Enhanced Japanese Electronic Dictionary Look-up

Timothy Baldwin*, Slaven Bilac†, Ryo Okumura†,
Takenobu Tokunaga†, Hozumi Tanaka†

* CSLI, Ventura Hall, Stanford University
Stanford, CA 94305-4115 USA
tbaldwin@csli.stanford.edu

†Department of Computer Science
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552 JAPAN
{sbilac,okuryo,take,tanaka}@cl.cs.titech.ac.jp

Abstract

This paper describes the process of data preparation and reading generation for an ongoing project aimed at improving the accessibility of unknown words for learners of foreign languages, focusing initially on Japanese. Rather then requiring absolute knowledge of the readings of words in the foreign language, we allow look-up of dictionary entries by readings which learners can predictably be expected to associate with them. We automatically extract an exhaustive set of phonemic readings for each grapheme segment and learn basic morpho-phonological rules governing compound word formation, associating a probability with each. Then we apply the naive Bayes model to generate a set of readings and give each a likeliness score based on previously extracted evidence and corpus frequencies.

1. Introduction

The dictionary lookup of unknown words presents a major obstacle in learning a foreign language. This is particularly true for non-alphabetic languages such as Japanese where dictionary entries are indexed on the phonemic realization of words, but the phonemic realization is not easily recoverable from the graphemic presentation of that word. We aim to create a robust and efficient dictionary interface that reduces the reading knowledge expectancy placed on learners of the Japanese language.

Modern day Japanese texts consist of the three orthographies of hiragana, katakana and kanji (NLI, 1986). Hiragana and katakana (collectively refered to as "kana") are isomorphic moraic scripts, each character of which bears a relatively straightforward relation to a phonemic form. They are relatively small character sets (46 characters each) and pose no major difficulty to the Japanese learner. The majority of Japanese dictionaries are indexed by *gojuu-on* or the alphabetic ordering of hiragana/katakana.

Kanji characters (ideograms) number up to 3,000, each of which has several different (often unrelated) phonemic realizations that are triggered by different lexical contexts. In addition to the sheer volume of data associated with kanji, the readings of compounds frequently undergo morpho-phonological alternation or take on one-off idiosyncratic readings.

Traditionally, in order to look up a kanji word whose reading is unknown, one would first have to use a kanji character dictionary to look up component characters and then look up the containing word in the index of words containing that kanji character. Kanji lookup is generally based on the "radicals" or main character-subunits making up the character and the total number of strokes needed to write it. Both of these methods are often confusing to the learner and require considerable practice to master.

1.1. Electronic Dictionaries

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 that enable 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 (Breen, 2000). Also, most dictionaries support regular expression-based searches allowing for the lookup of words from partial information, such as one component kanji which the user knows the reading for and can hence input (using a kana-kanji conversion system) into the system interface. Furthermore, several interactive reading aides have become available. Reading Tutor (Kitamura and Kawamura, 2000) performs the text segmentation and then provides translation and semantic information at the word level. The Rikai[1] system, on the other hand, displays the reading and translation of words pointed at with the mouse directly in the browser window. 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).

However, current dictionaries work best when the target text is available in electronic form and needs not be re-entered into the interface, and offer very little user support in the instance that the text is available only in hard copy. Here, current systems require that the user has abso-

[1] http://www.rikai.com

lute knowledge of the full reading of the word in order to achieve direct lookup. In some cases, regular expression-based searches allow the word to be looked up indirectly via a portion of the reading, or by inputing and converting each character of the word separately using a kana–kanji conversion system. While this is acceptable for proficient Japanese language users who posses significant knowledge of kanji characters and can read the word correctly, it is a major handicap for learners of the language.

1.2. Purpose

Learners often possess only limited knowledge of the readings of characters and the phonological and conjugational processes governing word formation. This makes it difficult to identify the correct reading for a string, and the boolean match mechanism adopted in conventional dictionary interfaces discourages the user from attempting to look up a word in the case that they are uncertain of the reading. We believe that if we can imitate the manner in which learners internalize the different readings of characters and the rules governing reading formation, we should be able to decipher which dictionary entry the user was after even when queried with a (predictably) wrong reading.

In this paper we will describe how we go about automatically learning the readings a given kanji segment can take, and the effects of phonological and conjugational alternation on the resultant reading. Once we have a model of the process of reading formation from the individual kanji character readings, we are able to construct a set of plausible readings for each dictionary entry and score them by their likeliness.

The remainder of this paper is structured as follows. Section 2. discusses common misreading errors. Section 3. and Section 4. describe the process of extracting and canonizing the readings of each kanji character, respectively, and Section 5. describes the process of generating and scoring readings.

2. Common Problems

There is a long history of research documenting the problems Japanese learners have in reading texts containing kanji (NLI, 1986; MEIJI, 1997). Commonly-listed problems are:

- *Multiple readings for a given kanji.* In some cases 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 *tai*, *dai* and *oo(kii)* depending on the context, so the string 大会 *taikai* “convention, congress” could feasibly be misread as *ookai* or *daikai*.

- *Insufficient knowledge of readings.* In some cases, learners are only aware of a proper subset of the readings a given kanji can take, and are thus forced into making wrong reading predictions when faced with new words drawing on a novel reading for that kanji. In the previous example, a user aware only of the *oo(ki)* reading for 大 would almost certainly try to read 大会 as *ookai*. 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”).

- *Incorrect application of phonological and conjugational rules governing reading formation.* For example, 発 *hatsu* and 表 *hyou* form the compound 発表 *happyou* “announcement”,[2] but readings such as *hatsuhyou* or *hahhyou* could equally arise from the component character readings.

- *Confusion due to graphic similarity of different kanji.* Learners who have had limited contact with kanji can easily confuse characters. For example, 基 *ki* “foundation” and 墓 *bo* “grave” are visually similar, resulting in the transfer of the reading of one kanji onto the other.

- *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 substituted for each other, resulting in an erroneous reading.

- *Confusion as to length of vowels or consonants.* For example, 主催 *syusai* “organization, sponsorship” can be mistakenly read as *syuusai*, or 最も *mottomo* “most, extremely” as *motomo*.

- *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 handle this type of error.

3. Extraction of segment readings

To be able to generate plausible readings for a given kanji string, we would like to know all the readings a given kanji can take. While kanji dictionaries list the most common readings each character can take, they do not give any information about the phonological and conjugational effects of compound formation. In order to get this data we take a set of kanji–reading string pairs and automatically align atomic segments of the kanji string, with their corresponding readings in the reading string. Note that “atomic segments” cannot be further segmented up into smaller parts which correspond meaningfully to partitions of the reading string, and can potentially extend over multiple kanji (see below). The particular dictionary used here and throughout the research is the publically-available EDICT dictionary (EDICT, 2000).

3.1. Grapheme–phoneme alignment

Alignment is achieved by way of grapheme–phoneme alignment between kanji (grapheme) strings and their readings in the form of hiragana (phoneme) strings (Divay and Vitale, 1997; Huang et al., 1994).[3] In this, we attempt to extract the complete set of phoneme realizations (component readings) for each grapheme segment (kanji segment).

[2] Here, *hatsu* undergoes gemination and *hyou* sequential voicing to produce *happyou*.

[3] Noting that hiragana characters are not strictly phonemes, but phoneme clusters such as か *ka* and ぶ *bu*.

Our method requires no supervision and could be applied to other languages in which the phonetic realization is not clearly derivable from the grapheme presentation (Baldwin and Tanaka, 1999).

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 and that the alignment is strictly linear.
2. Prune candidate alignments through the application of linguistic constraints. These constraints are the only component of the alignment process which is specific to the Japanese language, and include requiring segment boundaries at script boundaries (except for kanji-hiragana boundaries), and the preference that each reading segment contains only one voiced obstruent (Lyman's Law — Vance (1987)).
3. Score each alignment by a variant of the TF-IDF model (Salton and Buckley, 1990). The modification from the basic TF-IDF model allows for better handling of affixes and verbal/adjectival conjugation so as to not over-penalize commonly occurring grapheme–phoneme pairs.
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).

For full details, see (Baldwin and Tanaka, 1999; Baldwin and Tanaka, 2000).

Examples of resulting alignments are:

⟨ 発表 ⟩–⟨*happyou*⟩ ⇒ ⟨ 発|表 ⟩–⟨*hap*|*pyou*⟩

⟨ 風邪薬 ⟩–⟨*kazegusuri*⟩ ⇒ ⟨ 風邪|薬 ⟩–⟨*kaze*|*gusuri*⟩

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.

4. Reading Canonization

Based on the alignment data, we can read off a set of readings for each kanji segment. Such readings are subject to both phonological and conjugational alternation, however, such that the phonological variants of *hyou* and *byou* could be produced for 表 "chart", and the conjugational variants of *yomi* and *yomu* could be produced for the verb 読 "read".

In particular we focus on sequential voicing ("rendaku") and sound euphony ("onbin"), which commonly occur in word formation (Tsujimura, 1996; Vance, 1987). 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. Examples of sequential voicing are:

本 *hon* "book"+ 棚 *tana* "shelf" ⇒
本棚 *hondana* "bookshelf"
旅 *tabi* "travel"+ 人 *hito* "person" $Rightarrow$
旅人 *tabibito* "traveller"

Note that sequential voicing produces two voicing possibilities for the consonant /h/: full voicing (/b/) and semi-voicing (/p/). Assuming that we know that sequential voicing has taken place, however, it is generally possible to uniquely recreate the base form of the reading.[4].

"Onbin", or sound euphony, similarly occurs in binary word formation, and 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.[5] It has several different subforms limited to verbal and adjectival conjugational form including "*i* onbin" or velar vocalization and "*hatsu* onbin" or nasalization[6]. However the most common form, assimilatory gemination or "*soku* onbin", is a morphological process which occurs in word formation. The occurrence of sound euphony depends on voicing and the manner of articulation of the following segment. Examples of sound euphony are:

国 *koku* "country" + 境 *kyou* "boundary"⇒
国境 *kokkyou* "(national) border"
脱 *datsu* "remove" + 出 *shutsu* "leave,exit"⇒
脱出 *dasshutsu* "escape"
言う *iu* "say" + *te*[7] ⇒ 言って *itte* "say(ing)"

For simplicity, we will refer to the various forms of conjugation-related sound euphony (e.g. the third example above) as "conjugation", and the morphological process of assimilatory gemination (e.g. the first and second examples above) as "gemination" for the remainder of this paper. Note that conjugating endings are included within the conjugating segment along with the verb stem (i.e. 言って *itte* "say(ing)" above is considered to be a single segment), and that there are non-geminating forms of conjugation (e.g. conjunctive conjugation: 言う *iu* "say" ⇒ 言い *ii*). Also, conjugational segments can further undergo gemination (引き *hiki* "pull(ing)" + 越し *koshi* "go(ing) beyond" ⇒ 引っ越し *hikkoshi* "moving (house)").

Unlike sequential voicing, the simple knowledge that sound euphony has taken place is generally not sufficient to uniquely recreate the base form of the geminated consonant, even when the type of the proceeding consonant (which the geminated consonant is in harmony with) is taken into account. For example, in the first example above, the base form of 国 *kok* "country" given the right context

[4]Exceptions to this generalization are /p/ (possible base forms: /h/ and /b/), /zu/ (possible base forms: /tsu/ and /su/) and /zi/ (possible base forms: /tʃi/ and /ʃi/)

[5]Note that sound euphony occurs only when the base reading is made up of at least two morae, whereas sequential voicing occurs for readings of all lengths.

[6]Conjugational endings of verbs and adjectives are always written in hiragana and as such do not cause reading problems. We handle them in the alignment and canonization steps, but do not generate any readings based on these phenomena in the generation step

[7]Conective verbal conjugational ending

of /k/ could be, e.g., *koki*, *koku*, *kotsu*, each of which has equivalent phonological plausibility.

4.1. Canonization

The alignment data contains all possible readings for a given grapheme segment, within the context of the data at hand. These readings include alternants due to sequential voicing, sound euphony and conjugation, and possibly (but not necessarily) the base form of each reading. We would like to canonize the readings to separate the base reading data apart from the alternation probabilities, thereby minimizing the number of reading types and maximally extracting instances of alternation. This provides a means of overcoming data sparseness, and at the same time allows us to produce unobserved segment-level readings through novel alternation combinations over the base readings (hence increasing the coverage of predicted readings that a Japanese learner may come up with).

Above, we observed 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. Additionally, sequential voicing affects only the initial mora of the segment reading, and sound euphony only the final mora of the reading, and in the case that the reading is made up of a single mora (kana character), only sequential voicing can occur. To detect the two phenomena, therefore, we can classify segments according to the presence of left and right lexical context, and compare readings occurring in different contexts to determine whether an analysis exists whereby multiple reading alternants can be explained by way of a single base reading (Okumura, 2001).

Based on the presence of left and right lexical context, we classify segment readings into 4 groups:

- Level 0 ($-$left, $-$right context): no possibility of conjugation or phonological alternation[8].
- Level 1 ($-$left, $+$right context): possibility of gemination or conjugation
- Level 2 ($+$left, $-$right context): possibility of sequential voicing
- 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 (including the possibility of zero-derivation, whereby no phonetic alternation has taken place). We thus work through the various levels in decreasing numeric order, and determine whether a unique 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, increment the frequency of occurrence of that alternation by the frequency of the string in which alternation was found to occur, and combine the frequency of the derived reading with that of the base reading.

[8]Since we are dealing with dictionary entries in our alignment process, conjugating (verbal and adjectival) level 0 segments can be assumed to be in "base form".

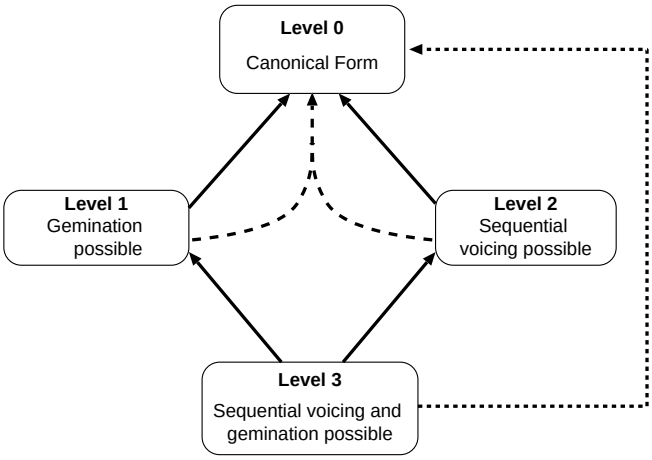

Figure 1: Canonization flowchart

The canonization process is depicted in figure 1.

For each level we employ a slightly different procedure. First, we perform conjugational analysis (Baldwin, 1998) at Levels 1 to 3 to establish whether it is possible to analyze each segment as having 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.

In the case that 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. This distribution of frequency extends to any phonological alternation or conjugation associated with each match.

Level 3 entries are treated in two passes. First we try to merge Level 3 entries with those at Levels 1 and 2, respectively, based on boolean match over the original reading, and failing this, analysis of gemination and sequential voicing, respectively. In the case of gemination, we make no assumptions about the possible range of base forms of the segment-final mora, and allow matches to any reading for the given kanji segment, which differ over the Level 3 reading only in the final mora. The analysis of sequential voicing is rather more constrained, in that the maximum number of possible base forms for a voiced initial mora is two (see above). All readings for the given kanji segment are thus searched over, and a match returned if the reading string consists of a string-initial devoiced variant of the Level 3 reading. In the case of multiple matches at Levels 1 and 2, the original frequency of the kanji–reading pair is distributed equally between all matching strings. Note that, despite Level 3 kanji–reading pairs being sandwiched between two segments, it is perfectly possible that no alternation has taken place, or that only one of gemination and sequential voicing has occurred.

If no merge with a Level 1 or 2 entry was possible, we proceed to carry out combined analysis of sequential voicing and gemination against Level 0 entries. If a match is found, the frequency of the original kanji–reading pair is

distributed between all matching entries. If no match is found, we directly create a new Level 0 entry and carry over the frequency from the original entry.

For Level 2, we first look for an identical entry at Level 0 and merge the two if possible. Failing this, if the reading contains a segment-initial voiced consonant, we replace the consonant in question with the underlying form(s), and look for a match at Level 0. If a match is found, we merge to Level 0. In the case that no sequential voicing-based analysis is immediately apparent for the given reading–kanji pair, we look for a canonical form in the Level 1 data, allowing for the possibility of the segment-final mora having been geminated in the Level 1 string at the same time as the segment-initial mora in the Level 2 string having been voiced. Assuming that a match is achieved, the two readings are merged together at Level 0, using the canonical reading and combining the respective frequencies. In the instance that no match is possible at any level, the kanji–reading segment pair is promoted to Level 0 as is.

Turning finally to Level 1, we first look to merge to an identical entry at Level 0, and failing this, carry out a gemination-based analysis of the original reading, and search for canonical forms at Level 0. In the instance that no match is possible, the kanji–reading segment pair is promoted to Level 0 as is.

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 enable us to calculate probabilities according to Equation 1:

$$P_\alpha(r) = \frac{\text{Number of observed } \alpha \text{ alternations}}{\text{Number entries satisfying the conditions on } \alpha} \quad (1)$$

where $\alpha \in \{\text{sequential\_voicing}, \text{gemination}, \text{conjugation}\}$. For each segment, we tease apart the frequencies for sequential voicing, gemination and conjugation so as to be able to reapply them as independent probabilities below.

Both sequential voicing and gemination have received significant attention in the literature and several rules governing/predicting their occurrence have been proposed. However, as we are attempting to model the knowledge of a Japanese learner, we want to assume as little linguistic knowledge as possible. Prediction of the two effects is thus based on only the immediate lexical context of the mora in question, that is the mora potentially undergoing alternation and the neighboring mora in the adjacent segment. Given a mora m_i and its single mora lexical context m_{ctxt}, therefore, we generate probabilities for m_i undergoing either sequential voicing (if m_i is segment-initial and there exists a left lexical context $m_{i-1} = m_{ctxt}$) or gemination (if m_i is segment-final, the segment is at least 2 morae in length and there exists a right lexical context $m_{i+1} = m_{ctxt}$). In the case of sequential voicing, if m_i contains the consonant /h/ or /f/,[9] we make a three-way distinction between no phonological alternation, and /h/ being fully or semi-voiced.

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

$\langle$ 発|表 $\rangle$–$\langle$*hap*|*pyou*$\rangle$ $\Rightarrow$ $\langle$*hatsu*|*hyou*$\rangle$
+gemination +voicing
$\langle$ 風邪|薬 $\rangle$–$\langle$*kaze*|*gusuri*$\rangle$ $\Rightarrow$ $\langle$*kaze*|*kusuri*$\rangle$
+voicing

Once we have the canonized data, it is trivial to 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.

4.2. Bigram Segmentation

In canonizing the kanji–reading data, we derived probabilities for a given kanji segment taking different readings, and also for different types of reading alternation to occur. In order to generate probabilities for different readings for a given kanji string, however, we must know how to partition it up into kanji segments in order to be able to apply the probabilities for component readings for each. This is achieved through the calculation of bigram probabilities, rating the likelihood of the given bigram being split into two segments, or chunked together into a single segment. Note that this differs from grapheme–phoneme alignment in that we do not consider the reading of the string at all, but are after a probabilistic model of how a user might partition a given string into segments in order to generate a reading for the overall kanji string.

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. For each bigram we count the probability of it being segmented as one or two units.

The grapheme–alignment data provides an explicit description of segmentation information, which we can read off directly to feed into the reading generation module.

5. Reading Generation

Above, we derived probabilities for different readings for a given kanji segment ($P(r|k)$), and for a given reading undergoing sequential voicing ($P_{voice}(\textit{head}(r))$), gemination ($P_{gem}(\textit{tail}(r))$) and conjugational ($P_{conj}(r)$) alternation.[10] The probability of each segment taking a given reading depends on the characters contained in the kanji segment whereas the probability of phonological and conjugational alternation depends only on the reading.

From the above data, we generate an exhaustive listing of reading candidates for each dictionary entry s consisting of n segments and calculate the overall probability of each reading in line with the naive Bayes model, as described in Equations 2 and 3. That is, we assume that the segmentation, reading, conjugational alternation and phonological alternation probabilities are independent of one another, and multiply together the component probabilities for each. In cases where several possible segmentations exist, we run the generation process for each such segmentation candidate.

$$P(r|s) = P(r_{1..n}|k_{1..n}) \quad (2)$$

[9] I.e. $m_i \in \{$*ha,hi,fu,he,ho*$\}$.

[10] Here, the *head*(r) and *tail*(r) operators return the first and last morae respectively of the reading string r.

$$P(r_{1..n}|k_{1..n}) = \prod_{i=1}^{n} P(r_i|k_i) \times P_{voice}(\mathit{head}(r_i)) \times P_{gem}(\mathit{tail}(r_i)) \times P_{conj}(r_i) \quad (3)$$

After obtaining the probability of the $P(r|s)$ we apply Bayes' rule (equation 4) to obtain the value we are interested in: the probability of string s given reading r, that is $P(s|r)$.

$$P(s|r) = \frac{P(r|s) \times P(s)}{P(r)} \quad (4)$$

The probability $P(s)$ can be calculated from the test corpus according to Equation 5. We used the complete EDR Japanese corpus as the training set (EDR, 1995).

We use the $P(s|r)$ values to present all dictionary entries s mapped onto from r in decreasing order, thus outputing the more likely dictionary entries first. Notice that the term $\sum_i F(s_i)$ in Equation 5 is constant for a given corpus and can be factored out of the final equation while maintaining the score-wise ranking of dictionary entries. Furthermore $P(r)$ is constant for a given r input and similarly does not affect the relative ranking of dictionary entries. We thus estimate the likeliness of a dictionary entry s given a reading r as given in Equation 6.

$$P(s) = \frac{F(s)}{\sum_i F(s_i)} \quad (5)$$

$$Grade(s|r) = P(r|s) \times F(s) \quad (6)$$

At the end of this process we have a set of generated readings for each dictionary entry and each of the readings has a likeliness score associated with it. For a given static dictionary, it is possible to pre-compute all possible dictionary entries reachable from a given (reading) input, and determine a score for each. When the user then queries the system, all that is required is that we do a boolean search over the generated readings, and in the case of a match, return all corresponding dictionary entries in descending numerical order of the likeliness score (as determined by Equation 6).

Note that the training data which feeds the generation process is the very same dataset as that for which readings are generated. That is, the training and test data are one in the same. This has the advantage that we are guaranteed to reach the correct reading for every dictionary entry, given that that dictionary entry forms part of the training data used in segmenting the string and compositionally generating a reading thereform. There is no guarantee, however, that the correct reading will assume the highest score, as the probabilities associated with alternative readings could plausibly be higher than those for the correct reading, and it will tend to occur that more salient incorrect readings for common words will rank higher than the correct readings for uncommon words.

One important quality of all steps of processing described above is that they are fully automated. This has benefits in terms of developing customized interfaces to different dictionaries (e.g. domain-specific lexicons) with no manual input, and also in updating the system each time the dictionary data is altered.

	Types	*Tokens*
Level 0 (initial)	5,622	5,622
Level 1	14,430	51,551
Level 2	7,867	51,334
Level 3	3,273	21,249
Overall	15,100	129,756
Level 0 (final)	7,092	129,756

Table 1: Number of kanji–reading tokens and types pre- and post-canonization

The overall dictionary interface has been implemented in a web-based environment (Bilac et al., 2002), and is available for public use at `http://hinoki.ryu.titech.ac.jp/dicti/`.

6. Evaluation

As stated above, the system currently uses the EDICT Japanese–English dictionary, which consists of 97,399 entries in total, 82,961 of which contain kanji and are used to generate readings.

To evaluate the performance of the alignment method, firstly, we aligned all 82,961 kanji-containing entries, and manually checked the alignment analyses of a random sample of 5,000 entries. For these, we rated alignment performance according to word accuracy (the proportion of words for which a fully correct alignment analysis was produced) and also segment precision and recall; segment precision describes the proportion of segments in the alignment output which were correctly aligned, whereas segment recall describes the proportion of segments in the manually-annotated data that were correctly realised in the alignment output. The results according to these three metrics were:

Word accuracy	*Segment precision*	*Segment recall*
97.22%	98.11%	98.67%

Next, we analyzed the efficacy of the reading canonization process according to the number of reading types at each level initially, and the number of reading types remaining at level 0 at the end of processing, the results of which are presented in Table 1. Here, we present the number of kanji–reading types and tokens at each level initially and in the final state, at the completion of processing (noting that all entries end up at Level 0, irrespective of whether a match at Level 0 was found in the original data). The success of the canonization process can be gauged from the reduction in the number of kanji–reading segment types from 15,100 initially, to 7,092 finally, a reduction of over 50%. This is due to the detection of instances of both conjugation and phonological alternation.

Finally, we provide a statistical breakdown of the reading generation process:

Total dictionary entries: 97,399
Total dictionary entries w/kanji: 82,961
Total generated readings (tokens): 2,646,137
Total generated readings (types): 2,194,159
Average readings per entry: 27.24
Average entries per reading: 1.21
Maximum readings per entry: 471
Maximum entries per reading: 112

For the 82,961 dictionary entries containing kanji, an average of 27.24 readings was generated for each entry. Fortunately, the level of overlap between readings is not high, such that the average number of dictionary entries per generated reading is a modest 1.21. The user is thus not generally overwhelmed with vast numbers of outputs, a distinct advantage when looking up a word using the correct reading.

One thing that is not evident from the above results is just how effective the proposed method is at directing the user to the correct dictionary entry. This presents an area for future research: carrying out user evaluation to determine (a) if useful errant readings are generated, (b) if the ranking of dictionary entries is reflective of the relative salience of the associated dictionary entries, and (c) patterns of error in user inputs.

Additionally, as mentioned in Section 2., semantic and graphic similarity can also lead to user errors, neither of which phenomenon we model at present. We envisage calculating separate probabilities for readings attributable to these different effects, interpolating over them to produce a consolidated probability for each reading given a kanji, and then weighting for the effects of conjugation, gemination and sequential voicing as per above.

7. Conclusion

In this paper we have proposed a method for constructing a system capable of handling motivated reading errors, to facilitate more efficient dictionary lookup for Japanese learners. Rather then requiring absolute knowledge of the readings of words in the foreign language, our method allows look-up of dictionary entries by way of readings which learners can predictably be expected to associate with them. We have exemplified the component processes of alignment, reading canonization and reading generation, which combine to produce a probability for the different readings which can be productively generated for a given kanji string. From this, we can then arrive at a ranked list of dictionary entries which the user can realistically be expected to be seeking in inputting a (potentially wrong) reading.

Acknowledgements

This research was supported in part by the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University. We would particularly like to thank Prof. Nishina Kikuko of the International Student Center (TITech) for hosting the web-accessible version of the system, and Francis Bond and Christoph Neumann for providing valuable feedback at various points during this research.

8. References

Timothy Baldwin and Hozumi Tanaka. 1999. The applications of unsupervised learning to Japanese grapheme-phoneme alignment. In *Proc. of ACL Workshop on Unsupervised Learning in Natural Language Processing*, pages 9–16, University of Maryland.

Timothy Baldwin and Hozumi Tanaka. 2000. A comparative study of unsupervised grapheme-phoneme alignment methods. In *Proc. of the 22nd Annual Meeting of the Cognitive Science Society (CogSci 2000)*, pages 597–602, Philadelphia.

Timothy Baldwin. 1998. The analysys of Japanese relative clauses. Masters's thesis, Tokyo Institute of Technology.

Slaven Bilac, Timothy Baldwin, and Hozumi Tanaka. 2002. Construction of a Japanese learner-friendly dictionary interface. In *Proc. of the Eight Annual Meeting of The Association for Natural Language Processing (NLP2002)*, pages 460–463.

Jim W. Breen. 2000. A WWW Japanese Dictionary. *Japanese Studies*, 20:313–317.

Michael Divay and Anthony J. Vitale. 1997. Algorithms for grapheme-phoneme translation for English and French: Applications for database searches and speech synthesis. *Computational Linguistics*, 23:495–523.

EDICT. 2000. EDICT Japanese-English Dictionary File. ftp://ftp.cc.monash.edu.au/pub/nihongo/.

EDR. 1995. *EDR Electronic Dictionary Technical Guide*. Japan Electronic Dictionary Research Institute, Ltd. In Japanese.

Caroline B. Huang, Mark A. Son-Bell, and David M. Baggett. 1994. Generation of pronunciationfs from orthographies using transformation-based error-driven learning. In *In Proc. of the International Conference on Speech and Language Processing*, pages 411–414.

Tatuya Kitamura and Yoshiko Kawamura. 2000. Improving the dictionary display in a reading support system. International Symposium of Japanese Language Education. (In Japanese).

Meiji Publishing Planning/Editing Group MEIJI. 1997. *Analysis of misuse of Japanese Language*. Meiji Publishing. (In Japanese).

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

Ryo Okumura. 2001. Basic research on an intelligent dictionary interface for learners of the Japanese language. Bachelor's thesis, Tokyo Institute of Technology. (In Japanese).

Gerald Salton and Chris Buckley. 1990. Improving retrievel performance by relevance feedback. *Journal of the American Society for Information Science*, 44:288–297.

Natsuko Tsujimura. 1996. *An Introduction to Japanese Linguistics*. Blackwell, Cambridge, Massachusetts, first edition.

Timothy J. Vance. 1987. *Introduction to Japanese Phonology*. SUNY Press, New York.