Orthographic similarity search for dictionary lookup of Japanese words

Lars Yencken and **Timothy Baldwin**
{lljy,tim}@csse.unimelb.edu.au
NICTA Research Lab
University of Melbourne

Abstract.
Finding an unknown Japanese word in a dictionary is a difficult and slow task when one or more of the word's characters is unknown. For advanced learners, unknown characters evoke the form and meaning of visually similar characters they are familiar with. We propose a range of character distance metrics to allow learners to leverage known characters to search for words containing unknown but visually similar characters. This new form of dictionary search is implemented as an extension to the FOKS dictionary system.

1 Introduction

Japanese kanji are logographic characters, akin to traditional Chinese hanzi in number and visual complexity, but with a more loosely coupled system for pronunciation in which kanji readings are both more numerous and context-specific. Learners spend much time laboriously looking up unknown characters, since there is no common method for typing them directly into a computer based on their shape alone. Any time saved from this onerous task is likely to speed up learning and aid motivation.

Since word lookup is by characters or by pronunciation, words containing unknown characters are typically found by looking up those characters one at a time in a character dictionary. The traditional method of kanji character lookup involves identifying the primary component (or "radical"), counting its strokes, looking it up in the index, counting the remainder of strokes in the original character, then finding the character in a sub-index. This presents several opportunities for error, but fortunately improvements have been made, as we discuss in Section 2.

We propose a method for looking up unknown words containing words with unfamiliar characters, based on similarity with known characters. In essence, our method is based on the user plausibly "mistyping" the word based on closely-matching kanji they are familiar with (and hence can readily access via a standard input method editor), from which we predict the correct kanji combination based on kanji similarity and word frequency. For example, given the input 補左, the system could suggest the word 補佐 [hosa] "help" based on similarity between the high-frequency 左 and the graphically-similar but low-frequency 佐.

The proposed method is combined with the FOKS lookup strategy proposed by (2) for looking up unknown words via plausibly incorrect *readings*.

The contributions of this paper are the proposal of a range of character similarity models for logographic scripts, a novel evaluation method for logographic character confusability, and the incorporation of kanji similarity into a word-level lookup model.

The remainder of this paper is structured as follows. Firstly, we review existing kanji lookup methods (Section 2), and go on to discuss how we measure and model kanji similarity, including an evaluation of the methods (Section 3). We then focus on the conversion of similarity models into confusion models, and their integration into a search interface (Section 4). Examining both our models and the interface itself, we discuss our findings (Section 5) before finally concluding (Section 6).

2 A review of kanji lookup methods

In this section, we provide a brief review of existing kanji lookup methods.

The SKIP (System of Kanji Indexing by Patterns) system of lookup removes the need to identify the primary radical, providing a more efficient indexing scheme which relies on overall shape instead (5). For example, 明 [aka] "bright" has skip code 1-4-4. The first number indicates it is split into two parts horizontally, and the second and third numbers are the stroke counts of those two parts respectively. Since the entire code can be determined at once, it is faster than traditional methods, though learners are prone to making stroke count errors.

The Kansuke dictionary forgoes radicals or shape and instead simplifies the method of counting strokes, counting the number of horizontal, vertical and other strokes, to form a three-number code which learners find easier to determine than traditional stroke counting methods (6). Characters can also be looked up from their components. For our earlier example, 明 consists of 日 with code 3-2-0 and 月 with code 3-1-1. These two components sufficiently constrain results to find our target.

In contrast, the commercial Kanjiru dictionary (8) focuses on innovative use of MacKay's Dasher interface, attempting to interactively assemble a character by shape and stroke in an adaptive manner. The user guides the search through mouse movements, providing the user with a way of building up components stroke by stroke until the desired character is found.

Finally, hand writing interfaces such as that used for WWWJDIC[1] attempt to circumvent the computer input problem altogether. However, the accuracy of the current generation of such interfaces is limited. They suffer from the awkwardness of drawing characters with a mouse, and also oversensitivity to stroke order and connectivity of

[1] http://www.csse.monash.edu.au/~jwb/hwr/

components in resolving the resulting drawing. Furthermore, learners tend to imitate kanji fonts when handwriting instead of using the more natural hand-written style of native speakers, resulting in subtly different character shapes.

Each of these start-of-the-art lookup methods contrasts with our proposed similarity-based search in several ways. Firstly, our search is unconventional in that it is word-based, yet provides the means to look up words with unknown characters without the use of wildcards. However, the need to use kanji in the search query limits its usefulness if a core set of common kanji is not known, limiting our scope to intermediate and advanced learners.

Secondly, as an extension to an existing search method, we are able to cater to both intentional similarity-based searches, and unintentional input errors made by users, increasing accessibility of the base dictionary. In this way, we share much with the FOKS dictionary interface (2), which provides error-correcting lookup for reading-based dictionary queries. We demonstrate the complementarity of the two methods by proposing a combined lookup model based on either readings or kanji.

The FOKS methodology can be illustrated by way of an example. Suppose a user wishes to look up the word 山車, but is unsure of its pronunciation. FOKS allows them to guess the pronunciation based on readings they know for each character in other contexts. In this case, they might combine 山 [yama] "mountain" and 車 [kuruma] "car" together and guess the combined reading as [yamakuruma]. In this case, the correct reading [dashi] is non-compositional, and hence cannot be guessed from the word's parts. Nonetheless, the search is constraining enough to allow users to find the word, and discover its meaning ("festival float").

On the other hand, suppose the user enters kanji as their search query, but makes an error. For example, they may be looking for the word 方言, but accidentally enter 万言. Our proposed method would take into account the similarity between 万 and 方, and provide the desired word in the results, allowing the user to determine both its pronunciation ([hōgeN]) and its meaning ("dialect").

3 Modelling similarity of kanji

3.1 Metric space models

There has been little work on methods for measuring or predicting the similarity between two kanji. While there have been many psycholinguistic studies on various specific aspects of perception of Chinese and Japanese logographic characters, few touch on direct aspects of orthographic confusion. For a brief discussion, see (10).

As a broad overview, current literature suggests that some form of hierarchical recognition occurs, building radicals from strokes, and whole characters from radicals. Each point of recognition and combination suggests a potential site for misrecognition or confusion with an orthographic or semantic neighbour; ideally a distance metric addresses them all. The study by Yeh and Li (9) found that, in a visual search task, characters with shared radicals were only distractors if the broad shape of the two characters were the same.[2] This indicates the importance of shared shape.

We previously considered two different similarity measures on kanji: a cosine similarity metric operating on boolean radical vectors, and the l_1 norm (Manhattan distance) between rendered images of kanji (10). Evaluating on a set of human similarity judgements, we determined that the cosine similarity method outperformed the l_1 norm, although it had lower precision for high-similarity pairs. Agreement on the experimental set was still quite low, so we attempted to improve these models in order to reduce any noise that occurring in similarity-based search results.

3.1.1 Bag of radicals with shape

As the intermediate level of structure between strokes and whole-characters, radicals have much perceptual salience. When learners of Japanese study a new character, they do not study its strokes, but instead its component radicals. For example, 明 [aka] "bright" could be analysed as being made up of the 日 [sun] "hi" and 月 [moon] "tsuki" radicals.

Radicals are useful in several ways. The number of radicals in any kanji is much smaller than the total number of strokes, making kanji easier to chunk and recall. Furthermore, they can provide semantic cues as to the whole-character meaning, and can also provide pronunciation cues.

The original metric used in (10) simply calculated the cosine similarity between radical vectors. This ignores the position of radicals, which is known to be important in similarity judgements, and also the number of each radical within a kanji. Hence, 木, 林 and 森 are all considered identical (radical = 木), as are 日 and 晶 (radical = 日).

To address this weakness and the findings of Yeh and Li's study, we augmented the original metric to always consider two characters different unless their basic shape is the same. This allows it to distinguish between the previous examples. The final form is:

$$d_{\text{radical}}(x,y) = \begin{cases} 1 - \frac{r_x \cdot r_y}{|r_x||r_y|} & \text{if shape}(x) = \text{shape}(y) \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

In this case, we approximate the broad shape by the use of the first part of each kanji's 3-part SKIP code, which can take values *horizontal*, *vertical*, *containment* or *other*. SKIP codes for each kanji are provided in kanjidic and radical membership in radkfile. This metric aims to capture the visual and semantic salience of radicals in kanji perception, and to also take into account basic shape similarity.

3.1.2 Distance of rendered images

In stark contrast to the highly language-specific radical component metric, we can instead consider a much more abstract approach to visual similarity of kanji by directly evaluating the similarity of how they are displayed on screen or in print. The simplest way to do this is to simply render two kanji to an image of fixed size, and to perform some comparison on that image. This is precisely the goal of the l_1 norm metric.

There are many possible methods of establishing a distance between images, but the simplest again is to consider the difference between pixels of the two images for some alignment. Fortunately, all kanji are intended to occupy an identically sized block, so we need not worry about scaling or otherwise transforming images before comparison. Considering $p_x(i,j)$ to be the image brightness for the pixel at position (i,j), we evaluate the l_1 norm as follows:

$$l_1(x,y) = \sum_{i,j} |p_x(i,j) - p_y(i,j)| \quad (2)$$

Note that the exact calculation will be different depending on the font and image settings. We used the MS Gothic font, rendering to 80x80 images, with anti-aliasing.

[2] In the experiment, each kanji was chosen so that it consisted of two parts, arranged horizontally or vertically. These are the most common arrangements of components, though many more fine-grained subcategories can also be determined.

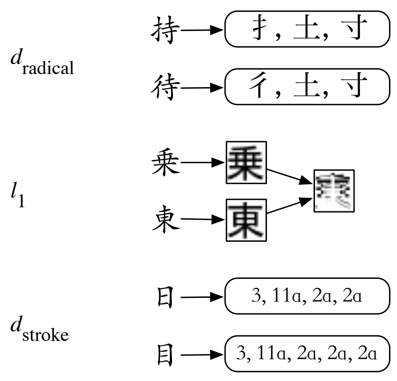

Figure 1. A visual overview of the three distance metrics. For d_{radical}, a token representing each radical present is used. For l_1, the images are rendered and the sum of the difference image is calculated. For d_{stroke}, the edit distance between stroke code vectors is used.

This metric is aimed at capturing the general overlap of strokes between the two characters, and also the overlap of whitespace, which gives useful structure information. This metric is known to be noisy for low-to-medium similarity pairs, but is very useful at distinguishing near neighbours.

3.1.3 Stroke edit distance

Of the metrics we have introduced so far, we firstly have d_{radical} which rewards shared radicals and shape, but discards radical position information and ignores any potential similarity between radicals themselves. For example, despite differing in only one stroke, 日 [hi] "sun" is considered distant to 目 [me] "eye". We also have the l_1 norm which rewards high levels of stroke overlap, but ignores the salience of radicals altogether, and is known to be noisy. If we can find a method which makes better use of radical position information but rewards sufficiently similar radicals which might be confused, we can expect such a metric to perform much better. This requires a data set which contains hierarchical information about kanji.

Such a data set was created by Apel and Quint (1), who envisioned its use in dictionary lookup applications. Each kanji is specified by its strokes, grouped into common stroke groups (components), and broken down in a hierarchical manner into relative positions within the kanji (for example: left and right, top and bottom). The strokes themselves are based on a pre-determined taxonomy of some 26 stroke types (46 including sub-variants).

There are many ways to construct distance metrics between such hierarchical representations of kanji, however the only method we have of evaluating similarity between radicals directly is the sequence of strokes within them. Between two such ordered sequences, the edit distance is a natural metric to choose. However, the components themselves have a natural ordering in which they are written, from left to right and top to bottom. We can then use this ordering to generate a sequence of strokes for each entire kanji, and compare whole kanji using the edit distance metric on these sequences. This forms our d_{stroke} metric.

String edit distance satisfies the triangle inequality (7), and has several desirable properties in this context. When written by hand, each radical in a kanji is written in sequence, completing one before the next. This means radicals form natural contiguous subsequences of these stroke descriptions, each with a reasonably unique signature. The dynamic programming algorithm thus aligns matching radicals whenever they are close enough in two descriptions. The order of components in a description also reflects their position as part of the larger compound: components are usually drawn in a left-to-right, top-to-bottom order. Finally, it provides a smooth blending from stroke similarity to radical similarity, and can recognize the similarity between cases like 日 and 目, as shown in Figure 1.

3.2 Evaluation

We evaluate our distance metrics over two data sets.

The first dataset is the human similarity judgements from our earlier experiment (10). This data set is overly broad, in that it weights equally the ability to distinguish low and medium similarity pairs with distinguishing medium and high similarity pairs. It is clear that for most applications, determining the high similarity pairs with high precision is what is most important. Nevertheless, this data set is useful for comparing our metrics with those in the earlier study.

In order to better measure performance on high-similarity pairs, which we expect to form the basis of incorrect kanji inputs, we need a set of human-selected confusion data. The second data set is based on the White Rabbit JLPT Level 3[3] kanji flashcards. Each flashcard contains either one or two highly-similar neighbours which might be confused with a given kanji. We use this set to determine our likely performance in a search task.

3.2.1 Similarity experiment data

The first data set was generated by exposing participants of various levels of Japanese proficiency to 100 random stimulus pairs, each consisting of two kanji, and asking them to rank their graphical similarity on a 5 point scale. The aim of the experiment was to collect data to directly evaluate similarity metrics such as those that we have proposed.

Since agreement between raters was low, rather than distill a gold standard similarity value for each pairing, we evaluate by comparing each metric's predictions with the human responses. Since we are interested in the ordering between kanji rather than exact similarity value, we use Spearman's rank correlation coefficient ρ to measure agreement between a metric and an individual rater, then take the mean across all raters in the group.

The key participant groups of interest are shown, in particular: (1) non-speakers of Chinese, Japanese or Korean (Non-CJK); (2) Japanese second-language learners (JSL); and (3) Japanese first-language speakers (JFL). Evaluating over these groups gives the results in Table 1.

Metric	Non-CJK	JSL	JFL
d_{radical} (+shape)	0.561	0.673	0.700
d_{radical} (−shape)	0.463	0.575	0.673
l_1	0.305	0.408	0.418
d_{stroke}	0.128	0.318	0.431

Table 1. Mean value of Spearman's rank correlation ρ over rater groups for each metric. For the d_{radical} variants, −shape indicates the original metric and +shape indicates our augmented version.

For each of the similarity metrics, the mean rank correlation increased as the participants' knowledge of Japanese increased (from

[3] Japanese Language Proficiency Test: the standard government test for foreigners learning Japanese, with four levels ranging from 4 (easiest) to 1 (most difficult).

Non-CJK to JSL to JFL), allowing them to make more motivated and consistent similarity judgements. The d_{radical} (+shape) metric clearly dominates over the other metrics at all levels of knowledge, including the original d_{radical} (−shape), which we included for comparison. This confirms the salience of radicals and the tendency for individuals to classify kanji by their broad shape, as suggested by Yeh and Li's study (9).

The l_1 norm performs relatively poorly, even for the Japanese native speakers. Our new metric d_{stroke} performs very poorly for the non-speakers, yet improves to around the same level as l_1 for native speakers. This suggests it is especially noisy in its broad ranking of pairs for similarity, but gives no definitive information on its high-similarity behaviour. We thus continue our evaluation on the flashcard data set for comparison.

3.2.2 Flashcard data set

Consisting of only high-similarity pairs, the flashcard data set is quite different from the similarity experiment set, warranting different methods of evaluation. We use two different methods. Firstly, taking each high-similarity pair (a kanji and its distractor), we randomly chose a third kanji from the jōyō (common use) government kanji set of 1945 characters. Since high-similarity pairs are statistically rare, the flashcard kanji and the randomly-chosen kanji are highly unlikely to be similar. We then compare how well each metric can classify the two pairs by imposing the correct ordering on them, in the form of classification accuracy. The results of this form of evaluation are shown in Table 2. We include a theoretical random baseline of 0.500, since any decision has a 50% chance of being successful.

Metric	Accuracy
d_{stroke}	0.968
l_1	0.957
d_{radical}	0.648
random baseline	0.500

Table 2. Accuracy in selecting the high-similarity pair over the low-similarity pair. High similarity pairs were chosen from the flashcard data. Low similarity pairs were randomly selected from the jōyō kanji set.

These results are a complete reversal of our earlier evaluation in Table 1. Surprisingly, d_{radical} performs extremely poorly on this task in comparison to the other metrics, quite close to our random baseline. This suggests that it is poor at determining the correct ordering between high and medium similarity pairs, though our earlier evaluation suggests it broadly orders examples correctly across the whole spectrum, as evidenced by its performance on the similarity experiment data. Its precision is simply too low for these high-similarity cases, but it is precisely these cases we are interested in for useful search and error correction.

The d_{stroke} and l_1 norm perform very highly on this task, indicating the ease with which they can distinguish such pairs. However, this does not guarantee that the neighbourhoods they generate will be free from noise, since the real-world prevalence of highly similar characters is likely to be very low. To better determine what search results might be like, now consider each flashcard kanji as a query, and its high-similarity distractors as relevant documents (and implicitly all remaining kanji as irrelevant documents, i.e. dissimilar characters). We can then calculate the Mean Average Precision (MAP, i.e. the mean area under the precision–recall curve of a given query) and the precision at N neighbours, for varied N. This forms our second evaluation strategy, the results for which are presented in Table 3.

Metric	*MAP*	*p@1*	*p@5*	*p@10*
d_{stroke}	0.564	0.293	0.143	0.094
l_1	0.493	0.249	0.137	0.088
d_{radical}	0.347	0.176	0.089	0.064

Table 3. The mean average precision (MAP), and precision at N, for $N \in 1, 5, 10$, on the flashcard data set.

The precision statistics confirm the ranking of metrics found in the earlier classification task. The d_{stroke} metric outperforms l_1 by a greater margin in the MAP statistic and precision at $N = 1$, but narrows again for greater N. This suggests that it is more reliable in the upper reaches of the similarity ranking.

Based on our evaluation over the flashcard data set, the d_{stroke} method seems most promising to use for search purposes, despite its poor performance on the similarity experiment data set.

4 From similarity to search

After considering several character distance metrics, and evaluating them over our two data sets, we now consider their application to dictionary word search.

4.1 Overall model

The broad probability model for looking up words based on similar kanji is identical to the FOKS model for search based on readings, save that we query via kanji rather than readings. For a user query q for a target word w, this leads to Equation 3:

$$\begin{aligned}\Pr(w|q) &\propto \Pr(w)\Pr(q|w)\\ &= \Pr(w)\prod_i \Pr(q_i|w, q_0 \ldots q_{i-1})\\ &\approx \Pr(w)\prod_i \Pr(q_i|w_i) \qquad (3)\end{aligned}$$

We assume that each query characters q_i are chosen independently by the user, and that they use only the target character w_i in their decision. These assumptions may not hold in all cases. For example, suppose a user is trying to look up 遠足 [eNsoku] "excursion, trip", but can't read the first character 遠. If they know the word 速足 [haya-ashi] "quick march", they might be primed to use 速 in place of 遠 in their query more often than other more similar neighbours. Although there are such examples where users may use context to either consciously or unconsciously prime their queries, we nonetheless keep this approximation since it allows us to much more efficiently calculate search candidates dynamically at run-time.

The final line of Equation 3 requires two models to be supplied. The first, $\Pr(w)$, is the probability that a word will be looked up. Here we approximate using corpus frequency over the Nikkei newspaper data, acknowledging that a newspaper corpus may be skewed differently to learner data. The second model is our confusion model $\Pr(q_i|w_i)$, interpreted either as the probability of confusing kanji w_i with kanji q_i, or of the user intentionally selecting q_i to query for w_i. It is this model which we now focus on.

4.2 Confusion model

Our confusion model must account for two main factors. Firstly, it must be based on the visual similarity of two characters q_i and w_i, which we address by use of one of our distance metrics. Secondly, users will tend to confuse unknown characters with characters that they are already familiar with. This is also clear since the only characters they can input are those they already know. We thus propose a generic confusion model based a similarity measure between kanji.

$$\Pr(q_i|w_i) \approx \frac{\Pr(q_i)s(q_i,w_i)}{\sum_j \Pr(q_{i,j})s(q_{i,j},w_i)} \tag{4}$$

The confusion model uses a similarity function $s(q_i,w_i)$ and a kanji frequency model $\Pr(q_i)$ to determine the relative probability of confusing w_i with q_i amongst other candidates. So far we have only dealt with distance models. For our prototype, we use the following similarity metric:

$$s_{\text{stroke}}(x,y) = \frac{1}{1+d_{\text{stroke}}(x,y)}$$

For efficiency, we also limit this model to a finite number of candidates per kanji. This is important since we the precision will fall as the number of candidates per kanji increases, so there is a natural trade-off between providing enough candidates and limiting the noise in the candidates that are present. Finding the right balance will maximize the accessibility of this form of search.

We limit the candidates using a thresholding method borrowed from Clark and Curran (4), where our threshold is set as a proportion of the first candidate's score. For example, using 0.9 as our threshold, if the first candidate has a similarity score of 0.7 with the target kanji, we would then accept any neighbours with a similarity greater than 0.63. Using the s_{stroke} similarity measure, with a ratio of 0.9, there are 2.65 neighbours for each kanji jōyō character set.

Search by similar grapheme has an advantage when considering candidate pruning in comparison to searches by word reading: searches by reading are naturally ambiguous, so broadening the scope of the search and increasing the number of error-correction candidates may push exact-match results further down in the ranking. Grapheme-based search on the other hand is naturally exact; there can normally only be one successful match, so additional secondary candidates are not in direct competition with existing search practices.

Having discussed an implementation of similarity-based search for Japanese kanji, we now return to consider broader aspects of this approach to lookup.

5 Discussion and future work

A current difficulty in modelling this form of search is the lack of available search data to objectively evaluate the search before deployment. This restricts evaluation to a longer-term post-hoc analysis based on query logs. If sufficient logs are kept, this would also provide some additional real-world similarity and confusion data.

How can similarity models be improved further? The different similarity metrics we have explored also make different mistakes in the highest similarity cases. This suggests that some combination model might be able to vote on whether a given pair is high similarity or not. There is also much promise in further exploiting Apel and Quint's data set to better take advantage of the relative location of elements. One possibility is to use a tree edit distance (3) between kanji representations. Yet another is to determine formal bounding boxes for each component, and determine the overlap of these bounding boxes when determining the similarity of otherwise identical components.

There are two additional needs which we foresee in the future for this form of interface. Firstly, the current FOKS extension provides two separate forms of search: intelligent grapheme search and intelligent reading search. Allowing users to combine these searches more freely in the same query could aid usability.

6 Conclusion

We have proposed a method of searching the dictionary for Japanese words containing unknown kanji, based their visual similarity to familiar kanji. In order to achieve this, we have considered several metrics over characters, improving over existing baselines. Evaluating over a flashcard set led to a complete reversal of the earlier human evaluation, suggesting that features crucial to distinguishing mid-to-low similarity pairs are inadequate for detecting high-similarity pairs. Of the metrics discussed, the edit distance taken over stroke descriptions performed the best for high-similarity cases, and was thus used to construct similarity-based search at the word level.

REFERENCES

[1] Ulrich Apel and Julien Quint, 'Building a graphetic dictionary for Japanese kanji – character look up based on brush strokes or stroke groups, and the display of kanji as path data', in *Proceedings of the 20th International Conference on Computational Linguistics*, (2004).

[2] Slaven Bilac, *Intelligent Dictionary Interface for Learners of Japanese*, Master's thesis, Tokyo Institute of Technology, 2002.

[3] Philip Bille, 'A survey on tree edit distance and related problems', *Theoretical Computer Science*, **337**(1-3), 217–239, (2005).

[4] Stephen Clark and James R. Curran, 'The importance of supertagging for wide-coverage CCG parsing', in *Proceedings of 20th International Conference on Computational Linguistics (COLING)*, p. 282–288, (2004).

[5] *The Kodansha Kanji Learner's Dictionary*, ed., Jack Halpern, Kodansha International, Tokyo, 1999.

[6] Kumiko Tanaka-Ishii and Julian Godon, 'Kansuke: A kanji look-up system based on a few stroke prototype', in *Proceedings of 21st International Conference on Computer Processing of Oriental Languages*, Singapore, (December 2006).

[7] Michael S. Waterman, Temple F. Smith, and William A. Beyer, 'Some biological sequence metrics', *Advances in Mathematics*, **20**(20), 367–387, (1976).

[8] Chris Winstead, 'Electronic kanji dictionary based on "Dasher"', *Adaptive and Learning Systems, 2006 IEEE Mountain Workshop on*, 144–148, (2006).

[9] Su-Ling Yeh and Jing-Ling Li, 'Role of structure and component in judgments of visual similarity of Chinese characters', *Journal of Experimental Psychology: Human Perception and Performance*, **28**(4), 933–947, (2002).

[10] Lars Yencken and Timothy Baldwin, 'Modelling the orthographic neighbourhood for Japanese kanji', in *ICCPOL '06: Proceedings of the 21st International Conference on the Computer Processing of Oriental Languages*, (2006).