Measuring and Predicting Orthographic Associations: Modelling the Similarity of Japanese Kanji

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

Abstract

As human beings, our mental processes for recognising linguistic symbols generate perceptual neighbourhoods around such symbols where confusion errors occur. Such neighbourhoods also provide us with conscious mental associations between symbols. This paper formalises orthographic models for similarity of Japanese kanji, and provides a proof-of-concept dictionary extension leveraging the mental associations provided by orthographic proximity.

1 Introduction

Electronic dictionary interfaces have evolved from mere digitised forms of their paper ancestors. They now enhance accessibility by addressing the separate needs of language consumers and language producers, of learners from non-speakers to native speakers, and by targeting the specific difficulties presented by individual languages.

For languages with logographic orthographies, such as Japanese and Chinese, accessibility remains poor due to the difficulties in looking up an unknown character in the dictionary. The traditional method of character lookup in these languages 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 are interested in the perceptual process of identifying characters, in particular the behaviour of perception within dense visual neighbourhoods. Within the dictionary accessibility space, we are motivated by the potential to correct confusion errors, but also to leverage the mental associations provided by visual proximity to allow advanced learners to find unknown characters faster. As proof-of-concept, we propose a method for looking up unknown 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 Bilac (2002) 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 related lookup systems (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 related systems

2.1 Associative lookup systems

Associative lookup systems are based on the premise that characters and words form a highly

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (http://creativecommons.org/licenses/by-nc-sa/3.0/). Some rights reserved.

connected lexical network. They focus on finding and making accessible the mental links provided by proximity within this network. In contrast, systems which correct for confusion model plausible errors in order to recover from them. Examples of associative systems are as follows:

	Semantic (for producers)	**Orthographic** (for consumers)
Monolingual	Visual WordNet	*this paper*
Bilingual	standard bilingual dictionaries	Pinyomi dictionary interface

Ferret and Zock (2006) introduce the distinction between language producers as encoders of semantic information, and language consumers as decoders of orthographic (or phonetic) information. We first consider systems to aid production of language.

Systems for production give form and sound to known semantics. The most common such systems are bilingual dictionaries which associate words in one language with their near-synonyms in a second language. Even within a monolingual context, the problem of selecting the right word can be difficult, whether the difficulty is one of limited knowledge or simply one of access, as in the case of the tip-of-the-tongue problem. Work in this area (Zock, 2002; Zock and Bilac, 2004) has more recently focused on extending WordNet with syntagmatic relationships (Ferret and Zock, 2006). Access could take the form of the Visual WordNet Project.[1]

For language consumers, the challenge is to find the meaning or sound of a word with known form. For logographic languages, where characters are entered phonetically[2] using an input method editor, computer input of an unknown word with known form remains difficult, since the reading is unknown.

In a bilingual context, the Pinyomi Chinese-Japanese dictionary interface overcomes this obstacle by allowing Japanese speakers to look up a Chinese word via the Japanese-equivalent characters based on orthographic associations between similar characters (Yencken et al., 2007).

Our proposed extension to the FOKS dictionary is functionally similar to Pinyomi, but in a monolingual Japanese context. In our case, a Japanese word containing unknown characters is found by querying with known characters that are visually similar. Unlike Pinyomi, which uses an ideogram transliteration table to determine associations, we use direct models of character similarity to determine associations.

2.2 Kanji lookup systems

We next provide a brief review of five kanji lookup systems in order to situate our proposed interface appropriately.

The SKIP (System of Kanji Indexing by Patterns) system of lookup provides an indexing scheme based on a kanji's overall shape rather than its primary radical (Halpern, 1999). For example, 明 [aka] "bright" has skip code 1-4-4, with the first number indicating it is horizontally split into two parts, and the second and third numbers representing the respective stroke counts of the two parts.

The Kansuke dictionary simplifies the method of counting strokes, to form a three-number code representing the horizontal, vertical and other strokes that make up a character (Tanaka-Ishii and Godon, 2006). 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.

The Kanjiru dictionary (Winstead, 2006) attempts to interactively assemble a character by shape and stroke via mouse movements, providing the user with structural ways of building up components until the desired character is found.

Finally, hand-writing interfaces attempt to circumvent the computer input problem altogether, but still suffer from several issues: the awkwardness of mouse input for drawing characters; sensitivity to both stroke order and connectivity of components; and the difference in hand-writing styles between learners and native speakers.

These lookup methods contrast with our proposed similarity-based search in several ways.

Firstly, our method combines word- and character-level information directly, yet provides the means to lookup words with unknown characters without the use of wildcards. The downside to this is that the user needs to use kanji in the search query, limiting potential users to intermediate and advanced learners with some knowledge of kanji.

Secondly, we are able to cater to both intentional similarity-based searches, and unintentional input errors, increasing the accessibility of the base dictionary. This approach shares much with the FOKS dictionary interface (Bilac, 2002), which provides

[1] http://kylescholz.com/projects/wordnet/

[2] A notable exception is the Wubizixing lookup method for Chinese.

error-correcting lookup for reading-based dictionary queries. Suppose, for example, a user wishes to look up the word 山車 “festival float”, 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” and guess the word reading as [yamakuruma]. The correct reading [dashi] cannot be guessed from the word’s parts, but our educated guess would lead the user to the word and provide access to both the correct reading and meaning.

This approach is complementary to our proposed method. Suppose, analogously, that the user wishes to look up the word 訪問 but is unfamiliar with the first kanji. A query for 方問 would trigger an inference based on the similarity between 訪 and 方, and provide the desired word in the results, allowing the user to determine both its pronunciation [hōmoN] and its meaning “visit”.

3 Modelling similarity

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 directly on orthographic confusion. For a brief discussion, see Yencken and Baldwin (2006).

Broadly, current literature suggests that kanji recognition may be hierarchical, 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.

The most directly relevant study involved two experiments by Yeh and Li (2002). In a sorting task, subjects tended to categorise characters by their structure, rather than their shared components. In a subsequent search task, presence of shared structure between target and distractors was the dominant factor in subjects’ response times.

We previously proposed two naive kanji similarity measures: a cosine similarity metric operating on boolean radical vectors, and the l_1 norm (Manhattan distance) between rendered images of kanji (Yencken and Baldwin, 2006). Evaluating on a set of human similarity judgements, we determined that the cosine similarity method outperformed the

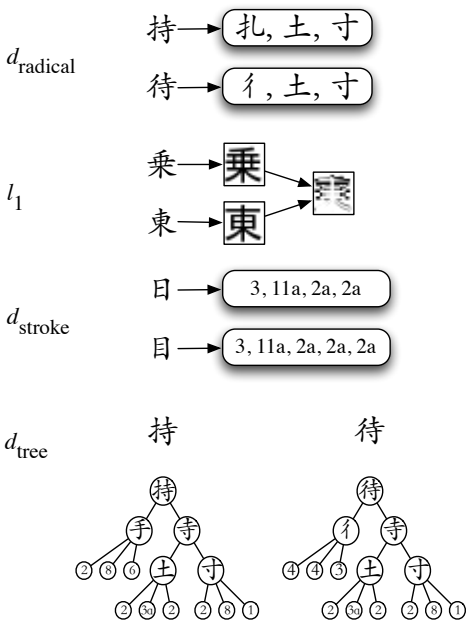

Figure 1: A summary of our kanji distance metrics

l_1 norm, although it had lower precision for high-similarity pairs.

3.1.1 Bag of radicals with shape

When learners of Japanese study a new character, they do not study its strokes in isolation, but instead build on prior knowledge of 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 number of strokes for any kanji, making such kanji easier to chunk and recall in memory. Furthermore, radicals can provide cues to the meaning and pronunciation of characters which contain them.[3]

The original metric used in Yencken and Baldwin (2006) simply calculates 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 times each radical occurs within a kanji. Hence, 木, 林 and 森 are all considered identical (radical = 木), as are 日 and 晶 (radical = 日). The metric is cal-

[3]For example, kanji containing the radical 月, such as 胸 [mune] “chest” and 腕 [ude] “arm”, are reliably body parts. Kanji containing the radical 同, as in 銅 [dō] “copper” and 胴 [dō] “body”, often have the Chinese or *on* reading [dō] amongst their valid pronunciations.

culated by:

$$d_{\text{radical}}(x,y) = 1 - \frac{r_x \cdot r_y}{|r_x||r_y|} \quad (1)$$

To address radical multiplicity, and the findings of Yeh and Li's study, we set the above metric to unit distance whenever the two characters differ in their basic shape. To approximate shape, we use 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,[4] and radical membership in the Radkfile.[5]

This change allows the metric to distinguish between examples with repeated components. The altered metric aims to capture the visual and semantic salience of radicals in kanji perception, and to also take into account some basic shape similarity.

3.1.2 Distance of rendered images

In contrast to the previous approach, we can consider kanji as arbitrary symbols rendered in print or on screen, and then attempt to measure their similarity. The simplest way to do this is to simply render each kanji to an image of fixed size, and to then use some distance metric over images.

A common and simple distance metric is the l_1 norm, which simply sums the difference in luminance between pixels of the two images for some alignment. Fortunately, all kanji are intended to occupy an identically sized block, so alignment is via a grid, constant across all kanji. Considering $p_x(i,j)$ to be the luminance of the pixel at position (i,j) of rendered kanji x, we evaluate the l_1 norm as follows:

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

This calculation depends on the image representation chosen, and could differ slightly across fonts, image sizes and rasterisation methods. We used the MS Gothic font, rendering to 80x80 images, with anti-aliasing.

This metric is aimed at capturing the general overlap of strokes between the two characters, along with 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 in distinguishing near neighbours.

[4] http://www.csse.monash.edu.au/~jwb/kanjidic.html

[5] http://www.csse.monash.edu.au/~jwb/kradinf.html

3.1.3 Stroke edit distance

A third possibility is to reduce kanji to the very strokes used to write them. Two features of the orthography make this possible: (1) kanji are not arbitrary symbols, but configurations of strokes chosen from within a finite and limited set; and (2) each kanji has a precise stroke order which is consistent for reused kanji components, such that if two or more arbitrary components were combined to form a new pseudo-character, native speakers would largely agree on the stroke order.

To define a metric based on strokes, we need both a source of stroke data and a comparison method. For stroke data, we look to a hierarchical data set for Japanese kanji created by Apel and Quint (2004). 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 taxonomy of some 26 stroke types (46 including sub-variants).

For any given kanji, we can flatten its hierarchy to generate an ordered sequence of strokes: a signature for that character. The natural distance metric across such sequences is the string edit distance. This forms our d_{stroke} metric.

Much useful information is preserved within stroke signatures. Since radicals are written in sequence, they form contiguous blocks in the signature. The edit distance will thus align shared radicals when their position is similar enough. Since components are usually drawn in a left-to-right, top-to-bottom order, the order of components in a signature also reflects their position as part of the larger character. Finally, it provides a smooth blending from stroke similarity to radical similarity, and can recognise the similarity between pairs like 日 [hi] "sun" and 目 [me] "eye".

3.1.4 Tree edit distance

In our previous approach, we discarded much of the hierarchical information available, relying on stroke order to approximate it. We can instead use the full data, and calculate the ordered *tree edit distance* between kanji XML representations. Tree edit distance is defined as the length of the shortest sequence of *inserts*, *deletions* and *relabellings* required to convert one tree into another (Bille, 2005). Though a cost function between labels can be specified, we gave inserts/deletions and relabellings unit cost.

Figure 1 provides an overview of the structure of each kanji's representation. Actual trees also contain phonetic elements, radicals, and stroke groups whose strokes are spread across several non-contiguous blocks. Another motivation for including tree edit distance is to determine if this additional information is useful in determining kanji similarity.

3.2 Evaluation

We evaluate our distance metrics over three data sets.

The first data set is the human similarity judgements from Yencken and Baldwin (2006). This data set is overly broad in that it weights the ability to distinguish low and medium similarity pairs equally with distinguishing medium and high similarity pairs. It is clear that for most applications, determining the high similarity pairs with high precision is most important. Nevertheless, this data set is useful for comparing our metrics with those proposed in previous research.

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 drawn from the White Rabbit JLPT Level 3[6] 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.

Our third data set is based on human confusability judgements for kanji pairings.

3.2.1 Similarity experiment data

The first data consists of human similarity judgements to pairs of kanji, scored on a 5 point scale (Yencken and Baldwin, 2006). The experiment had 179 participants, covering a broad range of Japanese proficiency. The key participant groupings are: (1) non-speakers of Chinese, Japanese or Korean (Non-CJK); (2) Japanese second-language learners (JSL); and (3) Japanese first-language speakers (JFL). Figure 2 gives the rank correlation ρ between each metric and a rater, averaged over all raters in each proficiency group.

For each metric, the mean rank correlation increased with the participants' knowledge of

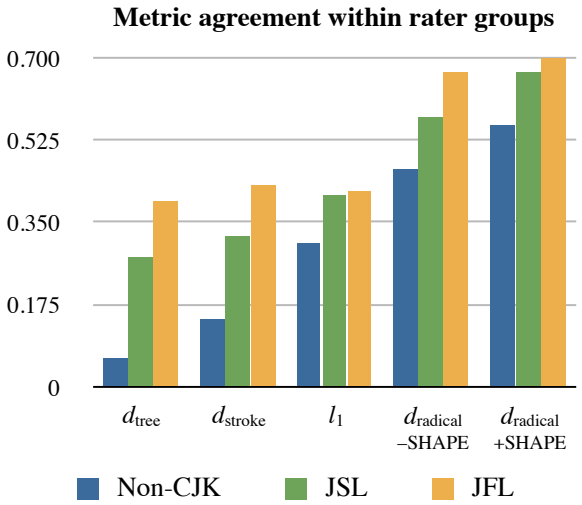

Figure 2: Mean value of Spearman's rank correlation ρ over rater groups for each metric ($d_{\text{radical}}(-\text{SHAPE})$ is the original metric, and $d_{\text{radical}}(+\text{SHAPE})$ is our augmented version)

Japanese (from Non-CJK to JSL to JFL), indicating that the raters made more motivated and consistent similarity judgements. The d_{radical} $(+\text{SHAPE})$ metric dominates the other metrics, including the original d_{radical} $(-\text{SHAPE})$, at all levels of knowledge. This confirms the salience of radicals and the tendency for individuals to classify kanji by their broad shape, as suggested by Yeh and Li (2002). l_1, d_{stroke} and d_{tree} perform poorly in comparison. Interestingly, these three metrics have large performance differences for non-speakers, but not for native-speakers.

Despite overall poor performance from our new metrics, we were able to improve on the original d_{radical} $(-\text{SHAPE})$. We now evaluate over the flashcard data set for comparison.

3.2.2 Flashcard data set

The flashcard data differs greatly from the previous experimental data, as it consists of only human-selected high-similarity pairs. Accordingly, we took two approaches to evaluation.

Firstly, for each high-similarity pair (a *pivot* kanji and its distractor), we randomly select a third kanji from the jōyō character set[7] and combine it with the pivot to form a second pair which is highly likely to be low similarity. 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 evaluation are shown in Table 1. We include a theoretical random baseline of 0.500, since any decision has a

[6] Japanese Language Proficiency Test: the standard government test for foreigners learning Japanese.

[7] The "common use" government kanji set, containing 1945 characters.

Metric	Accuracy
d_{tree}	0.979
d_{stroke}	0.968
l_1	0.957
d_{radical}	0.648
random baseline	0.500

Table 1: Accuracy at detecting which of two pairs (flashcard vs. random) has high similarity

Metric	*MAP*	*p@1*	*p@5*	*p@10*
d_{stroke}	0.594	0.313	0.151	0.100
d_{tree}	0.560	0.313	0.149	0.094
l_1	0.503	0.257	0.139	0.089
d_{radical}	0.356	0.197	0.087	0.063

Table 2: The mean average precision (MAP), and precision at $N \in \{1, 5, 10\}$ over the flashcard data

50% a priori chance of being successful.

The performance of d_{radical} – close to our random baseline, despite performing best in the previous task – is suggestive of the different characteristics of this task. In particular, a metric which orders well across the broad spectrum of similarity pairs may not be well suited to identifying high-similarity pairs, and vice-versa.

The other three metrics have accuracy above 0.95 on this task, indicating the ease with which they can identify high-similarity 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 dictionary search results might be like, we 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 for a query set) and the precision at N neighbours, for varied N. The results of this approach are presented in Table 2.

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 similarity ranking.

3.2.3 Distractor pool experiment

The flashcard data provides good examples of high-similarity pairs, but suffers from several problems. Firstly, the constraints of the flashcard format limit the number of high-similarity neighbours that can be presented on each flashcard to at most two; in some cases we might expect more. Secondly, the methodology behind the selection of these high-similarity neighbours is unclear.

For these reasons, we conducted an experiment to attempt to replicate the flashcard data. 100 kanji were randomly chosen from the JLPT 3 set (hereafter *pivots*). For each pivot kanji, we generated a pool of possible high-similarity neighbours in the following way. Firstly, we seeded the pool with the neighbours from the flashcard data set. We then added the highest similarity neighbour as given by each of our similarity metrics. Since these could overlap, we iteratively continued adding an additional neighbour from all of our metrics until our pool contained at least four neighbours.

Native or native-like speakers of Japanese were solicited as participants. After a dry run, each participant was presented with a series of pivot kanji. For each pivot kanji, they were asked to select from its pool of neighbours which (if any) might be confused for that kanji based on their graphical similarity. The order of pivots was randomised for each rater, as was the order of neighbours for each pivot. Kanji were provided as images using MS Gothic font for visual consistency across browsers.

Three participants completed the experiment, selecting 1.32 neighbours per pivot on average, less than 1.86 per pivot provided by the flashcard data. Inter-rater agreement was quite low, with a mean κ of 0.34 across rater pairings, suggesting that participants found the task difficult. This is unsurprising, since as native speakers the participants are experts at discriminating between characters, and are unlikely to make the same mistakes as learners. Comparing their judgements to the flashcard data set yields a mean κ of 0.37.

Ideally, this data generates a frequency distribution over potential neighbours based on the number of times they were rated as similar. However, since the number of participants was small, we simply combined the neighbours with high-similarity judgements for each pivot, yielding an average of 2.45 neighbours per pivot. Re-evaluating our metrics on this data gives the figures in Table 3.

Metric	*MAP*	*p@1*	*p@5*	*p@10*
d_{stroke}	1.046	0.530	0.228	0.146
d_{tree}	1.028	0.540	0.228	0.136
l_1	0.855	0.480	0.200	0.117
d_{radical}	0.548	0.270	0.122	0.095

Table 3: The mean average precision (MAP), and precision at $N \in \{1, 5, 10\}$ over the distractor data

Compared to the flashcard data set, the ordering and relative performance of metrics is similar, with d_{stroke} marginally improving on d_{tree}, but both significantly outperforming l_1 and d_{radical}. The near-doubling of high similarity neighbours from 1.32 to 2.45 is reflected by a corresponding increase in MAP and precision@N scores, though the effect is somewhat reduced as N increases.

4 From similarity to search

Having examined several character distance metrics, and evaluated them over our three data sets, we now consider their application to dictionary word search.

4.1 Overall model

Our 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 substitute readings for kanji in our query. A unigram approximation leads us to Equation 3 below, where $q = q_0 \dots q_n$ is the query given by the user, $w = w_0 \dots w_n$ is the desired word, and each q_i and w_i is a kanji character:

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

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 is 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 that we now focus on.

4.2 Confusion model

Although we can construct a confusion model using our distance metric alone, it is clear that frequency effects will occur. For example, the likelihood of confusion is increased if the target w_i is rare and unknown, but q_i is a highly-similar high-frequency neighbour; certainly this is a typical use case for intentional similarity-based querying. We thus propose a generic confusion model based a similarity measure between kanji:

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

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. We convert the desired distance metric d into s according to $s(x, y) = 1 - d(x, y)$ if the range of d is $[0, 1]$, or $s(x, y) = \frac{1}{1+d(x,y)}$ if the range of d is $[0, \infty)$.

To maximise the accessibility of this form of search, we must find the appropriate trade-off between providing sufficient candidates and limiting the noise. We use a thresholding method borrowed from Clark and Curran (2004), 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 d_{stroke} metric with a ratio of 0.9, there are on average 2.65 neighbours for each kanji in the jōyō character set.

4.3 Evaluating search

Search by similar grapheme has an advantage to search by word reading: reading results are naturally ambiguous due to homophony in Japanese, and attempts to perform error correction may interfere with exact matches in the results ranking. Grapheme-based search may have only one exact match, so additional secondary candidates are not in direct competition with existing search practices.

We can estimate the accessibility improvement given by this form of search as follows. Let us assume that learners study kanji in frequency order. For each kanji learned, one or more high-similarity neighbours also become accessible. Taking all pairings of kanji within the JIS X 0208-1990 character set, using the d_{stroke} metric with a cutoff ratio of 0.9, and assuming full precision on the neighbour graph this generates, we get the accessibility curve found in Figure 3. Our baseline is a single kanji accessible for each kanji learned.

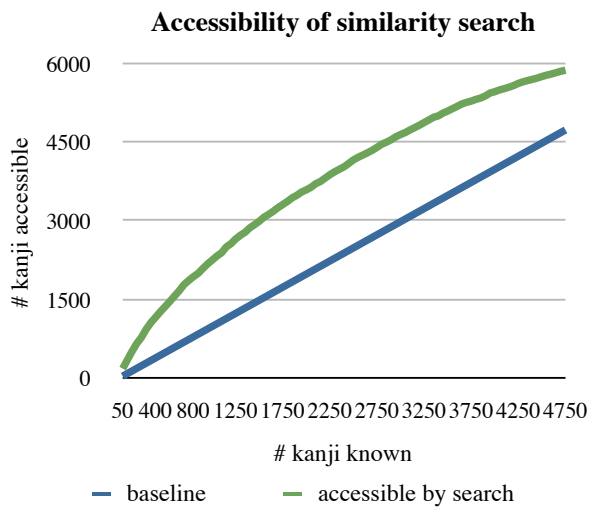

Figure 3: The accessibility improvement of kanji similarity search

Our actual precision makes the proportion of usable neighbours smaller; we will thus need to expose the user to a larger set of candidates to get this level of improvement. Improvements in precision and recall are still needed to reduce noise.

5 Discussion and future work

A current difficulty in evaluating this form of search is the lack of available query data to objectively evaluate the search before deployment. This restricts evaluation to longer-term post-hoc analysis based on query logs. Such logs will also provide additional real-world similarity and confusion data to improve our metrics.

This form of search is directly extensible to Chinese, and is limited only by the availability of character data. Indeed, preliminary similarity models for Chinese already exist (Liu and Lin, 2008). Our similarity modelling may also suggest approaches for more general symbol systems that lack adequate indexing schemes, for example heraldry.

There is much potential in the adaption of dictionaries as drill tutors in the context of language learning (Zock and Quint, 2004). The models presented in this paper could provide dynamic kanji drills, to aid early learners to distinguish similar kanji and provide challenge more advanced learners.

6 Conclusion

We have proposed a method of searching the dictionary for Japanese words containing unknown kanji, based on their visual similarity to familiar kanji. In order to achieve this, we have considered several metrics over characters, improved on existing baselines and evaluated further over a flashcard set. Of these metrics, the edit distance taken over stroke descriptions performed the best for high-similarity cases, and was used to construct similarity-based search at the word level.

References

[Apel and Quint2004] Apel, Ulrich and Julien Quint. 2004. 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 *Proc. COLING 2004*, Geneva, Switzerland.

[Bilac2002] Bilac, Slaven. 2002. Intelligent dictionary interface for learners of Japanese. Master's thesis, Tokyo Institute of Technology.

[Bille2005] Bille, Philip. 2005. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1-3):217–239.

[Clark and Curran2004] Clark, Stephen and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proc. COLING 2004*, page 282–288, Geneva, Switzerland.

[Ferret and Zock2006] Ferret, Olivier and Michael Zock. 2006. Enhancing electronic dictionaries with an index based on associations. In *Proc. COLING/ACL 2006*, pages 281–288, Sydney, Australia.

[Halpern1999] Halpern, Jack, editor. 1999. *The Kodansha Kanji Learner's Dictionary*. Kodansha International, Tokyo.

[Liu and Lin2008] Liu, Chao-Lin and Jen-Hsiang Lin. 2008. Using structural information for identifying similar Chinese characters. In *Proc. ACL 2008: HLT, Short Papers (Companion Volume)*, pages 93–96, Columbus, Ohio.

[Tanaka-Ishii and Godon2006] Tanaka-Ishii, Kumiko and Julian Godon. 2006. Kansuke: A kanji look-up system based on a few stroke prototype. In *Proc. ICCPOL 2006*, Singapore.

[Winstead2006] Winstead, Chris. 2006. Electronic kanji dictionary based on "Dasher". *Proc. IEEE SMCals 2006* , pages 144–148, Logan, USA.

[Yeh and Li2002] Yeh, Su-Ling and Jing-Ling Li. 2002. Role of structure and component in judgements of visual similarity of Chinese characters. *Journal of Experimental Psychology: Human Perception and Performance*, 28(4):933–947.

[Yencken and Baldwin2006] Yencken, Lars and Timothy Baldwin. 2006. Modelling the orthographic neighbourhood for Japanese kanji. In *Proc. ICCPOL 2006*, Singapore.

[Yencken et al.2007] Yencken, Lars, Zhihui Jin, and Kumiko Tanaka-Ishii. 2007. Pinyomi - dictionary lookup via orthographic associations. In *Proc. PACLING 2007*, Melbourne, Australia.

[Zock and Bilac2004] Zock, Michael and Slaven Bilac. 2004. Word lookup on the basis of associations: from an idea to a roadmap. In *Proc. COLING 2004*, pages 89–95, Geneva, Switzerland.

[Zock and Quint2004] Zock, Michael and Julien Quint. 2004. Why have them work for peanuts, when it is so easy to provide reward? Motivations for converting a dictionary into a drill tutor. In *Proc. PAPILLON 2004*, Grenoble, France.

[Zock2002] Zock, Michael. 2002. Sorry, what was your name again, or how to overcome the tip-of-the tongue problem with the help of a computer? In *Proc. COLING 2002*, pages 1–6, Taipei, Taiwan.