Improving Dictionary Accessibility by Maximizing Use of Available Knowledge

Slaven Bilac[†]**, Timothy Baldwin**[‡] **and Hozumi Tanaka**[†]

†*Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan*
 `{sbilac,tanaka}@cs.titech.ac.jp`
‡*CSLI, Stanford University, Stanford CA, USA*
 `tbaldwin@csli.stanford.edu`

ABSTRACT. *The dictionary lookup of unknown words is particularly difficult in Japanese due to the requirement of knowing the correct word reading. We propose a system which supplements partial knowledge of word readings by allowing learners of Japanese to look up words according to their expected, but not necessarily correct, reading. This is an improvement from previous systems which provide no handling of incorrect readings. In preprocessing, we calculate the possible readings each kanji character can take and different types of phonological alternations and reading errors that can occur, and associate a probability with each. Using these probabilities and corpus-based frequencies we calculate a plausibility measure for each generated reading given a dictionary entry, based on the naive Bayes model. In response to a user-entered reading, the system displays a list of candidate dictionary entries for the user to choose from. The system is implemented in a web-based environment and available for general use. In the evaluation on Japanese Proficiency Test data and naturally occurring misreading data, the system significantly reduced the number of unsuccessful dictionary queries when queried with incorrect readings.*

RÉSUMÉ.

KEYWORDS: *language learner, dictionary lookup, Japanese, kanji*

MOTS-CLÉS : *apprentissage des langues, consultation de dictionnaires, japonais, kanji*

1. Introduction

Learning a foreign language is a time consuming and painstaking process, and made all the more daunting by the existence of unknown words [GRO 00]. Without a fast, low-cost way of looking unknown words up in a dictionary, the learning process is impeded [HUM 01]. The problem of dictionary lookup is particularly evident in non-alphabetic languages such as Japanese where the learner can easily be overwhelmed by the sheer number of characters and multitude of readings associated with each.

Educators have tried to lessen the unknown word problem by focusing on effective ways of expanding learner vocabulary [LAU 01]. However, unless the learner lives in a closed language world, s/he is always going to be exposed to unknown words, particularly in the earlier stages of learning. Our philosophy is to accept the inevitability of unknown words and focus instead on minimizing the dictionary lookup overhead.

Learners often possess only limited knowledge of the readings of characters and the phonological and conjugational processes governing word formation. This can make it difficult to identify the correct reading for a grapheme string, and the boolean match mechanism adopted by conventional dictionaries discourages the user from attempting to look up a word in the case that s/he is uncertain of the reading. We believe that if we can imitate the manner in which learners acquire and classify the different readings of characters and the rules governing overall reading formation, we should be able to decipher which dictionary entry the user was after even when queried with a (predictably) wrong reading. Thus, the purpose of this research is to develop a comprehensive and efficient dictionary interface allowing language learners to look up words in an error-resilient and intuitive manner. Furthermore, an important underlying motivation of this research is to remove the assumption of perfect reading knowledge made by conventional dictionary interfaces, and encourage the user to query the system with plausible but not necessarily correct readings.

The particular language we target in this paper is Japanese, and we choose to model readings by way of kana (see below). The problem of dictionary lookup for Japanese is particularly complex due to there being over 2000 ideographic kanji characters each with numerous phonemic realizations, frequent word conjugation and a lack of spaces between adjacent words. A learner trying to look up a word in a dictionary needs to cope with all these problems at once. The proposed system aims to help a user by allowing direct lookups based on the best guess the user is able to construct for a target word in written text based on available knowledge.

As an example of user–system interaction, consider that a user comes across the novel kanji compound 発表 (*happyou* "presentation"[1]) and wishes to determine its En-

---

1. In this paper, we loosely follow the Hepburn system of romanization, with the exception that we romanize long vowels as separate characters giving rise to *hyou* instead of *hyoo* or *hyō* for ひょう. The other notable divergence—taken from [BAC 94]—is the use of the upper-case *N* for syllable-final nasals (corresponding to the kana ん) and lower-case *n* for syllable-initial nasals (as found in な, for example).

glish translation. Lacking prescriptive knowledge of the pronunciation for the string, the user applies knowledge of alternate string contexts for the component kanji characters 発 *hatsu* and 表 *hyou* to postulate that the string is read as *hatsuhyou*. S/he inputs the kana for this string into the dictionary search interface, and gets back a list of Japanese words (in both kanji and correct kana-reading forms) with English translations for each. From among these, s/he is able to detect the original string in kanji form, ascertain the correct pronunciation for the string (*happyou*) and obtain the desired translation ("presentation").

Although we focus on Japanese in this paper, the basic method we propose is applicable to any language where the mapping from reading to orthography is not self-evident. That is, given some means of describing readings (whether through a phonetic representation or some other means) and the canonical orthographies of words, it is possible to apply the same procedure in predicting patterns of reading confusion. Japanese is of particular interest because of the wide range of factors which affect pronunciation prediction for an unknown word (see Section 2.4).

The remainder of this paper is organized as follows. Section 2 gives a short introduction to the Japanese writing system and dictionaries, and discusses reading errors common in learners of Japanese. Section 3 describes the basic system philosophy, and Section 4 the processing steps necessary for generating and scoring readings. The evaluation of the system is given in Section 5 and the discussion of the results and possible future research directions are given in Section 6. Finally, Section 7 gives concluding remarks.

2. The Japanese Language, Existing Japanese Dictionaries and Reading Errors

2.1. *The peculiarities of the Japanese writing system*

The Japanese writing system consists of the three orthographies of hiragana, katakana and kanji, which appear intermingled in modern-day texts. The **hiragana** and **katakana** syllabaries, collectively referred to as **kana**, are relatively small (46 basic characters each), and most characters take a unique and mutually exclusive reading which can easily be memorized.[2] Generally speaking, the function of these two scripts is distinct although a wide range of variation occurs. Hiragana is mostly used for function words and conjugational endings of verbs and adjectives (e.g. する *suru* "(to) do"). Katakana, on the other hand, is mostly used for words of foreign (generally Western) origin, onomatopoeic and stressed expressions, and to some extent for plant and animal names (e.g. イチョウ *ichou* "gingko tree"). Katakana characters are also quite commonly used as pronunciation guides for words whose reading is not obvious

---

2. Among the few exceptions to the unique reading rule are kana characters づ and ず which are realized as /zu/ and characters ぢ and じ which are realized as /ʒi/. Here づ and ず are voiced versions of つ *tsu* and す *su*, respectively. Accordingly, ぢ and じ are voiced versions of ち *chi* and し *shi*.

(i.e. uncommon proper names written in kanji or foreign words written in alphabet) [KNI 98]. The kana syllabaries are limited in size and there is a strict correspondence between individual characters and readings. As such, they do not present a major difficulty to the learner of Japanese.

Kanji characters present a much bigger obstacle to the learner, most immediately through a combination of their sheer volume, ideographic nature and phonetic polymorphism. The Japanese government prescribes 1,945 kanji characters for daily use, and up to 3,000 appear in newspapers and formal publications [NLI 86]. Additionally, while the semantics of individual characters often have a bearing on the combined semantics of words in which they occur, they are not marked for phonetic content. That is, there is no way of predicting *a priori* the pronunciation of kanji character 発, for example.[3] Finally, each character can and often does take on several different and frequently unrelated readings. The kanji 発 "emit, depart", for example, has readings including *hatsu* and *ta(tsu)*, whereas 表 "table, exterior, show" has readings including *hyou*, *omote* and *arawa(su)*.

The problem is further complicated due to the existence of character combinations which do not take on compositional readings. For example, 風邪 *kaze* "common cold" is formed non-compositionally from 風 *kaze, fuu* "wind" and 邪 *yokoshima, ja* "evil". Note that every kanji word has a kana equivalent (i.e. reading), which is commonly used in indexing Japanese dictionaries (see Section 2.2).

As mentioned above, when kanji characters are combined to form words, the readings frequently undergo phonological change to give rise to surface readings. The two phenomena that are prevalent in compound formation are sequential voicing (*rendaku*) and sound euphony (*onbin*). **Sequential voicing** is the process of voicing the first consonant of the trailing segment when segments are combined in a binary fashion to produce words. For example, 本 *hoN* "book" is combined with 棚 *tana* "shelf" to give rise to 本棚 *hoNdana* "bookshelf". **Sound euphony** is the process of replacing the last mora (kana character) in the leading segment with a mora in phonetic harmony with the first mora of the trailing segment [FRE 95]. It has several forms, the most common of which is **assimilatory gemination** or *soku onbin*. For example, 国 *koku* "country" combined with 境 *kyou* "boundary" gives rise to 国境 *kokkyou* "(national) border". Notice that sequential voicing occurs in the presence of left lexical context while assimilatory gemination occurs in the presence of right lexical context.

2.2. *Japanese Dictionaries*

Conventional Japanese dictionaries are indexed on the phonemic realization of words, expressed in the form of kana. For example, the kanji compound 発表 *hap-*

---

3. This is not strictly true, as structurally similar kanji characters (e.g. 増, 憎 and 贈) can share a single common reading (*zou* in this case). Even here, however, alternate pronunciations tend to exist and be highly divergent (e.g. 増 "increase" can also be read as *ma(su)* and *fu(eru)*, 憎 "hate" as *niku(mu)* and 贈 "give" as *oku(ru)*).

pyou "announcement" is listed according to its kana-equivalent はっぴょう *happyou*. The phonemic ordering convention makes it easy to look up words in the case that the reading is known due to kana having a natural alphabetic ordering, unlike kanji. However, in many cases it is not straightforward to extract the reading from the word representation as present in a target text. As mentioned above the problem is mostly due to *kanji* characters, whose phoneme realization cannot be easily identified. Generally, each character's reading needs to be learned individually before a word can be looked up in a dictionary. For example, to look up 遷移 *seNi* "transition" the user must know that 遷 and 移 take on the readings *seN* and *i*, respectively. Frequently characters have several unrelated readings which occur in different word contexts (e.g. the readings *seN* and *utsuru* for 遷, and *i* and *utsuru* for 移) making it difficult to postulate the correct reading of the word even if a portion of the readings of each component character are known. When direct lookup fails, words need to be looked up using a different approach.

Kanji dictionaries provide an alternative lookup method aimed at the individual kanji characters. A complicated system of kanji radicals (*bushu*) and stroke counts is used to locate a component kanji in the dictionary (e.g. 遷 could be looked up either via its radical 辶 or stroke count of 15), and the target word is then located from a supplemental listing of words containing that kanji. If the word is not found in the listing, the process must be repeated for other kanji characters present in the word (e.g. 移 could be looked up via its radical 禾 or stroke count of 11). If the word cannot be found through any of the individual kanji,[4] the learner must resort to postulating a compositional reading for the whole word and searching for this reading in a conventional kana dictionary.

To make things worse, the kanji radical and stroke count system leaves a lot of room for error on the part of the uninitiated learner.[5] For example, 遷 also contains the radicals 西 and 己, whereas 移 also contains the radical 夕, potentially leading to confusion as to which radical to look up the character under. Additionally, both 一 and 乙 consist of a single stroke, which is not immediately obvious. Such confusion results in further burdening the lookup task. In some cases lexicographers have tried to expedite the process by devising additional forms of indexing kanji dictionaries (e.g. [HAL 98] boasts six different ways of looking up a character) but these indexing schemes are rarely standardized and in all cases need to be learned to be used. From the above we can see that a user wanting to look up the translation of a word (e.g. "transition" as a translation of 遷移) potentially needs to consult at least two different dictionaries, and search in several passes and through different indexing schemas in order to obtain the translation. Clearly, a system allowing direct and straightforward kanji word lookup would greatly assist the learner by removing or at least ameliorating the difficulties associated with the process of learning new words.

---

4. The word *seNi* is not listed under either character index in [NAG 81] or Sharp Electronic Dictionary PW-9100, while [HAL 98] only lists it under 遷.

5. Some dictionaries get around this problem by listing some characters under several different radical indexes and stroke counts.

2.3. *Existing electronic dictionaries and reading aids*

Above, we painted a bleak picture of Japanese dictionary lookup. However, with the advent of computers and electronic dictionaries, dictionary lookup has become somewhat more efficient. Electronic Japanese dictionaries have become increasingly popular during the last decade both in portable and server-based form due to their superior usability over paper dictionaries. One reason for this is that several different dictionaries (e.g. kanji, monolingual Japanese and bilingual Japanese-English) can be accessed through a single interface, and navigated between easily.

More significant, however, has been the introduction of several new search methods enabling faster lookups. For example, it is possible to copy/paste strings and get the translation directly when the source text is available in electronic form [BRE 00]. Also, most dictionaries support regular expression-based searches allowing for the lookup of words from partial (correct) information (e.g. looking up 遷移 with the glob-style query *seN∗*, or alternatively using kana–kanji conversion to input 遷 based on known readings for 遷). In another development, it has become possible to look up kanji characters via the readings of meaningful sub-units (other than radicals) contained in the character (using, e.g., the Sharp Electronic Dictionary PW-9100 or Canon Word Tank IDF4000).

2.3.1. *Open domain systems*

Also in the last decade, several interactive reading aids aimed at Japanese language learners have become available. A pioneer in this field was the DL system [TER 96], capable of performing morphological analysis of the input sentence and providing translations from the EDICT dictionary [EDI 01]. Similarly to DL, the Reading Tutor[6] [KAW 00, KIT 00] system performs text segmentation and then provides word-level translation and semantic information. Asunaro[7] [NIS 00, NIS 02], on the other hand, provides a multilingual English, Chinese and Thai interface capable of sentence segmentation and displaying parse trees as well as word-level translations. All of these systems aim to help the learner by removing the burden of segmenting sentences into words and converting them into a form suitable for dictionary searches. Syntax trees, semantic information, etc. are added to improve the sentence level comprehension of the target text.

While these dictionaries and reading aids are a valuable addition to the learner's repertoire, they work best when the target text is available in electronic form and needs not be re-entered into the interface. However, in the instance that the text is available only in hard copy, current systems offer very little or no user support. Here, current systems still require that the user has absolute knowledge of the full reading of the word in order to achieve direct lookup. While this is acceptable for proficient Japanese language users, it remains a major handicap for learners of the language.

---

6. `http://language.tiu.ac.jp/`
7. `http://hinoki.ryu.titech.ac.jp/`

2.4. *Problems encountered by Japanese learners*

There is a long history of research documenting the problems Japanese learners have in reading texts containing kanji [NLI 86, MEI 97]. Among the commonly-listed problems are:

1) *Multiple readings for a given kanji.* The learner is aware of the different readings a kanji character can take, but unable to decide on the proper reading in the given context. For example, 大 can be read as either *tai*, *dai* or *oo(kii)*, so the string 大会 *taikai* "convention, congress" could feasibly be misread as *ookai* or *daikai*.

2) *Insufficient knowledge of readings.* The learner is only aware of a proper subset of readings a given kanji can take, and thus cannot predict the correct reading when faced with new words drawing on a novel reading for that kanji. A user aware only of the *oo(kii)* reading for 大, e.g., would almost certainly try to read 大会 as *ookai*.

3) *Incorrect application of phonological and conjugational rules governing reading formation.* For example, 発 *hatsu* and 表 *hyou* form the compound 発表 *happyou* "announcement", but readings such as *hatsuhyou* or *hahhyou* could equally arise from the component character readings.

4) *Confusion as to the length of vowels or consonants.* For example, 主催 *shusai* "organization, sponsorship" can be mistakenly read as *shuusai*, or 最も *mottomo* "most, extremely" as *motomo*. This error type is common in speakers of languages which have no vowel/consonant length distinction.

5) *Confusion due to graphic similarity of different kanji.* Learners with limited contact with kanji can easily confuse characters. For example, 墓 *bo, haka* "grave" and 基 *ki, moto* "base" are graphically very similar, resulting in possible reading substitutions (e.g. between 墓地 *bochi* "graveyard" and 基地 *kichi* "base").

6) *Confusion due to semantic similarity of different kanji.* Characters like 右 *migi* "right" and 左 *hidari* "left" have a similar meaning and as such are often confused, resulting in an erroneous reading. Semantic confusion sometimes occurs at the word level, too, such as between 火事 *kaji* "fire" and 火災 *kasai* "(disastrous) fire".

7) *Confusion due to word-level co-occurrence.* When two characters commonly occur together their readings can be substituted when appearing with other characters. For example, 訴訟 *soshou* "lawsuit" can give rise to the erroneous reading *kishou* for 起訴 *kiso* "indictment". Also common is the superimposition of a known reading onto a word occurring with a common kana suffix, e.g. 慰める *nagusameru* "comfort, console" being read as *osameru* (due to knowledge of the string 修める *osameru* "study, cultivate").

8) *Random errors.* These are errors that do not belong to any of the above groups and are very hard to classify and/or predict. As such, it is hard to imagine a system being able to reliably handle this type of error.

Even though various error types are discussed in previous works [NLI 86, MEI 97], to our knowledge, there exists no previous research that has presented a quantitative analysis of the different error types.

Note that problems 1, 2, 3 and 8 (that is the effects of phonological alternation and phonetic polymorphism) also apply to spelling confusion in English, while all problems other than 4 apply in the case of Mandarin Chinese, for example. That is, English is similar to Japanese in that the same grapheme segment can be read differently in different contexts and phonology produces variable effects, but differs in that it lacks the vowel length and character-level semantic effects of Japanese. Mandarin Chinese is associated with the same basic scope for confusion as Japanese, although the bulk of characters are associated with a unique reading and problems 1 and 2 are therefore considerably less pronounced. In this sense, the Japanese writing system can be seen to be particularly hard for language learners.

3. System Outline

The **FOKS** (Forgiving Online Kanji Search) system aims to aid the learner in coping with the complicated Japanese writing system, and provide direct, linguistically- and statistically-sound support for the types of problems outlined above. The system has a single web-based interface for both known and unknown readings, which allows the learner to look up words directly according to their expected, but not necessarily correct readings. The system is intended to handle both strings in the form they appear in texts (i.e. in kanji) and readings expressed in kana. Given a reading as input, the system tries to establish a relationship between the reading and one or more dictionary entries, and rate the plausibility of each entry being realized with the entered reading.

In a sense, the problem of predicting which word a user seeks from a reading-based input is analogous to kana–kanji conversion (see, e.g., [TAK 96] and [ICH 00]). That is, we seek to determine a ranked listing of kanji strings that could correspond to the input kana string and provide access to the desired word as efficiently as possible. There is one major difference, however. Kana–kanji conversion systems are designed for native speakers of Japanese and as such expect accurate input.[8] In cases when the correct or standardized reading is not available, kanji characters have to be converted one by one. This can be troublesome due to segmentation ambiguity and the large number of characters taking on identical readings, resulting in long lists of kanji characters for the user to choose from.

FOKS does not assume absolutely accurate knowledge of readings, but instead expects readings to be predictably derived from the source kanji. One assumption we unavoidably make is that the user will only try to look up words contained in the base dictionary.[9] That is, we can only hope to direct users to words we have knowledge of, while keeping the number of candidate entries low enough so the user can quickly

---

8. Several kana–kanji conversion systems handle a limited number of input errors (e.g. colloquial readings and substitution of phonologically-indistinguishable kana characters such as づ *zu* and ず *zu*, and ぢ *ji* and じ *ji*). However, as far as we are aware, there is no kana–kanji conversion system that tries to systematically handle a wide range of input errors.

9. The coverage provided by the interface depends solely on the underlying dictionary. The version of FOKS interface publicly available at `http://www.foks.info/` provides access

determine when the desired word is not contained in the dictionary. Assuming we can keep the number of word candidates low enough, users can use a single interface to search for words by either the correct or derivable wrong reading. We return to this point in Section 5.

4. From One Dictionary to Another: the Methodology

While kanji dictionaries list the most common readings each character can take, they give very little additional information that would be useful in our task. For example, most dictionaries provide no information on the relative frequencies of the different readings a character can take, simply listing the readings. Also, while various publications discuss the phonological phenomena affecting the compound reading formation [TSU 96, NLI 84], they do not provide a quantitative analysis which could be used as a starting point for our system. Clearly, given the common readings of the characters it is straightforward to generate compound readings based on the simple concatenation of unit readings. However, if we were to proceed in this manner we would fail to reflect the relationship between the pervasiveness of some readings over others or the phonological effects of compound word reading formation. Hence, this simple approach fails to accomplish our initial goal of modeling the manner in which learners of Japanese are likely to form a candidate reading for a compound word they are not familiar with.

4.1. *Modular approach*

Instead of relying on the data provided in kanji dictionaries, we extract the data directly from the dictionary we are implementing the interface for. We employ a modular approach in dividing the overall problem into several smaller problems and solving each separately. Given the solution and the modularity of the system, each part of the system can be tested separately. The modular nature of our approach is depicted in Figure 1. The process is as follows:

1) Extract the complete set of readings associated with a given segment through a process of grapheme–phoneme alignment.

2) Reduce the obtained reading set by separating the genuine differences in readings from those which are phonological and/or conjugational derivations of underlying base readings in the process of canonization.

3) Exhaustively generate new readings for each dictionary entry and calculate their overall probability based on the probabilities of segment readings and corpus frequencies.

Below we describe each of the modules in detail.

---

to over 100,000 entries in the EDICT general use dictionary and over 200,000 entries in the ENAMDICT proper noun dictionary.
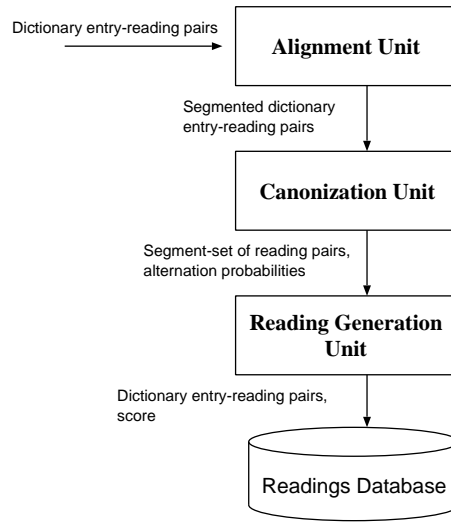
```
Dictionary entry-reading pairs
  ────────────────────────────▶   [ Alignment Unit ]
                                          │
                  Segmented dictionary    │
                  entry-reading pairs     │
                                          ▼
                                  [ Canonization Unit ]
                                          │
            Segment-set of reading pairs, │
            alternation probabilities     │
                                          ▼
                                  [ Reading Generation
                                          Unit ]
                                          │
          Dictionary entry-reading pairs, │
          score                           │
                                          ▼
                                  ( Readings Database )
```

Figure 1. *The modular structure of the FOKS system*

4.2. *Grapheme–phoneme alignment*

Given a dictionary entry and its reading given in hiragana, we want to extract the part of the hiragana reading resulting from each kanji character, that is align the kanji (grapheme strings) with their readings (phoneme strings). For example, given the compound 解析 *kaiseki* "analysis", we would like to identify 解 as having contributed a reading of *kai* and 析 a reading of *seki*, accounting for the word-level reading of *kaiseki*. We remind the reader that hiragana characters are not strictly phonemes, but phoneme clusters. Nonetheless, in our application the leap is permissible. In the alignment process, we attempt to extract the complete set of phoneme realizations (component readings) for each grapheme segment (kanji segment). The particular dictionary used here and throughout the research is the publicly-available EDICT dictionary [EDI 01]. Following the same alignment procedure for all dictionary entries containing a given kanji, we can extract a complete set of phonemic realizations of the kanji. [BAL 00] give a comparison of several machine-learning based methods as applied to unsupervised alignment. The method described below proved superior in accuracy when no alignment training data is available. It requires no supervision and could be applied to other languages in which the phonetic realization is not clearly derivable from the grapheme presentation. The alignment process proceeds as follows:

1) For each grapheme–phoneme string pair, generate a complete set of candidate alignment mappings. We constrain the alignment process by requiring that each grapheme character aligns to at least one character in the phonemic representation,

that the alignment is strictly linear (and non-intersective) and that characters are indivisible.

2) Prune candidate alignments through the application of linguistic constraints such as requiring segment boundaries at script boundaries,[10] direct alignment of kana equivalents and indivisible syllables. When multiple candidates exist, we also prune the candidates with multiple voiced obstruents in a reading segment [BAL 99].

3) Score each alignment by a variant of the TF-IDF model [SAL 90], which was developed for term weighing in information retrieval.

4) Iteratively work through the data selecting a single grapheme–phoneme string pair to align according to the highest-scoring candidate alignment at each iteration, and updating the statistical model accordingly (to filter out disallowed candidate alignments and score up the selected alignment mapping).

Examples of alignments extracted by our algorithm are: [11]

発表 ⟨*happyou*⟩ "announcement" ⇒ 発 | 表 ⟨*hap* | *pyou*⟩
割り引き ⟨*waribiki*⟩ "discount"⇒ 割り | 引き ⟨*wari* | *biki*⟩
風邪薬 ⟨*kazegusuri*⟩ "cold medicine" ⇒ 風邪 | 薬 ⟨*kaze* | *gusuri*⟩

4.3. *Canonization*

The alignment data contains all possible readings for a given grapheme segment as were available in the context of a dictionary used for alignment. It can include alternates due to sequential voicing, sound euphony and conjugation (e.g. phonological variants of *hyou* and *byou* for 表 *chart*, and the conjugational variants of *yomi* and *yomu* for the verb 読 *read*), and possibly (but not necessarily) the base form of each reading. We canonize the readings to separate the base reading data apart from the alternation derived data, thus minimizing the number of reading types and maximally extracting instances of alternation. This provides a means of overcoming data sparseness and allows us to produce unobserved segment-level readings through novel alternation combinations over the base readings and thus increase the coverage of predicted readings.

We observed above that sequential voicing occurs only when the given segment has left lexical context and that sound euphony occurs only in the presence of right lexical context. To detect the two phenomena, therefore, we can classify segments into 4 groups according to the presence of left and right lexical context [BAL 02].

a) Level 0 (−left, −right context): no possibility of conjugation or phonological alternation

---

10. With the exception of kanji-hiragana boundaries which are not enforced due to conjugative suffices of verbs and adjectives always being expressed in hiragana (i.e. *okurigana*) but forming a single lexical unit together with the head kanji character.

11. Notice that in some cases, grapheme segments can be made up of more than one kanji character, as occurs for 風邪 *kaze* "common cold" above.
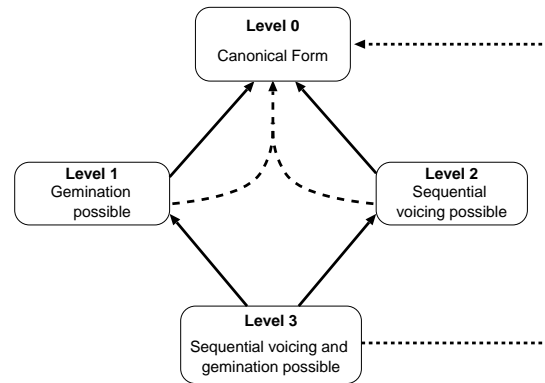
Figure 2. *Canonization flowchart*

b) Level 1 (−left, +right context): possibility of gemination or conjugation

c) Level 2 (+left, −right context): possibility of sequential voicing

d) Level 3 (+left, +right context): possibility of all of gemination or conjugation, and sequential voicing

Level 0 singleton segments can be assumed to comprise the base readings from which readings at other levels are derived. Quite commonly, readings are derived through zero-derivation, whereby no phonetic/conjunctive alternation takes place. We work through the various levels in decreasing numeric order, and determine whether a unique base reading exists for each grapheme segment from which the observed reading has been derived. In the case that such an analysis is possible, we record the type of alternation and update its frequency by incrementing the frequency of the alternation by the frequency of the string in which alternation was found to occur, combining it with that of the base reading. In the case that multiple matches are found for variants of the original reading with identical kanji content, the frequency of the original kanji–reading string is distributed equally between all matching entries. The canonization process is depicted in Figure 2.

First, we perform conjugational analysis [BAL 98] at Levels 1 to 3 to establish whether each segment has an underlying verbal or adjectival form. At each step, we then perform a match over both the original form and the base conjugational form(s) of the reading. This distribution of frequency extends to any phonological alternation or conjugation associated with each match.

Next we attempt to merge Level 3 entries with Level 1 and 2 entries, and then Level 1 and 2 entries with Level 0 entries. The reason for this particular ordering of the canonization process is that, where possible, we wish to isolate the effects of a single phonological process at a time to maintain analytical consistency throughout the canonization process. Many segments do not occur at Level 0 (i.e. as stand-alone

characters) but can be found in multiple instances at other levels. For example, 発 *hatsu* "emit" occurs at all of Levels 1 (e.g. 発 表 *happyou* "presentation"), 2 (e.g. 原 発 *geNpatsu* "nuclear power") and 3 (e.g. 未 発 行 *mihakkou* "unpublished"), but not level 0. We thus have no immediate indication of its canonical form, but based on the alignment data we know that it takes readings ha_G[12] and *patsu*. In this example, the Level 3 reading of ha_G is not voiced but has undergone gemination, meaning it is not in canonical form. Since we have no instances of unvoiced, non-geminate candidates at Level 3, we postpone disambiguating the canonical form and merge ha_G with the existing Level 1 reading. This leaves us with two readings: ha_G at Level 1 and *patsu* at Level 2. The canonical form for ha_G can be any one of the *hatsu, hachi, haku, haki,* etc. On the other hand, *patsu* is semi-voiced, and is therefore either the canonical form in itself or derived from the voiced *batsu* or unvoiced *hatsu*. Through the interaction of Levels 1 and 2, we can determine that both readings are derived from the canonical form *hatsu* so we record them as such and update the corresponding frequencies. In the case that no merging of readings is possible through the canonization process, each reading is promoted to Level 0 as a separate reading type.

After canonization, our data from above would look as follows:

発 | 表 $\langle hap \mid pyou \rangle \Rightarrow \langle hatsu \mid hyou \rangle$ +gemination +voicing
割り | 引き $\langle wari \mid biki \rangle \Rightarrow \langle wari \mid hiki \rangle$ +voicing
風邪 | 薬 $\langle kaze \mid gusuri \rangle \Rightarrow \langle kaze \mid kusuri \rangle$ +voicing

While canonizing the readings, we keep track of cases where genuine alternation took place (cases where entries at different levels were successfully merged together based on a conjugation, gemination and/or sequential voicing analysis) so as to be able to reapply them as independent probabilities below. Also we count the number of occurrences of each reading for a given kanji segment and convert this number into the probability of the given kanji segment taking each reading $P(r|k)$. Notice that this probability depends on the kanji character in question, unlike the probability of voicing and gemination alternations which depend on the reading realization of the segment in question. We further extend the set of alternations we consider with vowel shortening/lengthening, the probability of which is calculated as the percentage of short/long vowels in our dictionary set multiplied by a weight factor.[13]

4.4. *Reading Generation and Scoring*

After extracting the set of segment readings and calculating the various alternation probabilities, we proceed to generate and score plausible readings. The first step in this process is to segment up the target string, so as to be able to look up readings for the individual segments and compose these into an overall reading. For the string

---

12. Where "$_G$" indicates the final kana syllable has been geminated, i.e. ha_G equates to the kana-form はっ.
13. In all experiments described in Section 5 we use a weight factor of 0.05 for both vowel shortening and lengthening.

Function: SegmentReading()

BEGIN
   Input:
      s segment
      $R_s = \{(r_1, P_1), (r_2, P_2)....(r_k, P_k)\}$ where (r_i, P_i) is a reading, probability tuple
      $A = \{(a_1, a_2, ...a_j)\}$ where a_i is an alternation with probability P_{a_i}

    for ii from 0 to $k - 1$ do
      $(r_c, P_c) \leftarrow (r_{ii}, P_{ii})$
      for i from 0 to $j - 1$ do
        $r_{new} \leftarrow a_i(r_c)$
        $P_{new} \leftarrow P_c \times P_{a_i}$
        if $\exists (r, P)$ s.t. $(r, P) \in R_s \bigwedge r = r_{new}$
          $P \leftarrow P_c + P_{new}$
        else
          $R_s \leftarrow R_s \bigcup \{(r_{new}, P_{new})\}$
      end do
    end do

    normalize R_s s.t. $\displaystyle\sum_{i=0}^{n} P_i = 1 \quad \forall (r_i, P_i) \in R_s$

    return R_s
END

Figure 3. *Pseudo-code for the **SegmentReading** function*

発表する *happyousuru* "to present", for example, we would ideally partition it into the three segments 発, 表 and する; for the non-compositional 風邪 *kaze* "common cold", a single-segment analysis may be more appropriate. We test two segmentation methods, based on bigram probabilities and script boundaries.

The bigram-based method consists of taking each character bigram in the target string and using the grapheme–phoneme alignment data to rate the probability of a segment occurring at that point. As noted above, katakana and hiragana strings take a unique kana-based reading, irrespective of how we segment them up. We thus chunk all contiguous hiragana and katakana characters (and alpha-numeric strings) together into a unigram unit. The output of this method is a set of different string segmentations, each of which is associated with a probability based on the product of the bigram probabilities at each potential segment insertion point.

The script boundary segmentation method adopts a much simpler approach, in inserting a segment marker at each script demarcation point (e.g. between each kanji and kana character), and additionally inserting a segment between each pair of kanji characters. This segmentation schema results in a considerable simplification of the generation process, and produces a unique segmentation of a given string. This comes at the cost of preventing generation of the correct reading for multi-kanji segments (e.g. 風邪 *kaze* "common cold" from above).

Having segmented the strings, we next generate scored readings according to the following steps:

1) For each segment in word W, use the previously calculated set of readings R containing reading–probability tuples $(r,\ P(r|k))$ and expand it to include any additional readings resulting from application of alternations under consideration. For each applicable alternation a, we calculate a new tuple (r_{new}, P_{new}) where P_{new} is calculated under assumption of segment independence as in equation (1) and r_{new} is the resulting reading. If the reading was in the set originally, the probabilities are added and if not the new tuple is inserted into the reading set. After the complete set of reading–probability tuples is obtained we normalize the probabilities to sum to 1. Figure 3 gives the algorithm for generating a complete set of readings for a segment.

$$P_{new} = P_{(r|k)} \times P_a \tag{1}$$

2) Create an exhaustive listing of reading candidates r_W for each dictionary entry W by concatenating individual segment readings and calculate the probability $P(r_W|W)$ of each based on the evidence from step 1 and the naive Bayes model (assuming independence between all parameters) as given by equation (2). Figure 4 gives the simplified recursive version of the generation algorithm. The actual implementation is iterative and optimized to avoid unnecessary repetitive calculations.

$$
\begin{aligned}
P(r_W|W) &= P(r_{1..n}|s_{1..n}) \\
&= \prod_{i=1}^{n} P(r_i|s_i) \tag{2}
\end{aligned}
$$

While generating readings we apply a probability threshold keeping only the readings with a higher probability. Then, we normalize the probabilities of the pruned set of readings to sum to 1.

$$
\begin{aligned}
P(W) &= \frac{F(W)}{\sum_i F(W_i)} \tag{3} \\
P(W|r) &= P(W)\frac{P(r|W)}{P(r)} \\
&= \frac{F(W)}{\sum_i F(W_i)}\frac{P(r|W)}{P(r)} \tag{4}
\end{aligned}
$$

3) Calculate the corpus-based frequency $F(W)$ of each dictionary entry W in the corpus and then convert it into a string probability $P(W)$, according to equation (3). Notice that the term $\sum_i F(W_i)$ depends on the given corpus and is constant for all strings W in a same corpus. Use Bayes rule to calculate the probability $P(W|r)$ of each resulting reading according to equation (4). Here, as we are only interested in the relative score for each W given an input r, we can ignore $P(r)$ and the constant $\sum_i F(W_i)$. The final plausibility grade of a user searching for dictionary entry W by querying with reading r is thus estimated as in equation (5).

Function: WordReading()

BEGIN
    Input:
        $S[1, 2, ...n]$ where $S[i]$ is a segment
        $L[1, 2, ...n]$ where $L[i]$ is a set $R_{S[i]}$ of readings of $S[i]$ with associated probabilities
        $A = \{a_1, a_2, ...a_j\}$ where a_i is an alternation with probability P_i
    $R \leftarrow \text{SegmentReading}(S_1, L_1, A)$
    if $n > 1$
        $R \leftarrow R \otimes \text{WordReading}(S[2, ..., n], L[1, 2, ...n], A)$
            where $R_1 \otimes R_2 = \{<r, P> \mid \quad r = concat(r_1, r_2), P = P_1 * P_2$
                                  $\bigwedge <r_1, P_1> \in R_1 \bigwedge <r_2, P_2> \in R_2\}$

    prune R s.t. $R = \{(r, P) \quad \mid \quad P > P_{threshold}\}$
    normalize R s.t. $\sum_{i=0}^{n} P_i = 1 \quad \forall (r_i, P_i) \in R_s$
    return R
END

Figure 4. *Pseudo-code for the* **WordReading** *function*

$$Grade(W|r) = P(r|W) \times F(W) \qquad [5]$$

4) To complete the reading set we insert the correct readings for all dictionary entries W_{kana} that did not contain any kanji characters and for which no readings were generated above, with plausibility grade calculated by equation (6).[14]

$$Grade(W_{kana}|r) = F(W_{kana}) \qquad [6]$$

Furthermore, if the generation step failed to generate a correct reading for the dictionary entry containing kanji, we add it to the reading set since we want to assure the ability to search for a dictionary entry by its correct reading.

4.4.1. *Failure to generate the correct reading*

Even though we start out with a correct dictionary reading as the input to our system, it can fail to generate a correct reading for a dictionary entry due to one of the following reasons:[15]

a) *Incorrect segmentation.* When the initial segmentation of multi-kanji units is incorrect, it can obstruct generation of the correct reading. For example, if the initial segmentation of お土産 *omiyage* "souvenir" is 〈 お | 土 | 産 〉 the system may be unable to generate the correct reading since it is not composed of individual character readings.

---

14. Here, $P(r|W_{kana})$ is assumed to be 1, as there is only one possible reading (i.e. r).
15. In the worst case of experimental generation, the system failed to generate correct readings for 6277 readings

b) *Threshold probability.* In some cases, the correct reading is generated with a very low probability and filtered out as part of the pruning. However the pruning is necessary since during the test runs of our generation algorithm, we run into problems with very large numbers of readings being generated for each dictionary entry, resulting in our reading database growing beyond available disk capacity.

c) *Grapheme gapping.* Gapping takes place when certain part of the phoneme string is omitted from the grapheme string. For example, 山手 *yama<u>note</u>* "uptown" is commonly written without the *no* segment, whereas the more complete representation would be 山の手. The correct reading cannot be created since the the system cannot account for the gapped segment.[16]

d) *Alpha-numeric characters.* When dictionary entries contain alpha-numeric characters in the grapheme string the phoneme equivalent usually contains the transcribed kana equivalent (e.g. ＡＢＣ順 *eebiishiijuN* "alphabetic order" and １１０番 *hyakutoobaN* "emergency telephone number"[17]) but our system does not generate such transcriptions.

By default, we set the probability of such correct readings to equal the threshold probability applied in filtering readings during generation and calculate the score of the reading according to equation (5) as before.

Note again that all three stages of the above processing are fully automated, a valuable quality when dealing with a volatile dictionary such as EDICT. With minor modifications it should be possible to apply our methodology to a different language where phoneme representation is not clearly derivable from the grapheme representation.

5. Evaluation

Starting with the EDICT dictionary, we proceeded through the steps described in Section 4 to generate new sets of scored readings with the corpus frequencies from the complete set of 200,000+ sentences in the EDR Japanese corpus [EDR 95]. Then we implemented a web based interface with pregenerated reading sets accessible through a CGI interface.[18] Consequently, we are able to provide real-time dictionary look up without additional computational overhead. The currently available implementation covers the first four types of errors described in Section 2.4.

Here, we will provide an evaluation carried out with two basic goals in mind: (a) to evaluate the effectiveness of the proposed system in handling queries with erroneous

---

16. However, grapheme gapping is relatively infrequent phenomena appearing in only 0.1% of the 5000 randomly chosen dictionary entries used for alignment evaluation. As such, it does not significantly affect system performance

17. Here *eebiisii* is the Japanese pronunciation for ABC and *hyakutoo* is an idiosyncratic pronunciation for 110

18. The system is freely available at `http://www.foks.info/`

readings, and (b) to examine the effect additional search options and the size of the reading set have on the users ability to find the desired entry.

5.1. *Data sets*

From the outset of our project, we were faced with the problem of finding a collection of naturally-occurring reading errors that could be used to evaluate the FOKS system. While there was a lot of information on types of errors made by learners of Japanese (see Section 2.4), we were unable to locate a database of recorded naturally-occurring reading errors. Instead, we look to two other sources for test data sets.

The first source is a set of practice problems for the Japanese Proficiency Test [SUZ 96, MAT 95]. The Japanese government has established a four-level certification program aimed at evaluating the ability of non-native learners of Japanese in reading comprehension, listening and vocabulary. We have collected a number of different books used for the preparation for the proficiency exam and extracted 420 level 2 word reading problems. Each problem consists of a word given in its normal kanji form, with four potential readings in kana, only one of which is correct. During the test, the examinee is requested to choose the correct reading from among the four candidates. Here are some example words with candidate readings:[19]

訴訟 *soshou* "lawsuit":	*sousho*	*soushou*	*sosho*
傾く *katamuku* "lean":	*muku*	*kizuku*	*uchiaku*
肉親 *nikushiN* "blood relative":	*nikuoya*	*nisshiN*	*nikuya*
煙 *kemuri* "smoke":	*honoo*	*hi*	*susu*

The second set of data is a collection of 139 entries taken from a web site displaying real-world reading errors or *godoku* "wrong reading" made by native speakers of Japanese.[20] Each entry consists of a word given in kanji–kana combination and one incorrect and correct reading each. These entries were compiled from various sources and as such should reflect the wide variety of possible reading errors.

For both data sets we changed all the verb and adjective forms to basic dictionary form for both the word and all of its potential readings to make them appropriate for dictionary querying.

5.2. *Comparison with a conventional system*

We first created four databases of readings: (a) two using the bigram segmentation model (labeled "Bi" in consequent tables) trained on extracted alignment data and (b) the other two using the kanji–script boundary segmentation model (labeled "Ka" in

---

19. Here we give the correct reading in the gloss. In the actual test, the correct readings can be at any of the four positions.
20. `http://www.sutv.zaq.ne.jp/shirokuma/godoku.html`

	Conv.	Bi (1×10^{-3})	Bi (1×10^{-4})	Ka (1×10^{-3})	Ka (1×10^{-4})
Total readings	97,927	3,449,866	8,864,800	4,549,152	13,812,273
Size (MB)	1.3	116	314	164	534
Unique readings	77,627	3,005,900	8,553,828	4,543,893	13,807,014
Ave. R/E	1.03	36.37	93.46	47.96	145.62
Ave. E/R	1.26	1.30	1.26	1.21	1.14
Max. R/E	6	821	5394	317	2223
Max. E/R	27	162	182	167	189

Table 1. *Basic breakdown of different sets of readings*

consequent tables) putting each kanji character in a separate segment (see Section 4.4). For each model we used two different thresholds, 1×10^{-3} and 1×10^{-4} respectively. The basic breakdown of these sets is given in Table 1.[21]

Given the two data sets and four reading sets we ran the following experiment for each combination. For each entry we queried the system with the correct and then the incorrect readings. As a baseline we used direct matching over the base EDICT dictionary to mimic a conventional system. When executing the query we counted the number of results and whether the desired entry was among the candidates returned. Provided that the system successfully returned the desired word as a candidate we also counted its rank. In some cases, the word was not contained in the dictionary so we excluded it from the evaluation. The results of these experiments are given in Tables 2 and 3. In each table, we give the error reduction rate, calculated according to equation (7). This rate reflects the improvement over the conventional system.

$$Error\,Red. = \frac{Successful\ Queries - Baseline\ Successful}{\#\ Queries - Baseline\ Successful} \qquad [7]$$

For each test run we also give the Mean Rank and the Relative Normalized Mean Rank (RNM Rank) which exemplify how high the desired entry is ranked in the candidate listing and how the rank depends on the number of candidates, respectively. RNM Rank is calculated according to equation (8). The lower this value is, the better. In an ideal system, the desired entry would always rank 1st so the RNM Rank would be 0.

$$RNMR = \frac{\sum_{i=0}^{n} \dfrac{Rank\ of\ Candidate - 1}{Number\ of\ Candidates - 1}}{n} \qquad [8]$$

---

21. In this table R stands for readings and E for dictionary entries.

	Conv.	Bi (1×10^{-3})	Bi (1×10^{-4})	Ka (1×10^{-3})	Ka (1×10^{-4})
# Queries	1189	1189	1189	1189	1189
Ave. # Results	2.26	10.37	14.58	11.03	15.36
Successful	18	484	512	547	574
Error Red. (%)	0	39.80	42.19	45.18	47.48
Mean Rank	1.66	1.96	2.05	1.84	1.94
RNM Rank	0.22	0.08	0.07	0.07	0.06

Table 2. *Results for Level 2 words*

	Conv.	Bi (1×10^{-3})	Bi (1×10^{-4})	Ka (1×10^{-3})	Ka (1×10^{-4})
# Queries	77	77	77	77	77
Ave. # Results	1.53	7.85	10.96	8.22	11.48
Successful	10	38	39	51	55
Error Red. (%)	0	41.79	43.28	61.19	67.16
Mean Rank	1.4	3.58	3.90	3.16	3.36
RNM Rank	0.18	0.20	0.24	0.18	0.19

Table 3. *Results for godoku words*

From Table 2, we can see that our system is able to handle a large number of erroneous readings as compared to the conventional system. The error rate reduction ranges from 39.80% to 47.48%. The conventional system is able to handle 18 readings due to the fact that those readings might be appropriate in different contexts and as such are recorded in the dictionary. However, different readings usually coincide with different meanings (and hence translations). Due to the nature of the conventional search, the user would not be aware of alternate readings/translations not returned by the system. In our system, on the other hand, we offer a list of all potential readings and translations for the user to choose from so the user can make the decision as to which translation is appropriate in the given context.

From the error reduction rates, we can see that the "Ka" segmentation method results in better coverage of erroneous readings even for lower threshold values and smaller reading sets. Furthermore, the mean rank of the desired entry among the candidates returned is also lower than for the "Bi" segmentation model. As expected, as we decrease the cut-off threshold, the number of successfully handled queries rises, as does the average number of candidates returned. Nonetheless, the Mean Rank and RNM Rank are both quite low, showing that the desired entry on average ranks high in the candidate list.

Looking to Table 3, we can see that the error rate reduction is even higher, with the maximum improvement reaching 67.16% for the largest generated set. We can also

	Level 2	*godoku*
Queries	1189	77
Successful	587	55
Previous Best	574	55
Coverage Increase	13	0
Error Reduction (%)	48.59	67.16

Table 4. *Query results for readings sets generated with no threshold applied*

see that the number of candidates returned is somewhat lower and that the average rank of the desired entry is higher. This can be explained by the fact that the average character length of erroneous readings in this set was 4.35 characters as opposed to 2.49 characters for Level 2 readings. Elsewhere, we have established that the number of results returned is smaller for longer queries [BIL 02].

Here are several examples of successfully-handled erroneous queries, with the numbers in brackets corresponding to the error types in Section 2.4:

ryuushu ⇒ 留守 *rusu* "absence" [1,2]
zeki ⇒ 世紀 *seiki* "century" [1,3]
koki ⇒ 後期 *kouki* "second half" [4]

As seen above, decreasing the probability threshold to prune the generated readings increases the coverage of the erroneous readings. Nonetheless, we wanted to see if there is an upper limit to the error coverage, so we ran an additional experiment in which we created an exhaustive reading set for all the words in our data sets without applying any threshold; we used the same queries on this set as we did in the previous experiments. The results are depicted in Table 4. We can see that the system can handle 13 more queries for the Level 2 words than the previous best result on sets generated with a threshold, but that the total remains the same for the *godoku* words. Note that this increase in coverage comes with an exponential increase in the reading set size (178MB for 764 entries) which prevents generation and storage of the complete reading set for the whole dictionary.

From the above data, it would appear that the "Ka" segmentation schema results in higher coverage of erroneous readings. Recall that this segmentation schema forces each kanji character into a separate segment and inserts segment boundaries at each script boundary.

5.2.1. *Remaining Problems*

We analyzed the set of readings that none of the systems tested could handle, and found a number of systematic problems with our system arising from the manner in which we generate readings without accounting for all aspects causing reading errors. In the Japanese proficiency test data, the incorrect readings are commonly readings of semantically similar words. The word 煙 *kemuri* "smoke" has reading candidates

such as *honoo* "flame" and *hi* "fire". Also, the readings are often borrowed from the words with the same trailing kana content. For example, 定める *sadameru* "set" has a candidate reading of *kimeru* (derived from 決める *kimeru* "decide") due to the common suffix (*meru*). While we have discussed these phenomena in the context of common reading errors, they are presently not included in our generative model and consequently result in unsuccessful searches.

In the *godoku* reading data, two other common types of error presented a problem for our system. The majority of entries we could not handle were the result of confusion due to graphical similarity. For example, 御札 *ofuda* "talisman" can be confused with 御礼 *orei* "thanks, gratitude". Another common problem is kanji strings being interpreted as proper names, hence taking on unusual readings. Although we have implemented an analogous system for searching the EDICT proper noun dictionary with readings derived from common words, we currently offer no solution for the opposite problem of regular words being interpreted as proper names.

While we recognize that our system still has deficiencies at present, our experiments have shown that it significantly increases dictionary accessibility in the case that the prescriptive reading is not available, and as such should aid the learner of Japanese. Admittedly, this evaluation was over data sets of limited size, largely because of the difficulty in gaining access to naturally-occurring kanji–reading confusion data. The results are, however, promising as far as both coverage and appropriateness of the scoring function are concerned.

5.3. *Reading set analysis*

Since we create a large number of plausible readings, a potential problem was that a large number of candidates would be returned for each reading, obscuring dictionary entries for which the input is the correct reading and penalizing competent users who mostly search the dictionary with correct readings. Therefore, we tried to establish how many candidates are likely to be returned for an arbitrary user query. Due to space constraints we only look at the smaller "Ka" set with the 1×10^{-3} threshold.[22]

The distribution of number of word entries returned for the full range of reading types generated by the proposed method is given in Figure 5. In this figure, **Baseline** represents the readings in the original dictionary, the distribution of which is calculated over the original dictionary. **Existing** is the subset of readings in the generated set that existed in the original dictionary, and **All** is all readings in the generated set. The distribution of the latter two sets is calculated over the generated set of readings. The x-axis represents the number of results returned for the given reading and the y-axis represents the natural log of the number of readings returning that number of results. It can be seen that only a few readings return a high number of entries. 943 out of 4,543,893 or 0.02% of the readings return over 30 results. Note that the average number of dictionary entries returned per reading is 1.21 for the complete

---

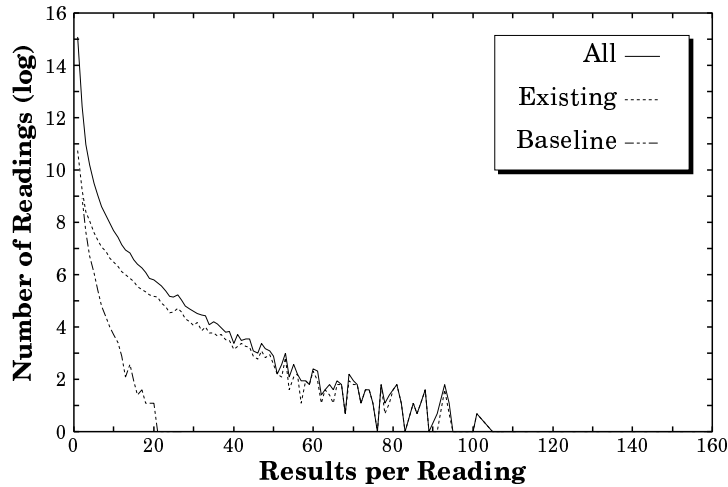22. This is the set that is accessible at `http://www.foks.info/`

Figure 5. *Distribution of results returned per reading for the "Ka" model*

set of generated readings. When queries are always the correct readings, the average number of entries returned is 3.23.

Above we have provided evidence underlining the ability of the system to direct the user to the desired dictionary entry from a wrong reading, and with minimal filtering of spurious hits. One outstanding point of interest is the relative speed-up in dictionary lookup the system offers to a language learner, which we leave for future research.

6. Discussion

In order to emulate the limited cognitive abilities of a language learner, we have opted for a simplistic view of how individual kanji characters combine in compounds. In step 2 of reading generation, we use the naive Bayes model to calculate an overall probability for each reading, and in doing so assume that component readings are independent of each other, and that phonological and conjugational alternation in readings does not depend on lexical context. Clearly this is not the case. For example, kanji readings deriving from Chinese and native Japanese sources (*on* and *kun* readings, respectively) tend not to co-occur in compounds. Furthermore, phonological and conjugational alternations interact in subtle ways and are subject to a number of constraints [VAN 87].

However, depending on the proficiency level of the learner, s/he may not be aware of these rules, and thus may try to derive compound readings in a more straightforward fashion which is adequately modeled through a simplistic independence model. As can be seen from our experiments our system is effective in handling a large number

of predictable reading errors, therefore justifying the soundness of our model. While the results vary according to the test data and the size of the generated reading set, our system outperforms the conventional system in all our experiments performed. Furthermore, the number of responses is on average low enough (less then 4) that it does not inhibit the usefulness of the improved search ability.

Nonetheless, the cognitive model can be improved further. We intend to modify it to incorporate further constraints in the generation process after observing the correlation between the inputs and selected dictionary entries. To this end, we are collecting usage data from our servers and feedback from users. Furthermore, as briefly pointed out in Section 5.2.1, the current cognitive model still does not cover all types of reading errors, with graphic and semantic similarity being notable sources of error currently not handled. The problem with including these types of errors into our cognitive model is that it is not straightforward to quantify them. In the case of graphic similarity, limited research has been conducted on analyzing kanji similarity at the stroke level [MAE 02] but the coverage is still too limited for general purpose dictionaries. On the other hand, semantic similarity has received a lot of attention in research on disambiguation and lexicography, but still remains one of the larger obstacles in NLP in general. Note that we have taken tentative steps towards handling reading confusion due to word-level co-occurrence, as detailed in [BIL 03].

Finally, all the work on this dictionary interface is conducted under the assumption that the target string is contained in the original dictionary and thus we base all reading generation on the existing entries, assuming that the user will only attempt to look up words we have knowledge of. The system allows for motivated reading errors, but it provides no immediate solution for random reading errors or for cases where user has no intuition as to how to read the characters in the target string.

7. Conclusions and Future Work

In this paper we have described FOKS, a system designed to accommodate user reading errors and supplement partial knowledge of the readings of Japanese words. Our method takes dictionary entries containing kanji characters and generates readings for each, scoring them for plausibility in the process. These scores are used to rank the different word entries with generated readings corresponding to the system input. The proposed system is web-based and freely accessible. Initial evaluation indicates significant increases in robustness over erroneous inputs.

8. References

[BAC 94] BACKHOUSE A. E., *The Japanese Language: An Introduction*, Oxford University Press, 1994.

[BAL 98] BALDWIN T., "The Analysis of Japanese Relative Clauses", Master's Thesis, Tokyo Institute of Technology, 1998.

[BAL 99] BALDWIN T., TANAKA H., "The Applications of Unsupervised Learning to Japanese Grapheme-Phoneme Alignment", *Proc. of ACL Workshop on Unsupervised Learning in Natural Language Processing*, College Park, USA, 1999, p. 9–16.

[BAL 00] BALDWIN T., TANAKA H., "A Comparative Study of Unsupervised Grapheme-Phoneme Alignment Methods", *Proc. of the 22nd Annual Meeting of the Cognitive Science Society (CogSci 2000)*, Philadelphia, USA, 2000, p. 597–602.

[BAL 02] BALDWIN T., BILAC S., OKUMURA R., TOKUNAGA T., TANAKA H., "Enhanced Japanese Electronic Dictionary Look-up", *Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Spain, 2002, p. 979–985.

[BIL 02] BILAC S., BALDWIN T., TANAKA H., "Bringing the Dictionary to the User: the FOKS system", *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, 2002.

[BIL 03] BILAC S., BALDWIN T., TANAKA H., "Increasing the error coverage of the FOKS Japanese dictionary interface", *Proc. of ASIALEX 2003*, Tokyo, Japan, 2003, (to appear).

[BRE 00] BREEN J., "A WWW Japanese Dictionary", *Japanese Studies*, vol. 20, 2000, p. 313–317, Japanese Studies Association of Australia.

[EDI 01] EDICT, "EDICT Japanese-English Dictionary File", `ftp://ftp.cc.monash.edu.au/pub/nihongo/`, 2001.

[EDR 95] EDR, *EDR Electronic Dictionary Technical Guide*, Japan Electronic Dictionary Research Institute, Ltd., 1995, (In Japanese).

[FRE 95] FRELLESVIG B., *A Case Study In Diachronic Phonology, The Japanese Onbin Sound Changes*, Aarhus University Press, 1995.

[GRO 00] GROOT P. J. M., "Computer Assisted Second Language Vocabulary Acquisition", *Language Learning & Technology*, vol. 4, num. 1, 2000, p. 60–81.

[HAL 98] HALPERN J., Ed., *New Japanese-English Character Dictionary*, Kenkyusha Limited, 6th edition, 1998.

[HUM 01] HUMBLE P., *Dictionaries and Language Learners*, Haag + Herchen, 2001.

[ICH 00] ICHIMURA Y., SAITO Y., KIMURA K., HIRAKAWA H., "Kana-Kanji Conversion System with Input Support based on Prediction", *Proc. of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany, 2000, p. 341–347.

[KAW 00] KAWAMURA Y., "The Role of the Dictionary Tools in a Japanese Language Reading Tutorial System", *Ljubljana University International Seminar*, Ljubljana, Slovenia, 2000, (In Japanese).

[KIT 00] KITAMURA T., KAWAMURA Y., "Improving the dictionary display in a reading support system", *International Symposium on Japanese Language Education*, Seoul, Korea, 2000, (In Japanese).

[KNI 98]   KNIGHT K., GRAEHL J., "Machine Transliteration", *Computational Linguistics*, vol. 24, 1998, p. 599–612.

[LAU 01]   LAUFER B., HULSTIJN J., "Incidental Vocabulary Acquisition in a Second Language: The Construct of Task-Induced Involvement", *Applied Linguistics*, vol. 22, 2001, p. 1–26.

[MAE 02]   MAEDA K., TATSUOKA R., HOKADA K., OSHIKI H., "Development of a Kanji Learning System toward providing Optimal Learning Materials", *Proc. of SNLP-Oriental COCOSDA 2002*, Hua Hin, Thailand, 2002, p. 243–249.

[MAT 95]   MATSUOKA T., *Problems from Japanese Proficiency Test, characters and vocabulary (Levels 1 and 2)*, Kokusyo Kankoukai, 1995.

[MEI 97]   MEIJI M., *Analysis of misuse of Japanese Language*, Meiji Publishing, 1997, (In Japanese).

[NAG 81]   NAGASAWA K., Ed., *Shinmeikai Kanji-Japanese Character Dictionary*, Sanseido Publishing, 2nd edition, 1981.

[NIS 00]   NISHINA K., OKUMURA M., SUGIMOTO S., YAGI Y., ABEKAWA T., TOTSUGI N., RYANG F., "Development research on multilingual Japanese reading aid for foreign students with scientific background", *Research Report of Telecommunications Advancement Foundation*, vol. 15, 2000, p. 151-159, (In Japanese).

[NIS 02]   NISHINA K., OKUMURA M., YAGI Y., TOTSUGI N., RYANG F., SUGIMOTO S., ABEKAWA T., "Development of Japanese Reading aid with a multilingual interface and syntax tree analysis", *Proc. of the Eight Annual Meeting of The Association for Natural Language Processing (NLP2002)*, Keihanna, Japan, 2002, p. 228-231, (In Japanese).

[NLI 84]   NLI, *Vocabulary, Research and Education*, vol. 13 of *Japanese Language Education Reference*, National Language Institute, 1984, (in Japanese).

[NLI 86]   NLI, *Character and Writing system Education*, vol. 14 of *Japanese Language Education Reference*, National Language Institute, 1986, (in Japanese).

[SAL 90]   SALTON G., BUCKLEY C., "Improving retrieval performance by relevance feedback", *Journal of the American Society for Information Science*, vol. 44, 1990, p. 288–297.

[SUZ 96]   SUZUKAWA K., KATORI F., Eds., *Japanese Proficiency Test Preparation Measure, characters and vocabulary (Level 2)*, Kokusyo Kankoukai, 1996.

[TAK 96]   TAKAHASHI M., SHICHU T., YOSHIMURA K., SHUDO K., "Processing Homonyms in the Kana-to-Kanji Conversion", *Proc. of the 16th International Conference on Computational Linguistics (COLING 1996)*, Copenhagen, Denmark, 1996, p. 1135–1138.

[TER 96]   TERA A., KITAMURA T., OCHIMIZU K., "Dictlinker, a Japanese reading support system", *Proc. of Conference on Japanese Education (Fall)*, Kyoto, Japan, 1996, p. 43-48, (In Japanese).

[TSU 96]   TSUJIMURA N., *An Introduction to Japanese Linguistics*, Blackwell, first edition, 1996.

[VAN 87]   VANCE T. J., *Introduction to Japanese Phonology*, SUNY Press, 1987.