of the *kanji* character 他 is .82, and all the incorrect predictions are about the formality pronunciations of 他 in the combination その他. This indicates that the LR classifier was not able to differentiate the Japanese formality robustly.

Pronunciation class size imbalanced error analysis

There are two *kanji* characters that each of which has four pronunciations, one is the *kanji* character 家 'home, house, family' and one is the *kanji* character 後 'later, back'. The character 家 's pronunciations are: *ka*, *ke*, *ie*, and *ya*. Among the 105 examples of the *kanji* character 家, the pronunciation *ka* counts for 62%, *ke* is 19%, *ie* is 17%, and *ya* only counts for 2%. The LR classifier could not distinguish the two commonly used pronunciations: *ka* and *ke* and predicted all the *ke* to *ka*. LR also predicted *ya* to *ka*. Because the ratio of the pronunciation *ya* is very low, it can be assumed that features of the pronunciation *ya* were not fully learned by the LR classifier during training, and the LR classifier used the most frequent pronunciation *ka* to predict it. Other examples can be found in Zhang (2023: 35f.)

3 Discussion and conclusion

This research has been motivated by providing the first annotated Japanese single *kanji* pronunciation data set to solve Japanese *kanji* reading ambiguities. Although the pronunciation of Japanese *kanji* characters is a bottleneck on TTS performance, it has not been studied seriously due to the lack of reliable publicly available data. At the beginning of Chapter 2, this research addressed weaknesses in the use of part of speech (POS) as a mean of Japanese *kanji* homograph disambiguation and expounded on the source and methods of obtaining and processing the data. Some Japanese characteristics, for instance, *rendaku*, formality, and geographical features were taken in account when processing data.

Baseline and logistic regression (LR) classifier were used to examine the data performance. While the baseline can only obtain 67% prediction accuracy, the LR classifier, with the help of the n-gram feature extractions, effective statistical analysis and regularization, improved the prediction accuracy to 83%. The following sections provide information about the known limitations of this research and directions for future research.

3.1 Known limitations and future research

As mentioned in Chapter 2, since data is annotated by the author in person, it may include some human error in labeling. However, this can be resolved through a review of the labels and the publication of an amended version. In addition, the sentence data is obtained from the Universal Dependencies (UD) Japanese-GSD data set, this is just one of eight UD Japanese corpora and other ones could be used to expand the data set. Also, this research treats the single *kanji* homographs as the target, and the work could undoubtedly be improved by expanding the research to *kanji* combinations. As discussed in Chapter 1, Japanese is one of the languages that lack word boundaries. Therefore, the first interesting point will be that when a *kanji* homograph is in a *kanji* combination or phrase, which *kanji* homographs will tie together to make a word to create a word boundary with other *kanji* homographs. Then the second point is how the new *kanji* combination can affect the pronunciation selection of those *kanji* homographs.

An anonymous reviewer suggests that we compare against the *kanji* disambiguation system embedded in `MeCab`. However, we leave this comparison for future work.

Finally, due to time constraints, this research extracts n-gram features for the target *kanji* homographs. There will be other features that help analyze the context to improve the model performance.

3.2 Conclusion

This research has pioneered labeling for the task of Japanese *kanji* homograph disambiguation in text-to-speech applications. It contributes to providing the first public free *kanji* homograph annotated data. New data sets are offered to the research community to provide further research on this work. This research also provides a typology of homographs based on specific language features. The direction is set for future research in Japanese homograph studies and can be extended to research on the writing system of Chinese characters.

Acknowledgements

I would like to express my gratitude to my advisor, Dr. Kyle Gorman, for his generous and professional guidance on my studies at the Graduate Center, City University of New York, for helping me to shape and fine-tune this research, and for sharing his expertise and time with me throughout the paper. I deeply appreciate the opportunity to have worked together. I would also like to thank anonymous reviewers for their helpful feedback.

References

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Genirative Model for Raw Audio. Proc. 9th ISCA Workshop on Speech Synthesis workshop (SSW9), 125.

Adwait Ratnaparkhi. 1997. A maximum entropy model for part-of-speech tagging. In *EMNLP*, pages 133–142. Reimers, N., and Gurevych, I. (2017). Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. In arXiv:1707.06799.

Anders Søgaard. 2010. Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 205–208.

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *CICLing*, pages 171–189.

Craig Olinsky and Alan W. Black. 2000. Non-Standard Word and Homograph Resolution for Asian Language Text Analysis. Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH 2000, pages 733-736.

Daniela Braga., Luís Coelho, and Fernando Gil V. Resende Jr. 2007. Homograph ambiguity resolution in front-end design for Portuguese TTS systems. In *INTERSPEECH*, pages 1761–1764.

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 88-95.

David Yarowsky. 1997. Homograph disambiguation in text-to-speech synthesis. In *Jan P. H. van Santen, et al., editors, Progress in speech synthesis*, pages 157–172.

Denilson C. Silva, Daniela Braga, and Fernando Gil V. Resende Jr. 2012. A rule-based method for homograph disambiguation in Brazilian Portuguese text-to-speech. *Journal of Communications and Information Systems*, 27(1), pages 1–9.

Drahomíra "johanka" Spoustová, Jan Hajič, Jan Raab, and Miroslav Spousta. 2009. Semi-Supervised Training for the Averaged Perceptron POS Tagger. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 763–771.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pages 2825–2830.

Hang Li and Jun-ichi Takeuchi. 1997. Using Evidence that is both Strong and Reliable in Japanese Homograph Disambiguation. In *SIGNL*119-9, pages 53-59. IPSJ.

Honghui Dong, Jianhua Tao, and Bo Xu. 2004. Grapheme-to-phoneme conversion in Chinese TTS system. In *Proc.2004 International Symposium on Chinese Spoken Language Processing*, pages 165-168.

Jennifer M. Seale. 2021. Label imputation for homograph disambiguation: theoretical and practical approaches. Doctoral dissertation, The Graduate Center, City University of New York.

Junhui Zhang, Junjie Pan, Xiang Yin, Chen Li, Shichao Liu, Yang Zhang, Yuxuan Wang, and Zejun Ma. 2020. A hybrid text normalization system using multi-head self attention for mandarin. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6694–6698.

Junhui Zhang, Wudi Bao, Junjie Pan, Xiang Yin, and Zejun Ma. 2022. A Novel Chinese Dialect TTS Frontend with Non-Autoregressive Neural Machine Translation. In arXiv:2206.04922.

Junko Itô and Armin Mester. 1996. Rendaku I: Constraint Conjunction and the OCP. Kobe Phonology Forum 1996.

Kristina Toutanvoa and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *EMNLP*, pages 63–70.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL* 3, pages 252–259.

Kyle Gorman, Gleb Mazovetskiy, and Vitaly Nikolaev. 2018. Improving homograph disambiguation with supervised machine learning. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Kyle Gorman, Lucas F.E. Ashby, Aaron Goyzueta, Arya D. McCarthy, Shijie Wu, and Daniel You. 2020. The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 40–50. Association for Computational Linguistics.

Libin Shen, Giorgio Satta, and Aravind Joshi. 2007. Guided learning for bidirectional sequence classification. In *ACL 2007*.

Llion Jones, Richard Sproat, Haruko Ishikawa, and Alexander Gutkin. 2022. Helpful Neighbors: Leveraging Neighbors in Geographic Feature Pronunciation. In arXiv:2210.10200.

Lucas F.E. Ashby, Travis M. Bartley, Simon Clematide, Luca Del Signore, Cameron Gibson, Kyle Gorman, Yeonju Lee-Sikka, Peter Makarov, Aidan Malanoski, Sean Miller, Omar Ortiz, Reuben Raff, Arundhati Sengupta, Bora Seo, Yulia Spektor, and Winnie Yan. 2021. Results of the Second SIGMORPHON Shared Task on Multilingual Grapheme-to-Phoneme Conversion. *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 115–125.

Marti A. Hearst and Xerox Palo Alto Research Center. 1991. Noun homograph disambiguation using local context in large text corpora. In *Using Corpora*, pages 185-188.

Masayuki Asahara and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 21–27.

Masayuki Asahara, Hiroshi Kanayama, Takaaki Tanaka, Yusuke Miyao, Sumire Uematsu, Shinsuke Mori, Yuji Matsumoto, Mai Omura, and Yugo Murawaki. 2018. Universal Dependencies Version 2 for Japanese. In *LREC*.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348. Association for Computational Linguistics.

Pascal Denis and Benoît Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Pacific Asia Conference on Language, Information and Computation*, pages 110–119.

Pual Taylor. 2009. *Text-to-speech synthesis*. Cambridge University Press, Cambridge.

Richard Sproat. 2009. *Language, Technology, and Society*. Oxford University Press.

Richard Sproat and Navdeeep Jaitly. 2016. RNN Approaches to Text Normalization: A Challenge. In arXiv:1611.00068.

Richard Sproat, Julia Hirschberg, and David Yarowsky. 1992. *A corpus-based synthesizer*. *In ICSLP*, pages 563–566.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 92-97. Association for Computational Linguistics.

Stalin Aguirre and Josaf´a de Jesus Aguiar Pontes. 2019. A Japanese Word Segmentation Proposal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 429-435.

Stephen Mussmann, Robin Jia, Percy Liang. 2020. On the importance of adaptive data collection for extremely imbalanced pairwise tasks. In arXiv:2010.05103.

Takaaki Tanaka, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. 2016. Universal Dependencies for Japanese. In *LREC*.

Thorsten Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000)*, pages 224-231.

Virongrong Tesprasit, Paisarn Charoenpornsawat, and Virach Sornlertlamvanich. 2003. A context sensitive homograph disambiguation in Thai text-to-speech synthesis. In *Proc. HLT-NAACL'2003, short papers*, vol. 2.

Wen Zhang. 2023. Pronunciation ambiguities in Japanese *kanji*. Master's thesis, The Graduate Center, City University of New York.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. Using bilingual materials to develop word sense disambiguation methods. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, volume 112. Citeseer.

Yoshihumi Ooyama, Miyazaki Masahiro, and Ikehara Satoru. 1987. Natural Language Processing in a Japanese text-to-speech system. In *Proceedings of the 15th annual conference on Computer Science*, pages: 44-47.

Zolzaya Byambador, Ryota Nishimura, Altangerel Ayush, Kengo Ohta, and Norihide Kitaoka. 2021. Text-to-speech system for low-resource language using cross-lingual transfer learning and data augmentation. In *EURASIP Journal on Audio, Speech, and Music Processing* 2021:42.

1